

SAINT PETERSBURG STATE UNIVERSITY

Manuscript Copy

Nina Golyandina

**GENERAL APPROACH TO THEORY AND METHODOLOGY OF  
SINGULAR SPECTRUM ANALYSIS**

Scientific specialization

1.2.2. Mathematical modeling, numerical methods and software complexes

Dissertation is submitted for the degree of doctor of  
physical and mathematical sciences

Saint-Petersburg — 2023

# Contents

<b>Introduction</b>	7
<b>Notation</b>	15
<b>Chapter 1. General structure of SSA-family methods</b>	16
1.1. Time series and digital images: common problems	17
1.2. Generic scheme of the SSA family and the main concepts	18
1.2.1. General SSA method	19
1.2.2. The main concepts	21
1.3. Different versions of SSA	24
1.3.1. Decomposition of $\mathbf{X}$ into a sum of rank-one matrices	25
1.3.2. Versions of SSA dealing with different forms of the object	27
1.4. Separability in SSA	28
1.5. Forecasting, interpolation, low-rank approximation and parameter estimation in SSA	29
1.6. Comparison of SSA with other methods	30
1.6.1. Fourier transform, filtering, noise reduction	30
1.6.2. Parametric regression	32
1.6.3. ARIMA and ETS	32
1.7. Bibliographical notes	33
1.7.1. Some recent applications of SSA	35
1.7.2. SSA for preprocessing / combination of methods	35
1.8. Basic SSA for one-dimensional time series	36
1.8.1. Method	36
1.8.2. Model of time series	38
1.8.3. Separability and choice of parameters	39
1.8.4. Decomposition of e-m harmonics	42
1.8.5. Algorithm	43
1.8.6. Modification Toeplitz SSA for stationary time series	44
1.8.7. Example of identification and decomposition	45
1.8.8. Example of problems with separability	47
1.9. Forecasting and parameter estimation for one-dimensional time series	50
1.9.1. Signal extraction via projections	50
1.9.2. Parameter estimation	50
1.9.3. Forecasting	54

<b>Chapter 2. Decomposition step for one-dimensional time series . . . . .</b>	60
2.1. SSA with projection . . . . .	60
2.1.1. SSA with centering . . . . .	60
2.1.2. SSA with projection . . . . .	61
2.1.3. Examples . . . . .	67
2.1.4. Summary of results . . . . .	73
2.2. Iterative Oblique SSA . . . . .	74
2.2.1. Oblique SVD and SSA . . . . .	74
2.2.2. Method . . . . .	82
2.2.3. Algorithms . . . . .	84
2.2.4. Example. Separability of sine waves with close frequencies . . . . .	86
2.3. Filter-adjusted O-SSA and SSA with derivatives . . . . .	88
2.3.1. SSA with derivatives. Variation for strong separability . . . . .	88
2.3.2. Filter-adjusted O-SSA . . . . .	91
2.3.3. Examples . . . . .	94
2.4. SSA-ICA . . . . .	100
2.4.1. Maximization of entropy . . . . .	100
2.4.2. SOBI-AMUSE . . . . .	104
2.5. Automatic identification . . . . .	111
2.5.1. Low-frequency method for trend identification . . . . .	112
2.5.2. Frequency method for identifying the oscillating component . . . . .	113
2.5.3. Method for identifying the oscillatory component by the regularity of angles	116
2.5.4. Comparison of methods for harmonic identification . . . . .	120
<b>Chapter 3. Model-based problems: gap filling, signal estimation, signal detection</b>	123
3.1. Subspace-based method of SSA gap filling . . . . .	123
3.1.1. Overview of gap filling methods in SSA . . . . .	123
3.1.2. Preliminary results . . . . .	128
3.1.3. Lagged vectors and trajectory spaces of time series of finite rank with missing data . . . . .	131
3.1.4. Finding trajectory spaces of the initial time series and of its additive components . . . . .	132
3.1.5. Comments to implementation of the subspace gap-filling method . . . . .	134
3.2. Possibility of construction of weights in the HSLRA problem . . . . .	137
3.2.1. Matrix and vectors convolution . . . . .	139
3.2.2. Relation between vector and matrix forms of HSLRA . . . . .	139
3.2.3. Generating functions and convolution of banded matrices . . . . .	140
3.2.4. Autocovariance matrices and their inverses for AR(1) and AR(2) models .	143

3.2.5.	Studying existence of solutions to the problem of blind deconvolution for the matrices proportional to inverses of covariance matrices in autoregressive models . . . . .	145
3.2.6.	Non-existence of the deconvolution $\mathbf{W}=\mathbf{A}*\mathbf{B}$ for a stationary AR( $p$ ) model with general $p$ and diagonal $\mathbf{B}$ . . . . .	151
3.3.	Detection of signals by Monte Carlo singular spectrum analysis: Multiple testing	154
3.3.1.	Statistical approach to hypothesis testing . . . . .	155
3.3.2.	Monte Carlo SSA . . . . .	158
3.3.3.	Numerical investigation . . . . .	163
3.4.	Choice of parameters . . . . .	170
3.4.1.	Introduction . . . . .	170
3.4.2.	Signal subspace . . . . .	171
3.4.3.	Signal extraction . . . . .	174
3.4.4.	Recurrent SSA forecast . . . . .	178
3.4.5.	Subspace-based methods of parameter estimation . . . . .	180
3.4.6.	The rate of convergence . . . . .	181
3.4.7.	Choice of the window length and separability . . . . .	186
3.4.8.	SSA processing of stationary time series . . . . .	189
3.4.9.	SVD-origins of SSA and the choice of SSA parameters . . . . .	193
<b>Chapter 4. SSA for multivariate time series</b>		195
4.1.	MSSA analysis . . . . .	195
4.1.1.	Method . . . . .	195
4.1.2.	Algorithm . . . . .	199
4.2.	Elements of MSSA theory . . . . .	200
4.2.1.	Separability . . . . .	200
4.2.2.	Ranks and subspaces . . . . .	201
4.2.3.	Decomposition of e-m harmonics . . . . .	203
4.2.4.	Comments on 1D-SSA and MSSA . . . . .	205
4.3.	MSSA forecasting . . . . .	206
4.3.1.	Method . . . . .	207
4.3.2.	Fast vector forecasting algorithm . . . . .	211
4.3.3.	Simulated example: numerical comparison . . . . .	213
4.4.	Automation of grouping in MSSA . . . . .	216
<b>Chapter 5. Shaped and multidimensional SSA</b>		222
5.1.	Shaped 2D-SSA . . . . .	222
5.1.1.	Method . . . . .	223
5.1.2.	Rank of shaped arrays . . . . .	225

5.1.3.	Algorithm . . . . .	226
5.2.	Particular cases of Shaped SSA . . . . .	227
5.2.1.	Shaped 1D-SSA . . . . .	227
5.2.2.	MSSA . . . . .	228
5.2.3.	2D-SSA . . . . .	230
5.2.4.	M-2D-SSA . . . . .	234
5.2.5.	Comments on nD extensions . . . . .	234
5.3.	Example of Shaped SSA . . . . .	236
<b>Chapter 6. Package Rssa</b>	. . . . .	239
6.1.	Brief introduction to RSSA . . . . .	240
6.2.	Implementation efficiency . . . . .	241
6.2.1.	Efficiency of the R-package RSSA . . . . .	241
6.2.2.	Example of calculations in RSSA . . . . .	242
6.3.	Unified approach for implementation of the SSA scheme . . . . .	242
6.3.1.	Basic analysis . . . . .	242
6.3.2.	Different modifications of the decomposition step . . . . .	244
6.3.3.	Subspace-based methods . . . . .	244
<b>Chapter 7. Applications to real-life data</b>	. . . . .	245
7.1.	EOP time series prediction using singular spectrum analysis . . . . .	245
7.1.1.	Data sources . . . . .	247
7.1.2.	Automatic choice of parameters . . . . .	247
7.1.3.	Forecasts on the test period . . . . .	250
7.1.4.	Conclusions . . . . .	252
7.2.	Application of SSA to density estimation . . . . .	252
7.2.1.	A new approach to density estimation . . . . .	252
7.2.2.	Performance of the SSA estimates . . . . .	258
7.2.3.	Application to c.d.f. estimation in market research . . . . .	261
7.3.	Two-exponential models of gene expression patterns for noisy experimental data .	263
7.3.1.	Methods . . . . .	264
7.3.2.	Results and discussion . . . . .	269
7.4.	Shaped Singular Spectrum Analysis for Quantifying Gene Expression, with Application to the Early <i>Drosophila</i> Embryo . . . . .	272
7.4.1.	Materials . . . . .	273
7.4.2.	Choice of parameters, separability and component identification . . . . .	274
7.4.3.	Periodic patterns produced by unmixing algorithms . . . . .	275
7.4.4.	Conclusions . . . . .	285

7.5. Shaped 3D Singular Spectrum Analysis for Quantifying Gene Expression, with Application to the Early Zebrafish Embryo . . . . .	285
7.5.1. Data . . . . .	288
7.5.2. Method . . . . .	289
7.5.3. Example for spherical-cap nuclear pattern . . . . .	293
<b>Conclusion</b> . . . . .	299
<b>Acknowledgment</b> . . . . .	302
<b>Bibliography</b> . . . . .	303

# Introduction

The main subject of research in this work is time series, namely the sequence of real values  $\mathbf{X} = (x_1, \dots, x_N)$ , where the indices usually denote equidistant time moments, but can also correspond to spatial coordinates. The analysis and forecasting of time series is a very common task, since a huge amount of data concerns characteristics varying over time. The work also studies time series systems  $\mathbf{X} = (\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(s)})$ , where  $\mathbf{X}^{(k)} = (x_1^{(k)}, \dots, x_N^{(k)})$ ,  $k = 1, \dots, s$ .

In addition to time series, digital images in the form of an array  $\mathbf{X} = \{x_{ij}\}_{i,j=1}^{N_x, N_y}$  are considered. The most general form of investigated objects is  $\mathbf{X} = \{x_{\mathbf{i}}\}_{\mathbf{i} \in \mathcal{N}}$ , where  $\mathbf{i}$  is a multi-index of some dimension, the set  $\mathcal{N}$  is linearly ordered and sets the shape of the object.

The initial object is assumed to be the sum of some components, e.g., signal and noise (a random process with zero mean, not necessarily stationary), a noisy sum of sinusoids, the sum of trend, periodicity, and noise, the sum of two deterministic components. Here, the signal means the non-random component of the object, and the trend/pattern means the non-random and slowly changing component. Note that a parametric model is often considered for a trend, but in this case, generally speaking, it is not assumed. The tasks can be to extract, analyze and predict the components of the signal, to fill in gaps, to check the hypothesis about the (non-)existence of a signal in a time series.

To solve the described problems, the singular spectrum analysis (SSA for short) has been taken as the basis.

The basic modification algorithm for time series analysis looks like this: A real-valued time series  $\mathbf{X} = (x_1, \dots, x_N)$  of length  $N$  and window length  $L$ ,  $1 < L < N$  is input. SSA transfers the time series into a trajectory matrix of size  $L \times K$ ,  $K = N - L + 1$ , by the embedding operator  $\mathcal{T}_L$ :

$$\mathbf{X} = \mathcal{T}_L(\mathbf{X}) = \begin{pmatrix} x_1 & x_2 & \dots & x_K \\ x_2 & \dots & \dots & x_{K+1} \\ \dots & \dots & \dots & \dots \\ x_L & x_{L+1} & \dots & x_N \end{pmatrix}.$$

Then we consider the decomposition of the matrix into a sum of elementary matrices of rank 1, in the basic version this is done using singular values decomposition, after that the matrices are summed to  $m$  groups and finally using the operator  $\mathcal{T}_L^{-1}$  we get the result:  $\mathbf{X} = \tilde{\mathbf{X}}^{(1)} + \dots + \tilde{\mathbf{X}}^{(m)}$ .

The essence of the method is to convert the original object into a so-called trajectory matrix in some way that depends on the object type, then decompose the trajectory matrix into a sum of matrices of rank 1, grouping and summing matrices by groups, followed by a return to the decomposition of the original object. Thus, the input of the method is the sum of objects, and the output is the evaluation of its summands. Besides the decomposition of the initial object, the result of the method is characteristics of summands defining their structure, which allows one to predict non-random components of series by forecasting the found structure, to fill gaps in them,

as well as to estimate parameters of the time series.

Let for simplicity the original object consist of two interpretable terms  $\mathbf{X} = \mathbf{X}^{(1)} + \mathbf{X}^{(2)}$ . Then the result of the method is the decomposition  $\mathbf{X} = \tilde{\mathbf{X}}^{(1)} + \tilde{\mathbf{X}}^{(2)}$ . The following questions arise: under what conditions on the terms and parameters of the method applied can the method isolate  $\mathbf{X}^{(1)}$  such that  $\tilde{\mathbf{X}}^{(1)}$  equals  $\mathbf{X}^{(1)}$  exactly or approximately (*separability* conditions), how to group elementary components to extract  $\mathbf{X}^{(1)}$  when separability conditions are met, how to check these separability conditions, how the signal estimation error looks like in the case of approximate separability and what recommendations can be given on the choice of method parameters to reduce this error.

The notion of separability of time series components using SSA was introduced by Nekrutkin V. V. [1], which significantly advanced the theoretical approach to the method. However, the basic version of the SSA method had disadvantages in terms of component separability. One of the goals of this work is to create modifications of SSA that expand the class of separable components of time series. Although theoretical results require that the components of the time series fit some model, which we will discuss later, the application of these results does not require that the components of the series obey this model. Therefore, modifications of Basic SSA, which is nonparametric, leave the method still nonparametric.

Although some other problems, such as prediction and gap-filling, require the model of the predicted component of the time series to be specified, the situation is about the same as in the construction of the time series decomposition. Namely, the theory is constructed for the case of a signal satisfying the model, but, for example, the prediction method is also applicable to the case where the signal approximately satisfies the model.

Let us describe the model for a time series. Let  $\mathbf{S} = (s_1, \dots, s_N)$  be the signal (more precisely, the component of the time series). Set the window length  $L$ ,  $1 < L < N$ ;  $K = N - L + 1$ . Consider the signal trajectory matrix  $\mathbf{S} = \mathcal{T}_L(\mathbf{S})$ . Let  $r$  denote the rank of the matrix  $\mathbf{S}$ .

The model can be formulated in different forms:

1.  $\mathbf{S}$  is a Hankel matrix of small rank  $r < \min(L, K)$ ; in this case, the model can be parameterized based on the parameterization of the space  $\text{colspace}(\mathbf{S})$  or orthogonal addition to it. Such time series are called *time series of finite rank*.
2. The time series is governed by a linear recurrence relation (LRR):

$$s_n = \sum_{k=1}^r a_k s_{n-k}, a_r \neq 0, n = r+1, \dots$$

Such time series are called *LRR-governed time series*.

3. time series The time series has an explicit parametric form as a finite sum:

$$s_n = \sum_k P_k(n) \exp(\alpha_k n) \sin(2\pi\omega_k n + \phi_k), \quad (1)$$

where  $P_k(n)$  is a polynomial of  $n$ ,  $\exp(\alpha_k n) = \rho_j^n$  for  $\rho_j = e^{\alpha_j}$ .

The first model is more general, but under some non-restricted conditions (e.g., for infinite time series), these three models are equivalent. One of the disadvantages of the third kind of model is that it is necessary to specify an explicit kind of signal element, while for the first two variants, it is sufficient to specify only the value  $r$ .

In real-life problems, the model assumes that a noisy signal of finite rank  $\mathbf{X} = \mathbf{S} + \mathbf{R}$  is observed, where  $\mathbf{S}$  is a finite-rank signal,  $\mathbf{R}$  is a random noise with zero expectation; noise does not have to be stationary. In this model, SSA-type methods can estimate the signal, estimate signal parameters, fill in gaps, and construct predictions. Mathematical results refer to the construction of prediction and gap-filling methods, the development of an algebraic approach to finite rank objects, and the construction of efficient (fast and robust) finite rank signal estimation algorithms.

If a signal has exactly a finite rank (the so-called low-rank signal), then the problem of its estimation is solved using the approximation of the trajectory matrix and is called Hankel structured low-rank approximation (HSLRA). To solve this problem, we can use either model 1 or model 2. In both cases, there is a parameterization for time series of rank  $r$  that differs from the explicit parameterization in the form of model 3. This problem is solved both explicitly as a weighted least squares method and in the matrix form of model 1. A large number of results on the HSLRA can be found in the book [2]. The Cadzow method [3] was proposed to solve the problem in matrix form independently of the development of SSA. Interestingly, the basic SSA method as a signal estimation method coincides with one iteration of the Cadzow method. This turns out to be a key difference, since Cadzow iterations are suitable only for the case of finite rank signals, whereas the SSA method is much more flexible.

The SSA method for one-dimensional time series processing is the most widely used one. However, the ideas of the method turned out to be in demand both for the analysis of multidimensional time series (MSSA, Multivariate or Multi-channel SSA) and for the analysis of digital images (2D-SSA). The methods were not originally bound to SSA, in particular, the MSSA method was called the EEOF (extended empirical orthogonal functions) method.

Another goal of this work is to build a general scheme of the method and its implementation, allowing the application of the method to objects of any dimensionality, from multidimensional time series to  $n$ -dimensional images. In this case, the object can have any shape, as well as a sliding window on it. The general approach allows us to generalize the results for one-dimensional SSA to the general case.

Any method of analyzing real data is impractical without its effective implementation. The size of time series and, even more so, of images can be very large, so the computational time and memory costs must be reasonable. Initially, there was an opinion that the method is very labour-consuming and therefore not applicable to large data. Existing earlier implementations of the method were made directly following the algorithm, without finding an effective implementation. Therefore, it is highly relevant to develop a package that, firstly, uses fast implementations of

mathematical procedures, and, secondly, allows a unified analysis of objects of different shapes and dimensions.

## **Actuality of the theme**

There are many approaches to time series, some of which solve specific problems or are suitable for specific data models, and some of which claim to solve a whole range of problems. The main task is usually the forecasting of series and many methods are developed exactly for the forecasting. Such methods include, for example, ARIMA-type methods or ETS-type methods (models with exponential smoothing). These methods are able to work with data that includes trend and periodicity, however, for example, a period value must be set and a period must be the only one. In addition, there are fairly strong requirements on the number of periods that must fit within the length of the series. For ETS, the type of trend must be specified. The forecast in ARIMA strongly depends on the number of differentiations of the series. The big advantage of these methods is that there is an automatic selection of parameters, though in a certain data model. Thus, classical methods have both advantages and disadvantages.

SSA-type methods are also able to solve a very wide range of problems for time series of different structures. One of the main advantages of the method is that it is not necessary to specify initially the time series model (in particular, the period value and the trend form); the method is often called nonparametric, which is both a great advantage of the SSA method and a disadvantage due to difficulties in automating the method in case of its nonparametric application.

There are plenty of papers dealing with different problems of time series analysis and forecasting based, to this or that extent, on the theory of SSA. Therefore, developing a general approach to the theory and application of SSA is a vital task.

## **Aims of the thesis**

The main objectives of the thesis work are as follows.

1. Creation of a general methodology for the SSA method.
2. A unified approach to the development and structure of methods from the SSA family.
3. Improving the SSA separability of interpretable components of objects under study (time series, digital images) to increase the accuracy of their extraction.
4. A general approach to multidimensional generalizations of SSA that allows the processing of objects of different dimensions and shapes.
5. Selection of method parameters, testing the hypothesis of the existence of the signal.
6. Methodological support for the RSSA package in terms of its structure and implementation.
7. Study of the capabilities of the method in application to real-world problems.

## **Methodology and methods of investigation**

The methodology of the research carried out in this dissertation is a comprehensive approach to the problem, which allows scaling the proposed algorithms to objects of different structures. Both theoretical methods using linear algebra, mathematical analysis, probability theory and mathematical statistics, and numerical research methods based on statistical modeling are applied for this purpose. The R programming language is used to implement the algorithms.

### **Statements to defend:**

1. A general scheme of SSA methods has been created that allows both the construction of different decompositions of a time series, adapting to its structure, and the extension to analyze different objects, such as time series, time series systems, digital images and multidimensional objects of different shapes.
2. ProjSSA, Iterative OSSA, DerivSSA, and SSA-ICA algorithms are proposed and justified to improve the selection of components of the studied objects, such as trend/pattern, regular fluctuations and noise.
3. The problem of signal estimation by the least squares method is considered as the problem of weighted approximation by Hankel matrices of low rank. The impossibility of choosing weights to achieve equivalence of these problems is proved.
4. The influence of SSA method parameters on the quality of time series component extraction is investigated. Methods of automatic identification of time series components have been proposed.
5. A general approach for multidimensional generalizations of SSA, called ShapedSSA, has been developed, which allows processing not only rectangular-shaped objects but also objects of complex shapes, starting from time series with gaps and ending with multidimensional objects of complex shapes in a unified style.
6. A statistical test controlling the family-wise error rate is proposed to test the hypothesis of the (non)existence of the signal in the time series.
7. A methodological approach to the structure and implementation of the RSSA package has been developed, which allows effective implementation with a convenient interface.
8. The SSA method has been applied to the tasks of predicting Earth rotation parameters, estimating distribution densities, constructing a parametric model of bicoid gene expression profiles, and isolating patterns in multidimensional gene expression data.

## **Scientific novelty**

The basis of scientific novelty is a holistic approach to the whole variety of methods and objects under study. The obtained theoretical results are new.

## **Theoretical and practical significance**

The theoretical value consists in the theoretical justification of the algorithms and their properties, which allows for obtaining more accurate analysis results.

The practical value consists of the application of the constructed algorithms to real-world data and expanding the range of practical problems to which SSA family methods can be applied. It is confirmed by the possibility to obtain new results in the real-world fields of the problems, to which one of the SSA methods is applied.

## **Approbation of the work**

**Talks** The main results were presented at the following seminars and conferences.

1. Scientific seminar of the Department of Statistical Modeling.
2. National Scientific Conference SPISOK, 2016, 2017, 2019, 2022.
3. Research seminar at School of Mathematics, Cardiff University, UK, 2019.
4. Invited talk at Colloquium in School of Mathematics, Cardiff University, UK, 2017.
5. Invited talk at the international conference ‘Structured low-rank approximation’, Grenoble, France, 2015.
6. Invited talk at the international conference ‘Optimal decisions in statistics and data analysis’, Cardiff, UK, 2013.
7. Invited talk at the international conference ‘Singular Spectrum Analysis and its applications’, Beijing, China (2012).

## **Grants**

1. RFBR 20-01-00067 Development of mathematical methods of analysis and prediction of one-dimensional and multivariate time series within the framework of singular spectrum analysis (2020-2022, principal investigator)
2. RFBR 16-04-00821 Analysis, classification and modelling of patterns of gene expression in early embryogenesis of some model organisms. (2016-2018, principal investigator)
3. RFBR 15-04-06480 Molecular mechanisms of embryo’s axial organization on example of the model insects (2015-2017, participant)

4. RFBR 13-04-02137 Using variable expressivity in Drosophila segmentation to understand the mechanisms of phenotypic stability (2013-2014, participant)

## Publications

### Papers

Twenty-two papers in journals indexed in Scopus/WoS have been published on the topic of this dissertation: [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25].

### Conference proceedings and book chapters

On the topic of the thesis, three papers were published in conference proceedings indexed in Scopus/WoS: [26, 27, 28]

### Monographs

Three monographs (co-authored) have been written on the topic of the thesis [1, 29, 30], one of which was also published in the second expanded edition: [31].

### Other publications

There are publications in collections and conference proceedings [32, 33, 34, 35, 36, 37, 38, 39, 40, 41].

Additional publications in the journals include [42, 43, 44, 45].

The dissertation includes results from the papers obtained by the author or to a large extent by the author when working together on the result.

## Structure of the dissertation

The thesis consists of Introduction, seven Chapters, Conclusion and Bibliography. Below we briefly describe the contents of the chapters and cite the works containing the main results.

Chapter 1 proposes a general scheme of methods of the SSA family, in terms of which the basic concepts of the SSA method and the structure of the presentation in the following chapters are described. A scheme of the method is given, in which both modifications of the method which improve the separability of time series components and multivariate generalizations of the method are included. The first six sections of the chapter are an overview of the SSA family methods. The last two sections describe the basic SSA methods for time series analysis and prediction. The review is based on [20].

In Chapter 2 and Chapter 3, the problems of one-dimensional time series processing are considered. Chapter 2 proposes different methods to improve separability by changing the decomposition step. These methods are described in the papers: SSA with projection [16], Iterative Oblique SSA [9], Filter-adjusted O-SSA and SSA with derivatives [9], SSA-ICA [14]. Chapter 2 concludes with a section describing methods of automatic grouping for the extraction of trend and periodic components, for the applicability of which signal components must be separable, for example, using the methods described in the previous sections.

Chapter 3 is devoted to the solution of some problems related to the model, to which the component of the series exactly or approximately satisfies. The first problem is filling the gaps in the time series [4].

The second problem is the estimation of the signal, which is a series of finite rank. Algorithms to solve this problem have been proposed in [15, 22, 23, 24]. In [13, 19], the relation of weights in two different formulations of the problem for their equivalence is discussed.

The third problem is signal detection in a noisy time series by the Monte Carlo SSA method. We use red noise as noise (it is an SSA specificity), and sinusoidal signal with unknown frequency as a signal. As a solution to the problem, a criterion is constructed to test the hypothesis that the series consists only of noise. The name of the method does not exactly reflect the essence, because here we do not construct a decomposition of the time series by SSA, but only use singular vectors of the trajectory matrix to construct the criterion. The words “Monte Carlo” in the name means that the distribution of criterion statistics is constructed using surrogate data modelling. Although the Monte Carlo SSA method has been proposed for a long time, it has not been considered in terms of multiple testing, which is done in [25].

Chapter 3 concludes with a discussion of the choice of parameters in the SSA method for solving analysis and prediction problems [5].

Although the MSSA extension of SSA can be considered a special case of Shaped SSA, which is considered in Chapter 5, a separate Chapter 4 is devoted to the MSSA method. In addition to a description of MSSA, Chapter 4 includes a description of the time series system decomposition and its properties, a discussion of the features of MSSA, an introduction of the concept of matched series, and a numerical comparison of MSSA applied to a two-series system and SSA applied to each series separately. In addition, the rationale for fast vector SSA and MSSA prediction is given. The main results of this paper are in [12]. Also, the chapter proposes an extension of the automatic grouping algorithm in MSSA.

The approach to the SSA family methods as special cases of Shaped SSA is described in Chapter 5 following [12]. In addition to describing Shaped SSA, the chapter contains descriptions of special cases such as Shaped 1D-SSA applied to series with gaps, MSSA, and 2D-SSA whose theory is described in [40].

In short chapter 6, we present the RSSA package, in which the methods of the RSSA family are implemented efficiently and in the same style as described in the previous chapters [8, 12].

The last Chapter 7 contains several applications of the SSA method. They refer to the prediction of the rotation parameters of the Earth [28], density estimates from empirical data [7], gene expression analysis in a large number of works [17, 21, 10, 11, 26, 27, 18, 6], so three sections are devoted to the latter topic.

## Notation

SVD	singular value decomposition
LRR	linear recurrence relation
SSA	singular spectrum analysis
HSLRA	Hankel structured low-rank approximation
$\mathbf{X}$ or $\mathbf{X}_N$	time series or ordered collection of numbers
$N$	length or size of $\mathbf{X}$
$L$	window length
$K = N - L + 1$	the number of $L$ -lagged vectors obtained from $\mathbf{X}$
$\mathcal{T}$	embedding operator
$\mathbf{X} = \mathcal{T}(\mathbf{X})$	the trajectory matrix associated with $\mathbf{X}$
$\ \mathbf{X}\ _{\mathbb{F}}$	Frobenius matrix norm
rank $\mathbf{X}$	rank of $\mathbf{X}$
$\mathcal{X} = \mathcal{X}^{(L)}(\mathbf{X})$	$L$ -trajectory space of $\mathbf{X}$
rank $_L(\mathbf{X})$	$L$ -rank of $\mathbf{X}$
$\Pi_{\mathcal{H}}$	hankelization operator
$\lambda_i$	$i$ th eigenvalue of the matrix $\mathbf{X}\mathbf{X}^T$
$U_i$	$i$ th eigenvector of the matrix $\mathbf{X}\mathbf{X}^T$
$V_i = \mathbf{X}^T U_i / \sqrt{\lambda_i}$	$i$ th factor vector of the matrix $\mathbf{X}$
$(\sqrt{\lambda_i}, U_i, V_i)$	$i$ th eigentriple of the SVD of the matrix $\mathbf{X}$
$\mathbf{I}_M$	identity $M \times M$ matrix
$\mathbb{R}^L$	Euclidean space of dimension $L$
$\mathbb{R}^{L \times K}$	space of $L \times K$ matrices
span( $P_1, \dots, P_n$ )	linear subspace spanned by vectors $P_1, \dots, P_n$

## Chapter 1

# General structure of SSA-family methods

In this chapter, we describe, how the SSA methodology can be extended in many different directions, and also unify different approaches and modifications within the SSA framework. Also, we introduce the main notions, principles and ideas of SSA, and present a unified view of SSA.

We start with Section 1.1 with a brief description of the solved problems.

Section 1.2 describes the generic structure of all methods from the SSA family and introduces the main concepts essential for understanding different versions of SSA and for making the application of SSA in practice efficient.

Section 1.3 classifies different versions of SSA. As explained in that section, there are two complementary directions in which versions of SSA can be created: one is related to geometrical features of the object  $X$  and the other one is determined by the choice of the procedure of decomposition of the trajectory matrix into rank-one matrices. These two directions of variations of SSA are not related to each other so that any extension of SSA related to the geometry of  $X$  can be combined with any procedure of decomposition of the trajectory matrix.

Section 1.4 discusses the concept of separability, which is the most theoretically important concept of SSA. Achieving separability (for example, of a signal from noise) is the key task of SSA in most applications. A correct understanding of this concept is therefore imperative for making a particular application of SSA reliable and efficient. We will be returning to separability in many discussions within the book.

Section 1.5 briefly introduces the main underlying model used to apply SSA for common problems such as forecasting, imputation of missing data and monitoring structural stability of time series. In the one-dimensional case, this model assumes that a part of the series can be described by a linear recurrent relation and in particular, by a sum of damped sinusoids. Estimation of the parameters of this model often constitutes the main objective in signal processing.

In Section 1.6, we briefly discuss the place of SSA among other methods of time series analysis, signal and image processing and also provide a short overview of recent publications where a comparison of SSA with several traditional methods has been made.

In Section 1.7 we make a short historical survey of SSA, refer to recent applications of SSA and to papers which discuss combinations of SSA with other methods.

Since Chapters 2–5 contains mostly original results, we add to this chapter Sections 1.8 and 1.9 with known algorithms, which will be used further.

The material of this chapter is contained in the papers [12] and [20]. Sections 1.8 and 1.9 follow the books [1, 31].

## 1.1. Time series and digital images: common problems

### Decomposition

Let us observe a time series  $\mathbf{X} = (x_1, \dots, x_N)$ , where  $\mathbf{X} = \mathbf{T} + \mathbf{P} + \mathbf{N}$ ,  $\mathbf{T}$  is a trend,  $\mathbf{P}$  contains regular oscillations and  $\mathbf{N}$  is noise. The common problem is to construct a decomposition  $\mathbf{X} = \tilde{\mathbf{T}} + \tilde{\mathbf{P}} + \tilde{\mathbf{N}}$ , see Fig. 1.1, which allows one to estimate the trend  $\mathbf{T}$ , the whole signal  $\mathbf{S} = \mathbf{T} + \mathbf{P}$ , or periodic components  $\mathbf{P}$  ( $\mathbf{P}$  can consist of periodic components with different fundamental periods). Once the signal is estimated and its structure is detected, different signal-based methods such as forecasting can be applied.

If the time series does not contain a signal, then the problem of filtering can be stated; this problem is reduced again to constructing the decomposition, into low- and high-frequency components.

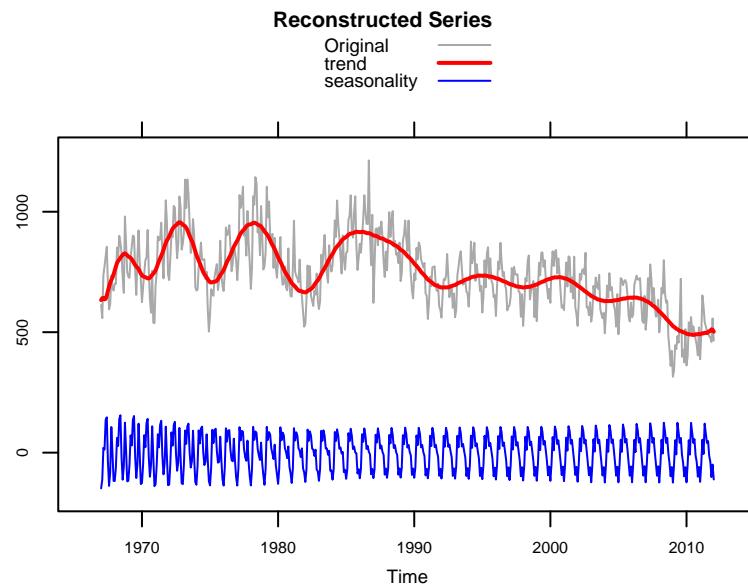


Figure 1.1: ‘MotorVehicle’, monthly sales: Decomposition.

Many problems in digital image processing are similar to the problems stated for time series. Let

$$\mathbf{X} = \begin{pmatrix} x_{1,1} & \dots & x_{1,N_2} \\ \dots & \dots & \dots \\ x_{N_1,1} & \dots & x_{N_1,N_2} \end{pmatrix}$$

be a digital image, which is modelled as a decomposition  $\mathbf{X} = \mathbf{T} + \mathbf{P} + \mathbf{N}$  into a pattern, regular oscillations (e.g. a texture) and noise. Then the problem of estimating the decomposition components arises again. See Fig. 1.2, which demonstrates the extraction of a pattern (which is obtained after the removal of regular oscillations).

Note that ‘digital images’ is a common name for 2D data, since one of the dimensions may be temporal. The decomposition problem for multidimensional data is also important in higher

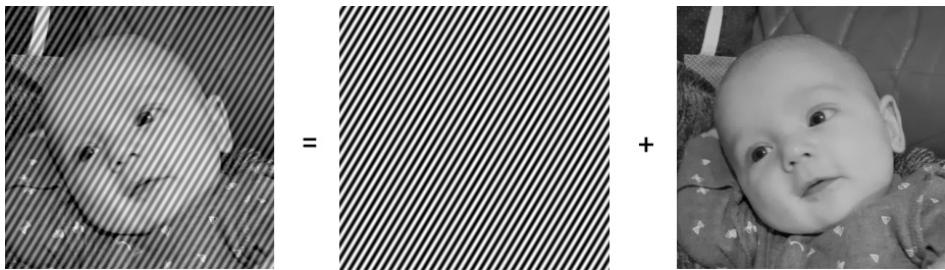


Figure 1.2: ‘Maya’: Image decomposition.

dimensions; these dimensions may again be of different nature. For example, one can consider both 3D spatial data and 2D data with the third temporal dimension as 3D data; and so on. We will call data with  $n$  dimensions  $n$ D data.

### Other time-series/image problems

The results of decomposition (in particular, of signal extraction) allow one to solve many problems for objects with different dimensions and shapes; among these problems are

- trend/tendency extraction;
- smoothing/filtering;
- noise reduction;
- extraction of periodic components (including seasonal adjustment);
- frequency estimation;
- construction of a model and parameter estimation;
- forecasting/prediction (of the extracted signal);
- missing data imputation;
- change-point detection.

Thus, the decomposition serves as a starting point for solving many important problems.

## 1.2. Generic scheme of the SSA family and the main concepts

We use SSA as a generic term for a family of methods based on a sequential application of the four steps schematically represented in Figure 1.3 below and briefly described in the next section.

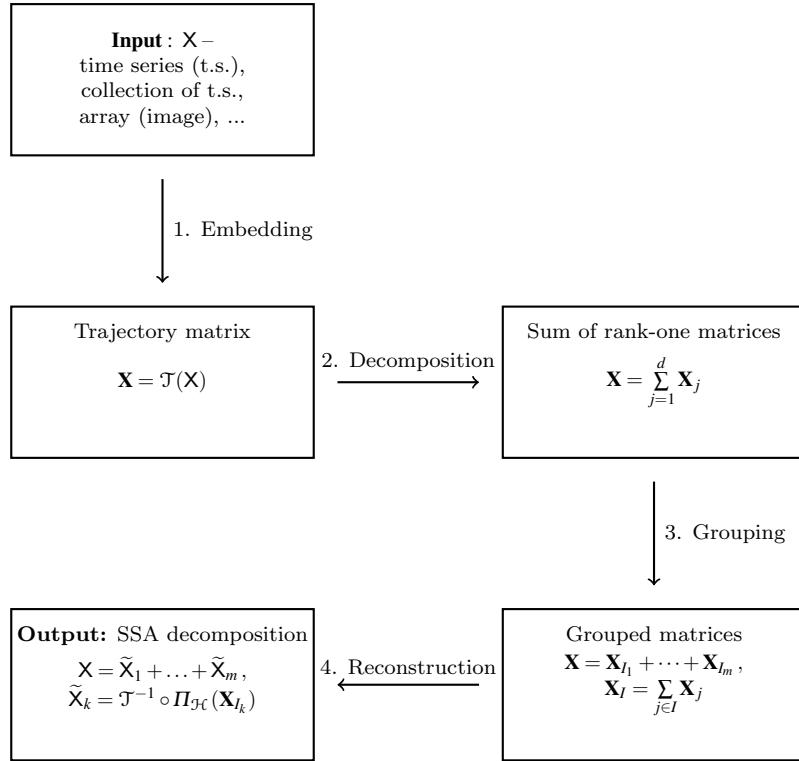


Figure 1.3: SSA family: Generic scheme.

### 1.2.1. General SSA method

We define an SSA method (or simply SSA) as any method performing the four steps depicted in Figure 1.3 and briefly described below. The input object  $\mathbf{X}$  is an ordered collection of  $N$  numbers (e.g. a time series or a digital image). We denote the set of such objects by  $\mathcal{M}$ . Unless stated otherwise, the entries of  $\mathbf{X}$  are assumed to be real numbers although a straightforward generalization of the main SSA method to the case of complex numbers is available.

**Input:**  $\mathbf{X}$ , an ordered collection of  $N$  numbers.

**Output:** A decomposition of  $\mathbf{X}$  into a sum of identifiable components:  $\mathbf{X} = \tilde{\mathbf{X}}_1 + \dots + \tilde{\mathbf{X}}_m$ .

**Step 1. Embedding.** The so-called *trajectory matrix*  $\mathbf{X} = \mathcal{T}(\mathbf{X})$  is constructed, where  $\mathcal{T}$  is a linear map transforming the object  $\mathbf{X}$  into an  $L \times K$  matrix of certain structure. Let us denote the set of all possible trajectory matrices by  $\mathcal{M}_{L,K}^{(H)}$ . The letter  $H$  is used to stress that these matrices have Hankel-related structure.

As an example, in 1D-SSA (that is, SSA for the analysis of one-dimensional real-valued time series),  $\mathbf{X} = (x_1, \dots, x_N)$  and  $\mathcal{T} = \mathcal{T}_{1D-SSA}$  maps  $\mathbb{R}^N$  to the space of Hankel matrices  $L \times K$  with equal values on the anti-diagonals, where  $N$  is the series length,  $L$  is the window length, which is a

parameter, and  $K=N-L+1$ :

$$\mathcal{T}_{\text{1D-SSA}}(\mathbf{X}) = \begin{pmatrix} x_1 & x_2 & x_3 & \dots & x_K \\ x_2 & x_3 & x_4 & \dots & x_{K+1} \\ x_3 & x_4 & x_5 & \dots & x_{K+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_L & x_{L+1} & x_{L+2} & \dots & x_N \end{pmatrix}. \quad (1.1)$$

**Step 2. Decomposition of  $\mathbf{X}$  into a sum of matrices of rank 1.** The result of this step is the decomposition

$$\mathbf{X} = \sum_i \mathbf{X}_i, \quad \mathbf{X}_i = \sigma_i U_i V_i^T, \quad (1.2)$$

where  $U_i \in \mathbb{R}^L$  and  $V_i \in \mathbb{R}^K$  are vectors such that  $\|U_i\| = 1$  and  $\|V_i\| = 1$  for all  $i$  and  $\sigma_i$  are non-negative numbers.

The main example of this decomposition is the conventional singular value decomposition (SVD) for real-valued matrices  $\mathbf{X}$ . If this conventional SVD is used, then we call the corresponding SSA method ‘Basic SSA’ [1, Chapter 1]. Let  $\mathbf{S} = \mathbf{X}\mathbf{X}^T$ ,  $\lambda_1 \geq \dots \geq \lambda_L \geq 0$  be *eigenvalues* of the matrix  $\mathbf{S}$ ,  $d = \text{rank } \mathbf{X} = \max\{j : \lambda_j > 0\}$ ,  $U_1, \dots, U_d$  be the corresponding *eigenvectors*, and  $V_j = \mathbf{X}^T U_j / \sqrt{\lambda_j}$ ,  $j = 1, \dots, d$ , be *factor vectors*. Denote  $\mathbf{X}_j = \sqrt{\lambda_j} U_j V_j^T$ . Then the SVD of the trajectory matrix  $\mathbf{X}$  can be written as

$$\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d. \quad (1.3)$$

The triple  $(\sqrt{\lambda_j}, U_j, V_j)$  consisting of the singular value  $\sigma_j = \sqrt{\lambda_j}$ , the left singular vector  $U_j$  and the right singular vector  $V_j$  of  $\mathbf{X}$  is called *jth eigentriple*.

**Step 3. Grouping.** The input in this step is the expansion (1.2) and a specification of how to group the components of (1.2).

Let  $I = \{i_1, \dots, i_p\} \subset \{1, \dots, d\}$  be a set of indices. Then the resultant matrix  $\mathbf{X}_I$  corresponding to the group  $I$  is defined as  $\mathbf{X}_I = \mathbf{X}_{i_1} + \dots + \mathbf{X}_{i_p}$ .

Assume that a partition of the set of indices  $\{1, \dots, d\}$  into  $m$  disjoint subsets  $I_1, \dots, I_m$  is specified. Then the result of Grouping step is the *grouped matrix decomposition*

$$\mathbf{X} = \mathbf{X}_{I_1} + \dots + \mathbf{X}_{I_m}. \quad (1.4)$$

If only one subset,  $I$ , of  $\{1, \dots, d\}$  is specified, then we still can assume that a partition of  $\{1, \dots, d\}$  is provided: this is the partition consisting of two subsets,  $I$  and  $\bar{I} = \{1, \dots, d\} \setminus I$ . In this case,  $\mathbf{X}_I$  is usually associated with the pattern of interest (for example, signal) and  $\mathbf{X}_{\bar{I}} = \mathbf{X} - \mathbf{X}_I$  can be treated simply as a residual.

The grouping of the expansion (1.2), where  $I_k = \{k\}$ , is called *elementary*.

**Step 4. Reconstruction.** At this step, each matrix  $\mathbf{X}_{I_k}$  from the decomposition (1.4) is transferred back to the form of the input object  $\mathbf{X}$ . This transformation is performed optimally in the following sense: for a matrix  $\mathbf{Y} \in \mathbb{R}^{L \times K}$ , we seek for an object  $\tilde{\mathbf{Y}} \in \mathcal{M}$  that provides the minimum to  $\|\mathbf{Y} - \mathcal{T}(\tilde{\mathbf{Y}})\|_F$ , where  $\|\mathbf{Z}\|_F = (\sum_{ij} |z_{ij}|^2)^{1/2}$  is the Frobenius norm of  $\mathbf{Z} = [z_{ij}] \in \mathbb{R}^{L \times K}$ .

Let  $\Pi_{\mathcal{H}} : \mathbb{R}^{L \times K} \rightarrow \mathcal{M}_{L,K}^{(H)}$  be the orthogonal projection from  $\mathbb{R}^{L \times K}$  to  $\mathcal{M}_{L,K}^{(H)}$  in the Frobenius norm. Then  $\tilde{\mathbf{Y}} = \mathcal{T}^{-1} \circ \Pi_{\mathcal{H}}(\mathbf{Y})$ . The projection  $\Pi_{\mathcal{H}}$  is simply the averaging of the entries corresponding to a given element of an object, see Section 1.2.2 for details. For example, in 1D-SSA the composite mapping  $\mathcal{T}^{-1} \circ \Pi_{\mathcal{H}}$  uses the averaging along anti-diagonals so that  $\tilde{y}_k = \sum_{(i,j) \in \mathcal{A}_k} (\mathbf{Y})_{ij} / |\mathcal{A}_k|$ , where  $\mathcal{A}_k = \{(i,j) : i + j = k + 1, 1 \leq i \leq L, 1 \leq j \leq K\}$ .

Let  $\widehat{\mathbf{X}}_k = \mathbf{X}_{I_k}$  be the reconstructed matrices,  $\tilde{\mathbf{X}}_k = \Pi_{\mathcal{H}} \widehat{\mathbf{X}}_k$  be the corresponding trajectory matrices and  $\tilde{\mathbf{X}}_k = \mathcal{T}^{-1}(\tilde{\mathbf{X}}_k)$  be the reconstructed objects. Then the resulting decomposition of the initial object  $\mathbf{X}$  is

$$\mathbf{X} = \tilde{\mathbf{X}}_1 + \dots + \tilde{\mathbf{X}}_m. \quad (1.5)$$

If the grouping is elementary, then the reconstructed objects  $\tilde{\mathbf{X}}_k$  in (1.5) are called *elementary components*.

For convenience of referencing, Steps 1 and 2 of the generic SSA scheme are sometimes combined into the so-called ‘Decomposition stage’ and Steps 3 and 4 are combined into ‘Reconstruction stage’.

### 1.2.2. The main concepts

#### Parameters of the SSA methods

Step 1: parameters of the linear map  $\mathcal{T}$ . For a given object  $\mathbf{X}$ , the trajectory matrix  $\mathbf{X} = \mathcal{T}(\mathbf{X})$  can be computed in different ways. In 1D-SSA, there is only one parameter in Step 1, the window length  $L$ .

Step 2: no parameters if the conventional SVD is performed. Otherwise, if an alternative decomposition of  $\mathbf{X}$  into a sum of rank-one matrices is used, there may be some parameters involved, see Section 1.3.1.

Step 3: the parameter (or parameters) that defines the grouping.

Step 4: no extra parameters.

#### Separability

A very important concept in the SSA methodology is separability. Let  $\mathbf{X} = \mathbf{X}_1 + \mathbf{X}_2$ . (Approximate) separability of  $\mathbf{X}_1$  and  $\mathbf{X}_2$  means that there exists a grouping such that the reconstructed

object  $\tilde{\mathbf{X}}_1$  is (approximately) equal to  $\mathbf{X}_1$ . The representation  $\mathbf{X} = \mathbf{X}_1 + \mathbf{X}_2$  can be associated with many different models such as ‘signal plus noise’, ‘trend plus the rest’ and ‘texture plus the main image’.

If  $\mathbf{X} = \mathbf{X}_1 + \mathbf{X}_2$  and  $\mathbf{X}_1$  and  $\mathbf{X}_2$  are approximately separable, then SSA can potentially separate  $\mathbf{X}_1$  from  $\mathbf{X}_2$ ; that is, it can find a decomposition  $\mathbf{X} = \tilde{\mathbf{X}}_1 + \tilde{\mathbf{X}}_2$  so that  $\tilde{\mathbf{X}}_1 \approx \mathbf{X}_1$  and  $\tilde{\mathbf{X}}_2 \approx \mathbf{X}_2$ .

Consider, as an example, Basic SSA. Properties of the SVD yield that the (approximate) orthogonality of columns and of rows of the trajectory matrices  $\mathbf{X}_1$  and  $\mathbf{X}_2$  of  $\mathbf{X}_1$  and  $\mathbf{X}_2$  can be considered as natural separability conditions.

There is a well-elaborated theory of separability for the one-dimensional time series [1, Sections 1.5 and 6.1]. Many important decomposition problems, from noise reduction and smoothing to trend, periodicity and signal extraction, can be solved by SSA. The success of 1D-SSA in making separation between separable objects is related to the simplicity of the Hankel structure of the trajectory matrices and the optimality features of the SVD.

We will come back to the important concept of separability in Section 1.4 where we define the main characteristic which is used in SSA for separability checking.

## Information for grouping

The theory of SSA exhibits the ways of helping to detect the components  $(\sigma_i, U_i, V_i)$  in the decomposition (1.2) related to the object component with certain properties to perform proper grouping under the condition of separability. One of the rules is that  $U_i$  and  $V_i$  (eigenvectors and factor vectors in the case of Basic SSA) produced by an object component emulates the properties of this component. For example, in Basic 1D-SSA the eigenvectors produced by slowly-varying series components are slowly-varying, the eigenvectors produced by a sine wave are sine waves with the same frequency; and so on. These properties help to perform the grouping by visual inspection of eigenvectors and also by some automatic procedures, see Section 2.5.

## Trajectory spaces and signal subspaces

Let  $\mathbf{X}$  be the trajectory matrix corresponding to some object  $\mathbf{X}$ . The *column (row) trajectory space* of  $\mathbf{X}$  is the linear subspace spanned by the columns (correspondingly, rows) of  $\mathbf{X}$ . The term ‘trajectory space’ usually means ‘column trajectory space’. The column trajectory space is a subspace of  $\mathbb{R}^L$ , while the row trajectory space is a subspace of  $\mathbb{R}^K$ . In general, for real-world data the trajectory spaces coincide with the corresponding Euclidean spaces, since they are produced by a noisy data. However, in the ‘signal plus noise’ model, when the signal has rank-deficient trajectory matrix, the signal trajectory space can be called ‘signal subspace’. Both column and row signal subspaces can be considered; note that the dimensions of the row and column subspaces coincide.

## Objects of finite rank

The class of objects that suit SSA are the so-called objects of finite rank. We say that the object (either time series or image) has *L*-rank *r* if the rank of its trajectory matrix is  $r < \min(L, K)$ ; that is, the trajectory matrix is rank-deficient. If the *L*-rank *r* does not depend on the choice of *L* for any sufficiently large object and trajectory matrix sizes, then we say that the object is of finite rank and has rank *r*, see Section 1.8.2 for rigorous definitions.

Since the trajectory matrices considered in SSA methods are either pure Hankel or consist of Hankel blocks, then the rank-deficient trajectory matrices are closely related to the objects satisfying some linear relations. These linear relations can be used for building forecasting methods. In the one-dimensional case, under some non-restrictive conditions, rank-deficient Hankel matrices are in the one-to-one correspondence with the linear recurrence relations (LRRs) of the form

$$x_n = a_1 x_{n-1} + \dots + a_r x_{n-r}$$

and therefore are related to the time series which can be expressed as sums of products of exponentials, polynomials and sinusoids, see Section 1.8.2.

Each SSA extension produces a class of objects of finite rank. The knowledge of ranks of objects of finite rank can help to recognize the rank-one components for the component reconstruction. For example, in order to reconstruct the exponential trend in the one-dimensional case, we need to group only one rank-one component (the exponential function has rank 1), while to reconstruct a sine wave we generally need to group two SVD components (the rank of a sine wave equals 2).

The real-life time series or images are generally not of finite rank. However, if a given object  $\mathbf{X}$  is a sum of a signal of finite rank and noise, then, in view of approximate separability, SSA may be able to approximately extract the signal and subsequently use the methods that are designed for series of finite rank.

## Reconstruction (averaging)

Let us formally describe the operation of reconstruction of a matrix used in Step 4 of the generic scheme described in Section 1.2.1. By analogy with the one-dimensional case this operation can also be called ‘averaging over diagonals’ even if the averaging will be performed over more complicated patterns.

Assume that the entries  $x_\tau$  of the object  $\mathbf{X} = \{x_\tau\}$  are indexed by the index  $\tau$  which can be simply a positive integer (for the one-dimensional series) or multi-index (for digital images).

A linear map  $\mathcal{T}$  is making a one-to-one transformation of  $\mathcal{M}$  to  $\mathcal{M}_{L,K}^{(H)}$ , the set of  $L \times K$  matrices of a specified structure. It puts elements of  $\mathbf{X}$  on certain places of the matrix  $\mathcal{T}(\mathbf{X}) = \mathbf{X} = [(\mathbf{X})_{ij}]$ .

Let  $\mathbf{e}_\tau \in \mathcal{M}$  be the object with 1 as the  $\tau$ th entry with all other entries zero. Define the set of indices

$$\mathcal{A}_\tau = \{(i, j) \text{ such that } (\mathbf{E}_\tau)_{ij} = 1\},$$

where  $\mathbf{E}_\tau$  is the matrix

$$\mathbf{E}_\tau = [(\mathbf{E}_\tau)_{ij}] = \mathcal{T}(\mathbf{e}_\tau) \in \mathcal{M}_{L,K}^{(H)}.$$

If  $\tau$  is the place of an element  $x_\tau \in \mathsf{X}$ , then  $(\mathbf{X})_{ij} = x_\tau$  for all  $(i, j) \in \mathcal{A}_\tau$ .

Assume now that  $\widehat{\mathbf{X}} \in \mathbb{R}^{L \times K}$  is an arbitrary  $L \times K$  matrix and we need to compute

$$\widetilde{\mathbf{X}} = (\widetilde{x}_\tau) = \mathcal{T}^{-1} \circ \Pi_{\mathcal{H}}(\widehat{\mathbf{X}})$$

by first making the orthogonal projection of  $\widehat{\mathbf{X}}$  to the set  $\mathcal{M}_{L,K}^{(H)}$  and then writing the result in the object space  $\mathcal{M}$ . This operation is the extension of the ‘diagonal averaging’ procedure applied in 1D-SSA: the elements  $\widetilde{x}_\tau$  of  $\widetilde{\mathbf{X}}$  are computed by the formula

$$\widetilde{x}_\tau = \sum_{(i,j) \in \mathcal{A}_\tau} (\widehat{\mathbf{X}})_{ij} / w_\tau = \langle \widehat{\mathbf{X}}, \mathbf{E}_\tau \rangle_F / \|\mathbf{E}_\tau\|_F^2, \quad (1.6)$$

where  $\mathbf{E}_\tau = \mathcal{T}(\mathbf{e}_\tau)$ ,  $w_\tau = |\mathcal{A}_\tau| = \|\mathbf{E}_\tau\|_F^2$  is the number of elements in the set  $\mathcal{A}_\tau$  and the Frobenius inner product  $\langle \cdot, \cdot \rangle_F$  is defined by

$$\langle \mathbf{Z}, \mathbf{Y} \rangle_F = \sum_{i,j} z_{ij} y_{ij}. \quad (1.7)$$

### 1.3. Different versions of SSA

Let us consider how the four steps of the generic scheme of SSA formulated in Section 1.2 can vary for different versions of SSA.

Step 1: the form of the object  $\mathbf{X}$  and hence the specificity of the embedding operator  $\mathcal{T}$  makes a big influence on how a particular version of SSA looks like.

Step 2: not only the conventional SVD but many other decompositions of  $\mathbf{X}$  into a sum of rank-one matrices could be used.

Step 3: formally, this step is exactly the same for all versions of SSA although the tools used to perform the grouping may differ.

Step 4: the embedding operator  $\mathcal{T}$  defined in Step 1 fully determines the operations performed at this step.

Therefore, we have two directions for creating different versions of SSA: the first direction is related to geometrical features of the object  $\mathbf{X}$  and a form of the embedding operator  $\mathcal{T}$ , while the second direction is determined by the form of the decomposition at Step 2. Essentially, Step 1 is determined by the form of the object; therefore its variations can be considered as extensions of 1D-SSA. If instead of SVD we use some other decomposition of  $\mathbf{X}$  into a sum of rank-one matrices at Step 2, then we call the corresponding algorithm a modification of Basic SSA. These

two directions of variations of SSA are not related to each other so that any extension of Step 1 can be combined with any modification of Step 2.

We start the discussion with outlining some modifications that can be offered for the use at Step 2.

### 1.3.1. Decomposition of $\mathbf{X}$ into a sum of rank-one matrices

#### Variations of SSA related to methods of decomposition

The conventional SVD formulated in the description of SSA in Section 1.2.1 is a decomposition of  $\mathbf{X}$  into a sum of rank-one matrices, which has some optimality properties, see [1, Chapter 4]. Therefore, Basic SSA, which is SSA with the conventional SVD used at Step 2, can be considered as the most fundamental version of SSA among all SSA methods.

Let us enumerate several variations of SSA, which could be useful for answering different questions within the framework of SSA.

A well known modification of Basic SSA is *Toeplitz SSA* (Section 1.8.6), which was created for dealing with stationary time series. This modification is devised for the analysis of a natural estimate of the auto-covariance matrix of the original time series and assumes that this time series is stationary. However, if the time series  $\mathbf{X}$  is non-stationary, then the reconstruction obtained by Toeplitz SSA can have a considerable bias.

An important variation of SSA is *SSA with projection* (Section 2.1). If we have a parametric model (which should be linear in parameters and agreeable with the finite-rank assumption) for one of the components of the series, such as trend of a one-dimensional series, then a projection on a suitable subspace is performed and is followed by a decomposition of the residual, e.g., by the SVD. The known methods of SSA with centering and SSA with double centering for extraction of constant and linear trends respectively are special cases of SSA with projection. More generally, an arbitrary polynomial trend can be extracted by a suitable version of SSA with projection. Another use of SSA with projection is to build a subspace from a supporting series and project the main series onto this subspace.

In some versions of SSA the intention is to improve separability properties of SVD. If we use an oblique version of the SVD, then the resulting SSA method becomes Oblique SSA. The following two versions of Oblique SSA seem to be useful in practice, namely *Iterative Oblique SSA* (Section 2.2) and *Filter-adjusted Oblique SSA* (Section 2.3). The latter is useful for separation of components with equal contribution.

### Nested application of different versions of SSA

Since Oblique SSA does not have good approximating features, it cannot replace Basic SSA which uses the conventional SVD. Therefore, Oblique SSA should be used in a nested manner so that Basic SSA is used first to extract several components without performing careful split of

these components and then one of the proposed oblique methods is used for separating the mixed components.

If we use Basic SSA for denoising and then some other version of SSA (like Independent Component Analysis or Oblique SSA) for improvement of separability, then we can interpret this as if using Basic SSA for preprocessing and using another method for a more refined analysis. There is, however, a significant difference between this and some methods mentioned in Section 1.7.2, where Basic SSA is used for preprocessing and then ARIMA or other methods of different nature are employed. Indeed, when we use Basic SSA for denoising and some other SSA technique like Oblique SSA for improvement of separability, then we are using the signal subspace estimated by Basic SSA rather than the estimated signal itself (recall that in the transition from the estimated signal subspace to the estimated signal we incur an additional error).

Let us schematically demonstrate the nested use of the methods as follows. Let  $\mathbf{X} = \mathbf{X}^{(1)} + \mathbf{X}^{(2)} + \mathbf{X}^{(3)}$  be a decomposition of the time series and  $\mathbf{X} = \tilde{\mathbf{X}}^{(1)} + \tilde{\mathbf{X}}^{(2)} + \tilde{\mathbf{X}}^{(3)}$  be the corresponding decomposition of the trajectory matrix of  $\mathbf{X}$ . Let Basic SSA return at Decomposition stage  $\mathbf{X} = \tilde{\mathbf{X}}^{(1,2)} + \tilde{\mathbf{X}}^{(3)}$  and assume that a chosen nested method makes the decomposition  $\tilde{\mathbf{X}}^{(1,2)} = \tilde{\mathbf{X}}^{(1)} + \tilde{\mathbf{X}}^{(2)}$ . Then the final result is  $\mathbf{X} = \tilde{\mathbf{X}}^{(1)} + \tilde{\mathbf{X}}^{(2)} + \tilde{\mathbf{X}}^{(3)}$  and, after the diagonal averaging,  $\mathbf{X} = \tilde{\mathbf{X}}^{(1)} + \tilde{\mathbf{X}}^{(2)} + \tilde{\mathbf{X}}^{(3)}$ . There is no need for reconstruction of the signal by Basic SSA as only the estimated signal subspace is used for making a refined decomposition.

## Features of decompositions

The result of Decomposition step of SSA (Step 2) can be written in the form (1.2). The SVD is a particular case of (1.2) and corresponds to the orthonormal systems of  $\{U_i\}$  and  $\{V_i\}$ . By analogy with the SVD, we will call  $(\sigma_i, U_i, V_i)$  eigentriples,  $\sigma_i$  singular values,  $U_i$  left singular vectors or eigenvectors,  $V_i$  right singular vectors or factor vectors. For most of SSA decompositions, each  $U_i$  belongs to the column space of  $\mathbf{X}$  while each  $V_i$  belongs to the row space of  $\mathbf{X}$ . We shall call such decompositions *consistent*.

If the systems  $\{U_i\}$  and  $\{V_i\}$  are linearly independent, then the decomposition (1.2) is *minimal*; that is, it has smallest possible number of addends equal to  $r = \text{rank } \mathbf{X}$ . If at least one of the systems  $\{U_i\}$  or  $\{V_i\}$  is not linearly independent, then the decomposition (1.2) is not minimal. If the decomposition (1.2) is not consistent, then it can be not minimal even if  $\{U_i\}$  or  $\{V_i\}$  are linearly independent, since their projections on the column (or row) space can be dependent.

If both vector systems  $\{U_i\}$  and  $\{V_i\}$  are orthogonal, then the decomposition (1.2) is called biorthogonal. If  $\{U_i\}$  is orthogonal, then the decomposition is called left-orthogonal; if  $\{V_i\}$  is orthogonal, then the decomposition is called right-orthogonal.

If  $\mathbf{X}_i$  are F-orthogonal so that  $\langle \mathbf{X}_i, \mathbf{X}_j \rangle_F = 0$  for  $i \neq j$ , then we say that the corresponding decompositions are F-orthogonal. Either left or right orthogonality is sufficient for F-orthogonality. For F-orthogonal decompositions (1.2),  $\|\mathbf{X}\|_F^2 = \sum_i \|\mathbf{X}_i\|_F^2$ . In general, however,  $\|\mathbf{X}\|_F^2$  may differ from  $\sum_i \|\mathbf{X}_i\|_F^2$ .

The contribution of  $k$ th matrix component  $\mathbf{X}_k$  is defined as  $\sigma_k^2/\|\mathbf{X}\|_F^2$ , where  $\sigma_k^2 = \|\mathbf{X}_k\|_F^2$ . For F-orthogonal decompositions, the sum of component contributions is equal to 1. Otherwise, this sum can considerably differ from 1.

## Decompositions in different versions of SSA

Let us gather several versions of 1D-SSA which are implemented in the RSSA package and based on different procedures used at Decomposition step, and indicate their features. Some of these variations are also implemented for multivariate and multidimensional versions of SSA.

Basic SSA: the conventional SVD, consistent, minimal, biorthogonal and therefore F-orthogonal decomposition.

Toeplitz SSA: generally, non-consistent, non-minimal F-orthogonal decomposition.

SSA with projection: F-orthogonal but non-consistent decomposition if at least one basis vector used for the projection does not belong to the column (row) trajectory space. The components, which are obtained by projections, are located at the beginning of the decomposition and have indices  $1, \dots, n_{\text{special}}$ .

Oblique SSA with filter preprocessing (Filter-adjusted O-SSA): consistent, minimal F-orthogonal decomposition. The main particular case is DerivSSA.

Iterative Oblique SSA (Iterative O-SSA): consistent, minimal oblique decomposition.

Oblique versions of SSA are made to perform in a nested manner.

### 1.3.2. Versions of SSA dealing with different forms of the object

In this section, we briefly consider versions of SSA which operate objects  $\mathbf{X}$  of different forms. The main difference between the versions of SSA of this section is the form of the embedding operator  $\mathcal{T}$ .

As has been mentioned above, SSA can be applied to multivariate and multidimensional objects. SSA for a system of series is called Multivariate (or Multichannel) SSA, shortly MSSA (Section 4.1); SSA for digital gray-scale images is called 2D-SSA (Section 5.2.3).

Shaped SSA (Sections 5.1 and 5.2.1) can process data of complex structure and arbitrary shape; the dimension of the object  $\mathbf{X}$  is irrelevant. Shaped SSA can be applied to many different kinds of data including time series, systems of time series, digital images of rectangular and non-rectangular shapes.

For both series and images, circular versions of SSA are available. For series, circular SSA works in the metric of a circle and therefore this version is suitable for series, which are indeed defined on a circle.

Table 1.1: Classification of different versions of SSA based on different geometrical features of the object  $\mathbf{X}$ .

Method	Data	Notation	Trajectory matrix
1D-SSA	time series (t.s.)	$\mathbf{X} = (x_1, \dots, x_N)$ of length $N$	Hankel
MSSA	system of t.s.	$\mathbf{X}^{(p)}$ of length $N_p$ , $p = 1, \dots, s$	Stacked Hankel
2D-SSA	rectangular image	$\mathbf{X} = (x_{ij})_{i,j=1}^{N_x, N_y}$	Hankel-Block-Hankel
Shaped SSA	shaped objects	$\mathbf{X} = (x_{(i,j)})_{(i,j) \in \mathfrak{N}}$	Quasi-Hankel
Circular SSA	circular/cylindrical	objects on circle or cylinder	Quasi-Hankel

For images, circular versions of SSA provide a possibility to decompose images given on a cylinder (for example, obtained as a cylindrical projection of a sphere or an ellipsoid) or given on a torus. Circular versions allow to eliminate the edge effects, which are unavoidable in the case of e.g. planar unfolding of a cylinder.

Table 1.1 contains a list of the extensions considered in this book.

## 1.4. Separability in SSA

In this section, we discuss the SSA separability in more detail; see also Sections 1.8.3, 2.1.2, 2.2.1, 2.3.1 for special cases.

Let us assume that we observe a sum of two objects  $\mathbf{X} = \mathbf{X}^{(1)} + \mathbf{X}^{(2)}$ . We say that SSA separates these two objects if a grouping at Grouping step (Step 3) can be found such that  $\tilde{\mathbf{X}}^{(k)} = \mathbf{X}^{(k)}$  for  $k = 1, 2$ . If these equalities hold approximately, then this defines approximate separability. Asymptotic separability can be introduced if the series length  $N \rightarrow \infty$ . In this case, approximate separability takes place for large enough series lengths. The separability property is very important for SSA as it means that the method potentially works; that is, it is able to extract the object components.

In Basic SSA and its multidimensional extensions, (approximate) separability means (almost) orthogonality of the object components, since the biorthogonal SVD decomposition is used. In other versions of SSA, different conditions of separability can be formulated.

If the decomposition (1.2) at Decomposition step is not unique, then two variants of separability are introduced. Weak separability means that there exists a decomposition such that after grouping we obtain  $\tilde{\mathbf{X}}^{(k)} = \mathbf{X}^{(k)}$ . Strong separability means that this equality is achievable for any admissible decomposition.

Conditions of exact separability are very restrictive whereas asymptotic separability takes place for a wide range of object components. For example, slowly varying smooth components are asymptotically separable from regular oscillations and they both are asymptotically separable from noise.

In order to verify separability of the reconstructed components  $\tilde{\mathbf{X}}_1$  and  $\tilde{\mathbf{X}}_2$ , we should check orthogonality of their reconstructed trajectory matrices  $\tilde{\mathbf{X}}_1$  and  $\tilde{\mathbf{X}}_2$ . A convenient measure of their

orthogonality is the Frobenius inner product  $\langle \tilde{\mathbf{X}}_1, \tilde{\mathbf{X}}_2 \rangle_F$  defined in (1.7).

The normalized measure of orthogonality is

$$\rho(\tilde{\mathbf{X}}_1, \tilde{\mathbf{X}}_2) = \langle \tilde{\mathbf{X}}_1, \tilde{\mathbf{X}}_2 \rangle_F / (\|\tilde{\mathbf{X}}_1\|_F \|\tilde{\mathbf{X}}_2\|_F).$$

Since the trajectory matrix consists of  $w_\tau = |\mathcal{A}_\tau| = \|\mathcal{T}(\mathbf{e}_\tau)\|_F^2$  entries of the  $\tau$ th element  $x_\tau$  of the initial ordered object, we can introduce the weighted inner product in the space  $\mathcal{M}$ :  $(Y, Z)_w = \sum_\tau w_\tau y_\tau z_\tau$ . Then the quantity

$$\rho_w(\tilde{\mathbf{X}}_1, \tilde{\mathbf{X}}_2) = \rho(\tilde{\mathbf{X}}_1, \tilde{\mathbf{X}}_2) = \frac{(\tilde{\mathbf{X}}_1, \tilde{\mathbf{X}}_2)_w}{\|\tilde{\mathbf{X}}_1\|_w \|\tilde{\mathbf{X}}_2\|_w} \quad (1.8)$$

will be called **w-correlation** by statistical analogy. Note however that in this definition the means are not subtracted.

Let  $\tilde{\mathbf{X}}_j$  be the elementary reconstructed components produced by the elementary grouping  $I_j = \{j\}$ . Then the matrix of  $\rho_{ij}^{(w)} = \rho_w(\tilde{\mathbf{X}}_i, \tilde{\mathbf{X}}_j)$  is called **w-correlation matrix**.

The norm  $\|\cdot\|_w$  is called the *weighted norm* and serves as a measure of contribution of the components in the decomposition (1.5): the contribution of  $\tilde{\mathbf{X}}_k$  is defined as  $\|\tilde{\mathbf{X}}_k\|_w^2 / \|\mathbf{X}\|_w^2$ .

If the weighted correlation between a pair of elementary components is large, then this suggests that these two components are highly correlated and should perhaps be included into the same group.

## 1.5. Forecasting, interpolation, low-rank approximation and parameter estimation in SSA

There is a class of objects, which is special for SSA. This is the class of objects satisfying linear recurrence relations. Trajectory matrices of these objects are rank-deficient and, moreover, for these objects the number of non-zero terms in the SVD (1.3) does not depend on the window length if this length is sufficiently large; we will say in such cases that the objects are of finite rank. The class of objects satisfying linear recurrence relations provides a natural model of the signal for SSA, which is a fundamentally important concept for forecasting of time series.

Linear recurrence relation (LRR) for a time series  $S_N = (s_i)_{i=1}^N$  is a relation of the form

$$s_{i+t} = \sum_{k=1}^t a_k s_{i+t-k}, \quad 1 \leq i \leq N-t, \quad a_t \neq 0, \quad t < N. \quad (1.9)$$

It is well-known (see e.g. [46, Theorem 3.1.1]) that a sequence  $S_\infty = (s_1, \dots, s_n, \dots)$  satisfies the LRR (1.9) for all  $i \geq 0$  if and only if for some integer  $p$  we have for all  $n$

$$s_n = \sum_{k=1}^p P_k(n) \mu_k^n, \quad (1.10)$$

where  $P_k(n)$  are polynomials in  $n$  and  $\mu_k$  are some complex numbers. For real-valued time series, (1.10) implies that the class of time series governed by the LRRs consists of sums of products of polynomials, exponentials and sinusoids.

A simplified model of (1.10) is  $s_n = \sum_{k=1}^p C_k \mu_k^n$ . Estimation of complex numbers  $\mu_k = \rho_k e^{i2\pi\omega_k}$  is equivalent to estimating the frequencies  $\omega_k$  and the rates  $\ln \rho_k$ .

For images  $S = [s_{mn}]$ , LRRs are two-dimensional and the common term of the signal (which can be called pattern) under the model has the form

$$s_{mn} = \sum_{k=1}^p P_k(m, n) \mu_k^m v_k^n. \quad (1.11)$$

This important fact is well known starting from [47, §2.20].

In many real-life problems, a noisy signal (or noisy pattern for images) is observed and the problem is to forecast the signal, impute gaps in the signal, estimate signal parameters, find change-points in the signal, and so on. Note that it is not compulsory to assume that the noise is random. In a very general sense, noise is a residual which does not require further investigation.

SSA may provide estimates of the signal space, which is the subspace spanned by the chosen basis during Grouping step in SSA. The estimation of the signal subspace can be performed by iterations (so-called Cadzow iterations), which consist of iterative SSA processing, see Section 1.9.1.

In the process of estimation of the signal subspace we also obtain a parametric estimate the signal; that is, of the set of  $\{\mu_k\}$  in the 1D case and the set of  $\{\mu_k, v_k\}$  in the 2D case. One of the most common methods is called ESPRIT for series (see Section 1.9.2) and its extension 2D-ESPRIT for images [41].

For time series, we can forecast the found structure, e.g., by application of the constructed LRR. By performing interpolation (which is forecasting inside the series) missing data can be filled-in.

## 1.6. Comparison of SSA with other methods

In this section, we provide some notes and a short bibliographical overview concerning comparison of SSA with several traditional methods [20].

### 1.6.1. Fourier transform, filtering, noise reduction

- Fourier Transform uses a basis given in advance, while SSA uses an adaptive basis, which is not restricted to a frequency grid with resolution  $1/N$ . The wavelet transform also uses fixed bases; the advantage of the wavelet transform is that a change in the frequencies can be detected by the used time-space basis. In the framework of SSA, the analysis of time series with changing frequency structure can be performed by using moving procedures, e.g., by subspace tracking.

- One of the state-of-art methods of frequency estimation is the high-resolution method ESPRIT, which is a subspace-based method. This method can be considered as an SSA-related method and indeed it is frequently used in the present book and in RSSA.
- Fourier Transform is very inefficient for series with modulations in amplitudes and frequencies. SSA can easily deal with amplitude modulation but cannot efficiently deal with frequency modulation.
- SSA decomposition can sometimes be viewed as an application of a set of linear filters [48, 49, 29] with an interpretation depending on the window length  $L$ . For small  $L$ , each decomposition component on the interval  $[L, K]$ , where  $K = N - L + 1$ , can be obtained by a linear filter. Therefore, the viewpoint of filtering on the decomposition result can be adequate. For example, the reconstruction by the leading components is close to application of the triangle filter.

If  $L \simeq K$  and hence the interval  $[L, K]$  is small, then it is not so. In this case, the separability approach, which is based on orthogonality of separated components, is more appropriate. Note that oblique versions of SSA can weaken the condition of orthogonality, see Section 2.2.

- There is a big difference between the moving averaging and SSA for noise reduction. Consider an example of a noisy sinusoid. The moving averaging will add a bias in estimation caused by the second derivative of the signal, while SSA with large  $L$  will provide a unbiased estimate of the signal.

Note that even for small  $L$ , when the reconstruction by the leading component is a weighted moving average with positive weights and therefore has the same drawbacks as the moving averaging, the user can add additional components to remove the possible bias.

- Filtering by SSA to obtain noise reduction can be considered from the viewpoint of the low-rank approximation. The good approximation properties yield appropriate noise suppression. Empirical-mode decomposition (EMD), in turn, starts Intrinsic Mode Functions (IMF) with high frequencies, while the trend is contained in the last IMFs.

As an example of comparison of SSA, Fourier transform and wavelet transform see e.g. [50]. The authors conclusion states: 'the SSA-based filtering technique is robust for regional gravity anomaly separation and could be effectively exploited for filtering other geophysical data'.

In [51], an SSA-based de-noising technique for removal of electrocardiogram interference in Electromyography signals is compared with the high-pass Butterworth filter, wavelets and EMD. The authors of that paper state: 'the proposed SSA approach is a valid method to remove the ECG artifact from the contaminated EMG signals without using an ECG reference signal.'

In [52], many different methods for trend extraction are compared for synthetic data simulating sea level behaviour; SSA is compared against moving average, wavelets, regression, EMD. The

author writes: ‘the optimum performing analytic is most likely to be SSA whereby interactive visual inspection (VI) techniques are used by experienced practitioners to optimise window length and component separability’.

Comparison of SSA filtering and Kalman filters (KF) can be found in [53], where it is shown that ‘both SSA and KF obtain promising results from the stations with strong seasonal signals, while for the stations dominated by the long-term variations, SSA seems to be superior’.

### 1.6.2. Parametric regression

Parametric regression naturally assumes a parametric model. There is a big difference between parametric and non-parametric models: if the assumed model is true, then the related parametric methods are the most appropriate methods (if there are no outliers in the data). If the assumed parametric model is not true, then the results of parametric methods are biased and may be very misleading. Drawbacks of non-parametric methods are also clear: there are problems with forecasting, testing the model, confidence interval construction, and so on. Frequently, non-parametric methods serve as preprocessing tools for parametric methods. As discussed in Section 1.7.2, this is often the case for SSA.

For comparison of SSA with double centering and linear regression see Section 2.1. It appears that SSA with double centering as preprocessing method considerably improves the accuracy of linear trend estimation.

SSA has a very rare advantageous property: it can be a non-parametric method for preliminary analysis and can also be parametric for modeling the series governed by LRRs. Moreover, the forecast by an LRR uses the parametric model in implicit manner; therefore, it is more robust to deviations from the model than the forecast based on explicit parameter estimation.

One of the subspace-based method for constructing the model of the signal, which is governed by a LRR, is HSLRA. HSLRA can be considered as a method of parameter estimation in a parametric model, where only the rank of the signal is given rather than exact parametric form, see Section 3.2.

### 1.6.3. ARIMA and ETS

First, the (Seasonal) ARIMA and Exponential smoothing models (ETS, which means Error, Trend, Seasonal) totally differ from the model of SSA (for a comprehensive introduction to ARIMA and ETS see [54]). In particular, in ARIMA the noise is added at each recurrence step, while for SSA the noise is added after the signal is formed. Also, trends/seasonality in SSA are deterministic, while in ARIMA/ETS the trends/seasonality are random. As in many classical methods, ARIMA and ETS need the period values to be specified for the periodic components.

However, if one considers the analysis/forecast of real-life time series, then these time series do not exactly follow any model. Therefore, the problem of comparison of methods of different nature is not easy.

As a rule, confidence intervals for ARIMA forecasts are too large but the mean forecast can often be adequate. Advantage of Seasonal ARIMA and ETS is that the model and its parameters can be fitted automatically on the base of information criteria.

Rigorously substantiated information criteria are not constructed for SSA. One of the reasons for this is the fact that SSA is a non-parametric method. The most standard approach for the choice of parameters, when there are no given models, is the minimization of the forecasting error on the validation period. In the most frequent case, when the forecast is constructed on the base of  $r$  leading eigentriples, SSA has only two parameters ( $L$  and  $r$ ), which can be estimated by the minimization of the forecasting error for several forecasts performed within the validation period.

Comparison of SSA and ARIMA/ETS was performed in many papers. Some examples are as follows.

- It is demonstrated in [55] that SSA has topped several other methods in an example involving forecasting of tourist arrivals,
- It is exhibited in [56] that for predicting ambulance demands ‘SSA produces superior longer-term forecasts (which are especially helpful for EMS planning), and comparable shorter-term forecasts to well established methods’.
- The author of [57] concludes: ‘The forecasting results are compared with the results of exponential smoothing state space (ETS) and ARIMA models. The three techniques do similarly well in forecasting process. However, SSA outperforms the ETS and ARIMA techniques according to forecasting error accuracy measures.’
- In [58, 59], the univariate and multivariate SSA were favorable in a comparison with ARIMA and VAR for forecasting of several series of European industrial production.

## 1.7. Bibliographical notes

**References** The origin of singular spectrum analysis (SSA) is usually referred to the papers [60] and [61]; although, the algorithm of SSA can be found e.g. in [62]. SSA became widely known in climatology after publication of [63, 64]. After several years, the book [65] summarized the basic information about SSA existing to that moment. In parallel, SSA (named ‘Caterpillar’) was created in Russia; the results were published in [66] (in Russian). The history of the ‘Caterpillar’ method starts from [67], where O. M. Kalinin is indicated as the author of the ideas underlying the method (it is difficult to find the access to this review; therefore, we refer to the book [68, Chapter 3, Section 8], where this review is cited together with the algorithm of the first stage of SSA). Another source of the Russian branch was the paper [69].

A breakthrough in the theory of SSA was made in the fundamental book [1], where the theory is presented together with examples. The next book is [29] in the series Briefs in Statistics; it contains a brief description and some updates from 2001, including description of SSA as a set

of filters. From 2013, a large jump was performed, when SSA became a method for analysis of objects of different dimensions and shapes. Also, the R-package RSSA [70] was developed with a very fast implementation of SSA for different kinds of objects. The proposed structured approach to SSA, its multivariate extensions (MSSA and 2D-SSA) together with algorithms and description of the implementation in RSSA are contained in the recent book [30].

Three mentioned monographs of Golyandina and coauthors cover a very wide range of problems solved by SSA; however, they only briefly discuss practical applications of SSA to stationary processes. At the same time, the applications of SSA to stationary time series were developed by the team from UCLA (starting from [71]), mostly for climatic data. Some practical applications, in particular, in economics and biomedicine, are considered in the works of H.Hassani, S.Sanei and their coauthors (see, e.g., the book [72] and the review [73]). A separate branch is related to real-world problems in geophysics, where traces in the form of straight lines should be extracted; a preliminary processing is performed by the discrete Fourier transform of the image rows and then the complex-valued version of SSA is applied to the Fourier coefficients [74, 75]. These branches seem to be developing somewhat independently; therefore, it would be very helpful to enrich one another.

**Sketch of the algorithm** The SSA algorithm consists of two stages. The first stage is called Decomposition, where the studied object (e.g. a time series) is transformed into a trajectory matrix (a Hankel matrix) and then the singular value decomposition is applied to the trajectory matrix to obtain a decomposition into elementary rank-one matrix components. The second stage, which is called Reconstruction, creates grouped matrix components in a clever way and transforms the grouped matrix decomposition back to a decomposition of the initial object by the so-called diagonal averaging.

Since the idea of considering moving subseries of one time series as different observations and then applying principal component analysis (PCA) or Karhunen-Loëve transform (KLT) to the obtained sample is straightforward, Decomposition stage of SSA can be found in many papers; it is hard to detect which paper was first. For example, the mentioned above papers [67, 60, 61, 63] contain description only of Decomposition stage. Also, we can cite [76] and [77] as references related to the first stage of SSA. ‘Diagonal averaging’ from the second stage is used in [62, 78, 3, 79, 69]. Nowadays, Reconstruction stage is considered as an essential part of SSA.

Another origin of SSA traces back to properties of Hankel matrices [80]. Sometimes, an origin of SSA is drawn from [81], where the modelling of the signal in the form of a sum of exponential series was considered; this origin is related to the parameter (frequency) estimation problem.

We suggest to call the method ‘SSA’ if both Decomposition and Reconstruction stages are involved. The methods based on Decomposition stage only are called subspace-based methods. Although many subspace-based methods were developed before SSA, these methods may be called SSA-related.

### 1.7.1. Some recent applications of SSA

The number of publications devoted to applications of SSA is steadily increasing. In addition to the standard applications areas such as climatology, meteorology and geophysics, there are now many papers devoted to applications in engineering, economics, finance, biomedicine and other areas. One can find many references to recent publications in [82] and many papers in the two special issues of *Statistics and Its Interface* (2010, v.3, No.3 and 2017, v.10, No.1), which are either fully or partly devoted to SSA. In this short section we briefly mention some recent applications of SSA. In most of these papers, only the simplest versions of SSA (that is, Basic SSA of Section 1.8 and Toeplitz SSA of Section 1.8.6) have been used.

Advantages of 2D-SSA (described in Section 5.2.3) over some other methods of image processing are demonstrated in [83, 84] in application to hyperspectral imaging. Application of 2D-SSA to gap-filling is considered in [85]. Application of Multivariate SSA for detecting oscillator clusters in multivariate datasets is proposed in [86].

It is not easy to find applied areas related to analysis of temporal data, where 1D-SSA was not applied. Let us give some examples. In [87] and several other papers of the same authors, SSA has been used as the main technique in the development of a tool-wear monitoring system. Security of mobile devices is considered in [88], where SSA is used for preprocessing. In [89], SSA was used for extraction of economic cycles. [90] uses SSA for gap filling of precipitation data. Some recent applications in climatology were considered in [91, 92] and in [93]. In [94], SSA helps to solve the problem of unauthorized modification in speech signals. In [51], SSA is used for de-noising in the problem of removal of electrocardiogram interference in Electromyography signals. The paper [95] is related to the decomposition and reconstruction of long-term flowering records of eight eucalypt species. In [96], SSA was used as a preprocessing tool prior to making a classification of a medical data; the authors wrote: ‘the results have demonstrated the robustness of the approach when testing on large scale datasets with clinically acceptable sensitivity and specificity’.

### 1.7.2. SSA for preprocessing / combination of methods

For many different methods, SSA provides improvement if it is used as a preprocessing tool. There are dozens of papers, where hybrid methods incorporating SSA are considered. In the most of applications, SSA serves for either denoising or feature extraction. Let us give some examples of papers considering hybrids of SSA and other methods.

SSA is used as preprocessing for ARIMA in [97]. A cooperative hybrid of SSA, ARIMA, and Holt-Winters is suggested in [98]. In [99] it is shown that the hybrid SSA + ARMAX is better than ARMAX alone for detection of structural damages for problems of Structural Health Monitoring.

In machine learning, SSA is frequently used to obtain new characteristics of time series for a subsequent use of them in other models and methods. This is called feature extraction. The paper [100] is considered as a one of the first papers, where SSA is used together with Support Vector Machines (SVM). A hybrid of SSA with Neural Networks was suggested in [101].

In [102], support vector machine regression (SVR) is applied separately to the trend and fluctuations, which are extracted by SSA. The constructed method is applied to forecast a time series data of failures gathered at the maintenance stage of the Boeing 737 aircraft. It is shown that the suggested hybrid SSA+SVR outperforms Holt-Winters, ARIMA, multiple linear regression, group method of data handling, SSA, and SVR used separately. Similar techniques are considered in [103], where SSA is employed for extraction of the trend and seasonality and then Neural Networks and fuzzy logic are applied to them separately with consequent combination. In [104], SSA is successfully used for noise removal before Neural Networks are applied. This work contains a review of different approaches to Rainfall-runoff modeling by means of SSA used in combination with other methods.

In [83], SSA has been applied in hyperspectral imaging for effective feature extraction (noise removal), and then SVM was used for classification. It appeared that SSA performed preprocessing better than Empirical Mode Decomposition (EMD). Note that SSA and EMD do not only compete; they can be successful as hybrids. For example, in [105] EMD is used for trend extraction and then SSA is applied to forecast changes in the trend.

## 1.8. Basic SSA for one-dimensional time series

Basic SSA is the SSA for the analysis of one-dimensional series such that the decomposition into rank-one matrices at Step 2 (see the general scheme in Figure 1.3) is done by the SVD.

### 1.8.1. Method

#### Step 1: Embedding

The series  $\mathbf{X}$  is mapped to a sequence of  $L$ -lagged vectors, which form the trajectory matrix  $\mathbf{X} = \mathcal{T}_{\text{SSA}}(\mathbf{X})$ , as shown in (1.1).

#### Step 2: Decomposition

Set  $\mathbf{S} = \mathbf{X}\mathbf{X}^T$  and denote by  $\lambda_1, \dots, \lambda_d$  the positive *eigenvalues* of  $\mathbf{S}$  taken in the decreasing order of magnitude ( $\lambda_1 \geq \dots \geq \lambda_d > 0$ ) and by  $U_1, \dots, U_d$  an orthonormal system of the *eigenvectors* of the matrix  $\mathbf{S}$  corresponding to these eigenvalues;  $V_i = \mathbf{X}^T U_i / \sqrt{\lambda_i}$  are called factor vectors. At this step, we perform the singular value decomposition (SVD) of the trajectory matrix:

$$\mathbf{X} = \sum_{i=1}^d \sqrt{\lambda_i} U_i V_i^T = \mathbf{X}_1 + \dots + \mathbf{X}_d. \quad (1.12)$$

The matrices  $\mathbf{X}_i = \sqrt{\lambda_i} U_i V_i^T$  in (1.12) have rank 1; such matrices are called *elementary matrices*. The collection  $(\sqrt{\lambda_i}, U_i, V_i)$  consisting of the singular value  $\sqrt{\lambda_i}$ , the left singular vector  $U_i$  and the right singular vector  $V_i$  will be called *i*th *eigentriple* (abbreviated as ET). Note that  $\lambda_i = \|\mathbf{X}_i\|_F^2$

and  $\|\mathbf{X}\|_{\text{F}}^2 = \|\mathbf{X}_1\|_{\text{F}}^2 + \dots + \|\mathbf{X}_d\|_{\text{F}}^2$ . The contribution of  $i$ th component  $\mathbf{X}_i$  can thus be measured by  $\lambda_i / \sum_j \lambda_j$ .

For real-world time series,  $d = \text{rank } \mathbf{X}$  is typically equal to  $\min(L, K)$ ; that is, the trajectory matrix is of full rank.

### Step 3: Eigentriple grouping

The input in this step is the expansion (1.12) and the specification of how to group the components of this expansion.

Let  $I = \{i_1, \dots, i_p\} \subset \{1, \dots, d\}$  be a set of indices. Then the resultant matrix  $\mathbf{X}_I$  corresponding to the group  $I$  is defined as  $\mathbf{X}_I = \mathbf{X}_{i_1} + \dots + \mathbf{X}_{i_p}$ .

Assume that a partition of the set of indices  $\{1, \dots, d\}$  into  $m$  disjoint subsets  $I_1, \dots, I_m$  is specified. Then the expansion (1.12) leads to the grouped decomposition

$$\mathbf{X} = \mathbf{X}_{I_1} + \dots + \mathbf{X}_{I_m}. \quad (1.13)$$

The procedure of choosing the sets  $I_1, \dots, I_m$  is called *eigentriple grouping*. If  $m = d$  and  $I_j = \{j\}$ ,  $j = 1, \dots, d$ , then the corresponding grouping is called *elementary*.

The grouping is performed by analyzing the eigentriples so that each group corresponds to an identifiable series component. The choice of several leading eigentriples corresponds to an optimal approximation of the time series, in accordance with the well-known optimality property of the SVD.

### Step 4: Reconstruction (Diagonal averaging)

The diagonal averaging (1.6) applied to a resultant matrix  $\mathbf{X}_{I_k}$  produces a *reconstructed series*  $\tilde{\mathbf{X}}^{(k)} = (\tilde{x}_1^{(k)}, \dots, \tilde{x}_N^{(k)}) = \mathcal{T}^{-1} \circ \Pi_{\mathcal{H}}(\mathbf{X}^{(k)})$ . This way, the initial series  $(x_1, \dots, x_N)$  is decomposed into a sum of  $m$  reconstructed series:

$$x_n = \sum_{k=1}^m \tilde{x}_n^{(k)}, \quad n = 1, \dots, N. \quad (1.14)$$

The reconstructed series produced by the elementary grouping will be called *elementary reconstructed series*.

If grouping is sensible, then we obtain a reasonable decomposition into identifiable series components. Typical resultant decompositions are signal plus noise or trend plus seasonality plus noise.

As well as in the generic scheme, Steps 1 and 2 of Basic SSA are sometimes combined into the so-called ‘Decomposition stage’ and Steps 3 and 4 are combined into ‘Reconstruction stage’.

### 1.8.2. Model of time series

#### Time series of finite rank

Although the SSA method is model-free and therefore SSA can be considered as an exploratory method, there is a model that perfectly suits SSA.

We say that a series has  $L$ -rank  $r$  if its  $L$ -trajectory matrix has rank  $r$ . Series  $S$  is called a series of finite rank  $r$  ( $\text{rank } S = r$ ) if  $\text{rank}_L S = r$  for any sufficiently large series length  $N$  and window length  $L$ . The term ‘finite rank’ also has a meaning for the case of infinite series. For a generic infinite times series (cut at some  $N$ ),  $L = L(N)$  can tend to infinity and in this case the rank of the trajectory matrix would typically tend to infinity too. For a time series of finite rank, the rank  $r$  of the trajectory matrix is finite and fixed for large enough  $N$  and  $L$  such that  $r \leq \min(L, N - L + 1)$ .

It is useful to know rank of a time series and the form of the singular vectors of its trajectory matrix, since knowing rank means knowing the number of the SVD components, which we should group for extraction of the corresponding series component, while the form of the singular vectors along with properties of eigenvalues helps in finding these SVD components.

We refer to [1, Chapter 5] for details and full description. Here we mention that an exponential series  $s_n = Ae^{\alpha n}$ ,  $n = 1, 2, \dots$ , has rank 1, a linear function  $s_n = an + b$ ,  $a \neq 0$ , has rank 2, a sinusoid with  $s_n = A \sin(2\pi\omega n + \phi)$  has rank 2 for  $0 < \omega < 0.5$  and rank 1 for  $\omega = 0.5$ . Singular vectors of trajectory matrices of time series have the same form as the series itself, which follows from the fact that rows and columns of the trajectory matrices are subseries of the original series. This information helps in choosing the groups at Grouping step.

#### Linear recurrence relations, characteristic polynomials and roots

Time series of finite rank are closely related to the series governed by linear recurrence relations (LRRs). In particular, for infinite time series, the class of time series governed by LRRs coincides with the class of time series of finite rank.

**Definition 1.** A time series  $S_N = (s_i)_{i=1}^N$  is governed by an LRR, if there exist  $a_1, \dots, a_t$  such that

$$s_{i+t} = \sum_{k=1}^t a_k s_{i+t-k}, \quad 1 \leq i \leq N-t, \quad a_t \neq 0, \quad t < N-1. \quad (1.15)$$

The number  $t$  is called the order of the LRR and  $a_1, \dots, a_t$  are the coefficients of the LRR. If  $t = r$  is the minimal order of an LRR that governs the time series  $S_N$ , then the corresponding LRR is called minimal.

The minimal LRR is unique and its order is equal to the series rank.

As was mentioned in Section 1.5, it is well-known that the time series  $S_\infty = (s_1, \dots, s_n, \dots)$

satisfies the LRR (1.15) for all  $i \geq 0$  if and only if

$$s_n = \sum_{m=1}^p \left( \sum_{j=0}^{k_m-1} C_{mj} n^j \right) \mu_m^n, \quad (1.16)$$

where the complex coefficients  $C_{mj}$  depend on the first  $t$  points  $s_1, \dots, s_t$ .

For real-valued time series, (1.16) implies that the class of time series governed by the LRRs consists of sums of products of polynomials, exponentials and sinusoids

$$s_n = \sum_{m=1}^{\tilde{p}} \left( \sum_{j=0}^{k_m-1} \tilde{c}_{mj} n^j \right) e^{\alpha_m n} \cos(2\pi\omega_m n + \phi_m), \quad 0 \leq \omega_m \leq 0.5. \quad (1.17)$$

The minimal LRR determines all, except  $C_{mj}$ , parameters in (1.16) and all, except  $\tilde{C}_{mj}$  and  $\phi_m$ , parameters in (1.17).

**Definition 2.** *The polynomial  $P_t(\mu) = \mu^t - \sum_{k=1}^t a_k \mu^{t-k}$  is called characteristic polynomial of the LRR (1.15).*

Roots of the characteristic polynomial are called *characteristic roots* of the corresponding LRR. The roots of the characteristic polynomial of the minimal LRR governing the series, which can be called *signal roots of the LRR* or *characteristic roots of the series*, determine the values of parameters  $\mu_m$  and  $k_m$  in (1.16) as follows. Let the time series  $S_\infty = (s_1, \dots, s_n, \dots)$  satisfy the LRR (1.15) with  $a_t \neq 0$  and  $i \geq 1$ . Consider the characteristic polynomial of the LRR (1.15) and denote its different (complex) roots by  $\mu_1, \dots, \mu_p$ , where  $p \leq t$ . All these roots are non-zero as  $a_t \neq 0$  with  $k_m$  being the multiplicity of the root  $\mu_m$  ( $1 \leq m \leq p$ ,  $k_1 + \dots + k_p = t$ ).

Let  $C_{k_p-1,j} \neq 0$  for all  $j$ ; this corresponds to the case of the minimal LRR. Then the rank of time series  $S_\infty$  given in (1.16) is equal to  $r = \sum_{m=1}^p k_m$ . In the real-valued case, if  $\tilde{c}_{k_p-1,j} \neq 0$  for all  $j$ , then the rank of time series  $S_\infty$  given in (1.17) is equal to  $r = \sum_{m=1}^{\tilde{p}} \delta_m k_m$ , where  $\delta_m = 1$  for  $\omega_m$  equal 0 or 0.5 and  $\delta_m = 2$  for  $0 < \omega_m < 0.5$ .

If we find the signal roots  $\mu_m = \rho_m e^{\pm i 2\pi\omega_m}$  of the characteristic polynomial of the LRR governing the signal, then we can estimate the signal parameters. For example, the frequency  $\omega_m$  of an exponentially-modulated sinusoid can be found using the argument of the corresponding conjugate roots, whereas the root modulus  $\rho_m$  gives the exponential rate  $\alpha_m = \ln \rho_m$ .

### 1.8.3. Separability and choice of parameters

Understanding separability is very important for understanding how SSA works. Recall that if two time series  $X_N^{(1)}$  and  $X_N^{(2)}$  are separable, then  $X_N^{(1)}$  can be extracted from the observed series  $X_N = X_N^{(1)} + X_N^{(2)}$ . This means that there exists a partition into groups at Grouping step such that  $\tilde{X}_N^{(m)} = X_N^{(m)}$ .

Let us define the separability formally. Let  $\mathbf{X}^{(m)}$  be the trajectory matrices of the considered

series components,  $\mathbf{X}^{(m)} = \sum_{i=1}^{d_m} \sqrt{\lambda_{m,i}} U_{m,i} V_{m,i}^T$ ,  $m = 1, 2$ , be their SVDs. The column and row spaces of the trajectory matrices are called *column* and *row trajectory spaces* correspondingly.

**Definition 3.** Let  $L$  be fixed. Two series  $\mathbf{X}_N^{(1)}$  and  $\mathbf{X}_N^{(2)}$  are called weakly separable (or simply separable) if their column trajectory spaces are orthogonal and the same is valid for their row trajectory spaces; that is,  $(\mathbf{X}^{(1)})^T \mathbf{X}^{(2)} = \mathbf{0}_{K,K}$  and  $\mathbf{X}^{(1)} (\mathbf{X}^{(2)})^T = \mathbf{0}_{L,L}$ .

**Definition 4.** Two series  $\mathbf{X}_N^{(1)}$  and  $\mathbf{X}_N^{(2)}$  are called strongly separable, if they are weakly separable and the sets of singular values of their  $L$ -trajectory matrices are disjoint; that is,  $\lambda_{1,i} \neq \lambda_{2,j}$  for any  $i$  and  $j$ .

Weak separability means that at Decomposition step there exists such an SVD that allows the proper grouping. A possibility of a non-separating SVD expansion which does not allow a proper grouping is related to the non-uniqueness of the SVD in the case of equal singular values. Strong separability means that any SVD of the trajectory matrix admits the proper grouping. Therefore, in order to be sure that SSA makes an accurate separation we have to require strong (approximate) separability.

By the definition, weak separability means orthogonality of the column and row spaces of the trajectory matrices of the series components  $\mathbf{X}_N^{(1)}$  and  $\mathbf{X}_N^{(2)}$ . For approximate (asymptotic) separability with  $\tilde{\mathbf{X}}_N^{(m)} \approx \mathbf{X}_N^{(m)}$ , we need the condition of approximate (asymptotic) orthogonality of subseries of the considered components. Asymptotic separability is considered if  $L$  and/or  $K$  tend to infinity.

For sufficiently long time series, SSA can approximately separate, for example, signal and noise, sine waves with different frequencies, trend and seasonality ([1, Chapter 6, Section 1.5] and [29, Section 2.3.3]).

Let us demonstrate the separability of two sinusoids with different frequencies  $\omega_1$  and  $\omega_2$ :  $x_n^{(i)} = A_i \cos(2\pi\omega_i n + \phi_i)$ . These sinusoids are asymptotically weakly separable; that is, their subseries are asymptotically orthogonal as their lengths tend to infinity. However, the rate of convergence depends on the difference between the frequencies. If the frequencies are close and the time series length is not long enough, the two series can be far from orthogonal and therefore not separable. Note that two sinusoids with equal amplitudes are asymptotically weakly separable, but not strongly asymptotically separable and therefore are mixed in the SSA decompositions.

## Separability measure

The so-called  $\mathbf{w}$ -correlation matrix contains very helpful information that can be used for detection of separability and identification of groups. This matrix consists of weighted cosines of angles between the reconstructed time series components.

Let  $w_n$  ( $n = 1, 2, \dots, N$ ) be the weights defined in (1.6):  $w_n$  is equal to the number of times the series element  $x_n$  appears in the trajectory matrix. Define the  $\mathbf{w}$ -scalar product of time series of length  $N$  as  $(\mathbf{Y}_N, \mathbf{Z}_N)_{\mathbf{w}} = \sum_{n=1}^N w_n y_n z_n = \langle \mathbf{Y}, \mathbf{Z} \rangle_F$ , where  $\mathbf{Y}$  and  $\mathbf{Z}$  are the  $L$ -trajectory matrices of

the series  $\mathbf{Y}_N$  and  $\mathbf{Z}_N$  respectively. Then we get a particular case of  $\mathbf{w}$ -correlations (1.8) between  $\mathbf{Y}_N$  and  $\mathbf{Z}_N$  for time series:

$$\rho_{\mathbf{w}}(\mathbf{Y}_N, \mathbf{Z}_N) = (\mathbf{Y}_N, \mathbf{Z}_N)_{\mathbf{w}} / (\|\mathbf{Y}_N\|_{\mathbf{w}} \|\mathbf{Z}_N\|_{\mathbf{w}}).$$

Well separated components in (1.14) have weak (or zero) correlation whereas poorly separated components typically have high correlation. Therefore, looking at the matrix of  $\mathbf{w}$ -correlations between elementary reconstructed series  $\tilde{\mathbf{X}}_N^{(i)}$  and  $\tilde{\mathbf{X}}_N^{(j)}$  one can find groups of correlated series components and use this information for the subsequent grouping. One of the main rules is: ‘do not include highly correlated components into different groups’. The  $\mathbf{w}$ -correlations can also be used for checking the grouped decomposition.

It is very instructive to depict the matrix of absolute values of  $\mathbf{w}$ -correlations between the series components graphically in the white-black scale, where small correlations are shown in white and correlations with their absolute values close to 1 are shown in black; see, for example, Figures 1.6.

## Choice of parameters

The conditions of (approximate) separability yield recommendations for the choice of the window length  $L$ : it should be large enough ( $L \sim N/2$ ) and if we want to extract a periodic component with known period, then the window lengths, which are divisible by the period, provide better separability [1, Section 1.6]. Choice of parameters is discussed in Section 3.4. If we choose a few leading eigentriples, then SSA with small  $L$  performs smoothing of the series as a filter of order  $2L - 1$ , see [29, Section 3.9]. Generally, the choice of the window length is important but the result is usually stable with respect to small changes in the values of  $L$ .

If the time series has a complex structure, then the so-called Sequential SSA [29, Section 2.5.5] is recommended. Sequential SSA consists of two stages; at the first stage, trend is extracted with a small window length and then periodic components are detected and extracted from the residual with  $L \sim N/2$ .

## Justification

If we use SSA as a model-free and exploratory technique, then the justification of the decomposition cannot be formal; it must be based on the separability theory and the interpretability of the results. Real-time or batch processing by SSA is possible if the class of series is not too broad and well-determined so that one can fix the rule for choosing proper parameters. For performing statistical testing, a model of the time series should be specified.

#### 1.8.4. Decomposition of e-m harmonics

Since we consider the oscillatory component as a sum of exponentially-modulated harmonics, let us describe the properties of the decomposition of such harmonics.

The elements of a real-valued exponential-modulated harmonic series  $\mathbf{S}$  have the form

$$s_n = a e^{\alpha n} \cos(2\pi\omega n + \phi), \quad (1.18)$$

where  $0 \leq \phi < 2\pi$ ,  $a \neq 0$ ,  $0 < \omega \leq 0.5$ ,  $\sin \phi \neq 0$  for  $\omega = 0.5$ .

As before,  $L$  is the window length,  $K = N - L + 1$ ,  $\Lambda^{(1)}$  and  $\Lambda^{(2)}$  are the subspaces of rows and columns of the trajectory matrix of the series  $\mathbf{S}_N$ .

$L$ -Rank  $d$  of the series is the rank of its trajectory matrix and, therefore, the number of left or right singular vectors corresponding to nonzero singular numbers of the trajectory matrix  $\mathbf{X}$ .

**Proposition 1.** [1, Section 5.1]

1.  $L$ -Rank  $d$  of a series  $\mathbf{S}$  whose elements are (1.18) is 1 if  $\omega = 0.5$ ; in other cases  $d = 2$ .
2. If  $\omega \neq 0.5$ , then the subspaces  $\Lambda^{(1)}$  and  $\Lambda^{(2)}$  have bases

$$\begin{aligned} &\{(1, e \cos(2\pi\omega), \dots, e^{\alpha(L-1)} \\ &\quad \cos(2\pi(L-1)\omega))^T, \\ &(0, e \sin(2\pi\omega), \dots, e^{\alpha(L-1)} \sin(2\pi(L-1)\omega))^T\} \quad \text{and} \\ &\{(1, e \cos(2\pi\omega), \dots, e^{\alpha(K-1)} \cos(2\pi(K-1)\omega))^T, \\ &(0, e \sin(2\pi\omega), \dots, e^{\alpha(K-1)} \sin(2\pi(K-1)\omega))^T\} \end{aligned}$$

respectively.

3. In the case  $\omega = 0.5$ , the one-dimensional  $\Lambda^{(1)}$  and  $\Lambda^{(2)}$  are linear spans of vectors  $(1, c, \dots, c^{L-1})^T$  and  $(1, c, \dots, c^{K-1})^T$  respectively, where  $c = -e^\alpha$ .

**Proposition 2.** [33, Proposition 2.3] Let  $\alpha = 0$ ,  $\omega \neq 0.5$  and  $L\omega$  and  $K\omega$  be integers. Then the eigenvalues of the SVD for the series  $\mathbf{S}$  with elements of the form (1.18) are the same and have the form  $\lambda_1 = \lambda_2 = a^2 L K / 4$ .

It follows from the statement 1 that for  $\omega \neq 0.5$  the series  $\mathbf{S}$  has exactly two left singular vectors  $U_1$  and  $U_2$  corresponding to nonzero eigenvalues of the trajectory matrix, and that the elements of vectors  $U_1$  and  $U_2$  can be represented in the following form:

$$u_k^{(1)} = a_1 e^{\alpha k} \cos(2\pi\omega k + \phi_1), \quad u_k^{(2)} = a_2 e^{\alpha k} \cos(2\pi\omega k + \phi_2), \quad (1.19)$$

where  $1 \leq k \leq L$ ,  $0 \leq \phi_1, \phi_2 < 2\pi$ ,  $a_1, a_2 \neq 0$ .

**Proposition 3.** Let  $\alpha = \alpha_N = C/N$ , where  $C$  is some constant, and  $L = [\beta N]$ , where  $0 < \beta < 1$ . Then for singular vectors  $U_1$  and  $U_2$  of the series  $S$  whose elements have the form (1.19), the relations  $\lim_{L \rightarrow \infty} |\phi_1 - \phi_2| = \pi/2 \pmod{\pi}$  and  $\lim_{L \rightarrow \infty} (a_1/a_2) = 1$  are satisfied.

### 1.8.5. Algorithm

In Section 1.8.1 we described the Basic SSA method. Here we formally present the algorithm of Basic SSA. Note that the RSSA package implements the algorithms efficiently (see Section 6.2 for a brief discussion). Since effective implementation is complicated and hides the sense of algorithm steps, we write down the algorithms in the original form.

Input data for the whole algorithm of Basic SSA are the window length and the way of grouping of the elementary components  $\mathbf{X}_i$  in (1.12). However, the rule for grouping is made after the decomposition (1.12) is made. Therefore, the grouping becomes the input data for Reconstruction stage. For this reason, we split the algorithm into two parts. Note that modifications of Basic SSA mostly differ by Decomposition step only; Reconstruction stage is the same for virtually all SSA versions.

#### ALGORITHM 1.1: Basic SSA: decomposition

*Input:* Time series  $\mathbf{X}$  of length  $N$ , window length  $L$ .

*Output:* Decomposition of the trajectory matrix on elementary matrices  $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d$ , where  $d = \text{rank } \mathbf{X}$  and  $\mathbf{X}_i = \sqrt{\lambda_i} U_i V_i^T$  ( $i = 1, \dots, d$ ).

- 1: Construct the trajectory matrix  $\mathbf{X} = \mathcal{T}_{\text{SSA}}(\mathbf{X})$ .
- 2: Compute the SVD  $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d$ ,  $\mathbf{X}_i = \sqrt{\lambda_i} U_i V_i^T$ .

Reconstruction algorithms are almost the same for different versions of SSA; their inputs have a decomposition of the trajectory matrix into a sum of rank-one matrices and the split of the rank-one components into groups. We therefore formulate a general algorithm of reconstruction and will make comments concerning specific features of modifications in the corresponding sections. The specific feature of Basic SSA is: the input decomposition is the SVD and hence the biorthogonal decomposition into the rank-one components is ordered according to component contribution  $\sigma_i^2 = \lambda_i$  so that  $\sigma_1 \geq \dots \geq \sigma_d$ .

#### ALGORITHM 1.2: Reconstruction

*Input:* Decomposition  $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d$ , where  $\mathbf{X}_i = \sigma_i U_i V_i^T$  and  $\|U_i\| = \|V_i\| = 1$ ; partition of indices:  $\{1, \dots, d\} = \bigsqcup_{j=1}^m I_j$ .

*Output:* Decomposition of the time series  $\mathbf{X}$  into identifiable components  $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_m$ .

- 1: Construct the grouped matrix decomposition  $\mathbf{X} = \mathbf{X}_{I_1} + \dots + \mathbf{X}_{I_m}$ , where  $\mathbf{X}_I = \sum_{i \in I} \mathbf{X}_i$ .

2: Compute  $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_m$ , where  $\mathbf{X}_i = \mathcal{T}_{\text{SSA}}^{-1} \circ \Pi_{\mathcal{H}}(\mathbf{X}_{I_i})$ .

---

### 1.8.6. Modification Toeplitz SSA for stationary time series

#### Method

Toeplitz SSA differs from Basic SSA only in Step 2 of the generic scheme presented in Figure 1.3; that is, in the decomposition of  $\mathbf{X}$  into rank-one matrices. Basic SSA uses the SVD at this step with  $U_i$  calculated as eigenvectors of  $\mathbf{S} = \mathbf{X}\mathbf{X}^T$ . If the length  $N$  of the series  $\mathbf{X}$  is not large and the series is assumed to be stationary, then the usual recommendation is to replace the matrix  $\mathbf{S}$  by some other matrix, which is constructed under the assumption of stationarity.

Note first that in Basic SSA we can consider the *lag-covariance matrix*  $\mathbf{C} = \mathbf{S}/K$  instead of  $\mathbf{S}$  for obtaining the SVD of the trajectory matrix  $\mathbf{X}$ . Indeed, the eigenvectors of the matrices  $\mathbf{S}$  and  $\mathbf{C}$  are the same and the eigenvalues differ only by the factor  $1/K$ .

Denote by  $c_{ij} = c_{ij}(N)$  the elements of the lag-covariance matrix  $\mathbf{C}$ . If the time series is stationary with zero mean,  $L$  is fixed and  $K \rightarrow \infty$ , then  $\lim c_{ij}(N) = R_{\mathbf{X}}(|i - j|)$  as  $N \rightarrow \infty$ , where  $R_{\mathbf{X}}(k)$  stands for the lag- $k$  term of the time series autocovariance function. We can therefore define a Toeplitz version of the lag-covariance matrix by putting equal values  $\tilde{c}_{ij}$  at each matrix auxiliary diagonal  $|i - j| = k$ . The most natural way for defining the values  $\tilde{c}_{ij}$  and the corresponding matrix  $\tilde{\mathbf{C}}$  is to compute

$$\tilde{c}_{ij} = \frac{1}{N - |i - j|} \sum_{m=1}^{N - |i - j|} x_m x_{m+|i - j|}, \quad 1 \leq i, j \leq L. \quad (1.20)$$

While using this formula it is usually assumed that the time series  $\mathbf{X}$  is centered so that the mean  $\bar{x} = \sum_{i=1}^N x_i / N$  is subtracted from all  $x_i \in \mathbf{X}$ .

Let  $L \leq K$  and denote by  $P_i$  ( $i = 1, \dots, L$ ) the eigenvectors of  $\tilde{\mathbf{C}}$ ; these vectors form an orthonormal basis of  $\mathbb{R}^L$ . Then the decomposition on elementary matrices can be written as  $\mathbf{X} = \sum_{i=1}^L P_i (\mathbf{X}^T P_i)^T$ . Ordering of addends is performed by the magnitudes of  $\sigma_i = \|\mathbf{X}^T P_i\|$ . Note that this ordering generally differs from the ordering of eigenvalues of the matrix  $\tilde{\mathbf{C}}$  corresponding to the eigenvectors  $P_i$ . Some of these eigenvalues could even be negative as the matrix  $\tilde{\mathbf{C}}$  is not necessarily positive definite.

If the original series is stationary with zero mean, then the use of *Toeplitz lag-covariance matrix*  $\tilde{\mathbf{C}}$  can be more appropriate than the use of the lag-covariance matrix  $\mathbf{C}$ . On the other hand, Toeplitz SSA is not suitable for nonstationary series; if the original series has an influential nonstationary component, then Basic SSA works better than Toeplitz SSA. For example, if we are dealing with a pure exponential series, then it is described by a single eigentriple for any window length, while Toeplitz SSA produces  $L$  eigentriples for the window length  $L$ ; moreover, the eigenvectors in Toeplitz SSA have some special properties [106]. The same effect takes place for the linear series, exponentially-modulated sinusoids, etc.

Let us remark that using  $\mathbf{C}$  to obtain the SVD of the trajectory matrix is sometimes called ‘BK’ following [60], while using  $\tilde{\mathbf{C}}$  to get the eigenvectors is initiated by the spectral analysis and is called ‘VG’ following [63].

## Algorithm

---

### ALGORITHM 1.3: Toeplitz SSA: decomposition

---

*Input:* Time series  $\mathbf{X}$  of length  $N$ , window length  $L$ .

*Output:* Decomposition of the trajectory matrix on elementary matrices  $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_L$ , where  $\mathbf{X}_i = \sigma_i P_i Q_i^T$ ,  $\|P_i\| = \|Q_i\| = 1$ .

- 1: Construct the trajectory matrix  $\mathbf{X} = \mathcal{T}_{\text{SSA}}(\mathbf{X})$ .
- 2: Obtain the decomposition

$$\mathbf{X} = \sum_{i=1}^L \sigma_i P_i Q_i^T = \mathbf{X}_1 + \dots + \mathbf{X}_L, \quad (1.21)$$

where  $\{P_i\}_{i=1}^L$  are eigenvectors of the matrix  $\tilde{\mathbf{C}}$  with entries computed by (1.20),  $S_i = \mathbf{X}^T P_i$ ,  $Q_i = S_i / \|S_i\|$  and  $\sigma_i = \|\mathbf{X}_i\|_F = \|S_i\|$ . Components are ordered by the magnitudes of  $\sigma_i$ :  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_L$ .

---

The reconstruction algorithm is exactly the same as Algorithm 1.2 with vectors  $(P_i, Q_i)$  substituted for  $(U_i, V_i)$ .

#### 1.8.7. Example of identification and decomposition

Let us demonstrate how to visually identify the SSA components and to obtain the SSA decomposition, by a simple example. In the 1D scatterplots (Fig. 1.4), one can find a slowly varying component (ET 1), whereas in the 2D scatterplots (Fig. 1.5) regular polygons say about pairs of sine-wave components (ET 2–3 for  $T = 12$ , ET 4–5 for  $T = 4$ , ET 6–7 for  $T = 6$ , ET 8–9 for  $T = 2.4$  and ET 10–11 for  $T = 3$ ). In these figures,  $U_i(k)$  denotes the  $k$ th coordinate of the  $i$ th eigenvector obtained in the SVD step of SSA. The eigenvector numbers are indicated at the captions of the graphs. Fig. 1.6 with the depicted  $w$ -correlations provides a guess for grouping, since strongly correlated components should be included in the same group (the black color shows correlations close to 1; the white color corresponds to zero correlations).

The resultant decomposition into the trend, the seasonality and noise is depicted in Fig. 1.7.

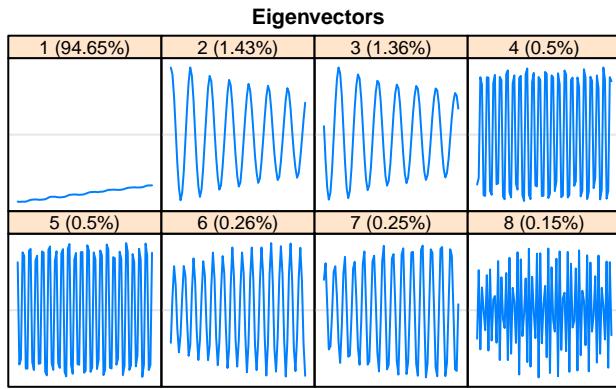


Figure 1.4: ‘Fortified wines’,  $L = 84$ : 1D graphs of eigenvectors  $(k, U_i(k))$ ,  $k = 1, \dots, L$ .

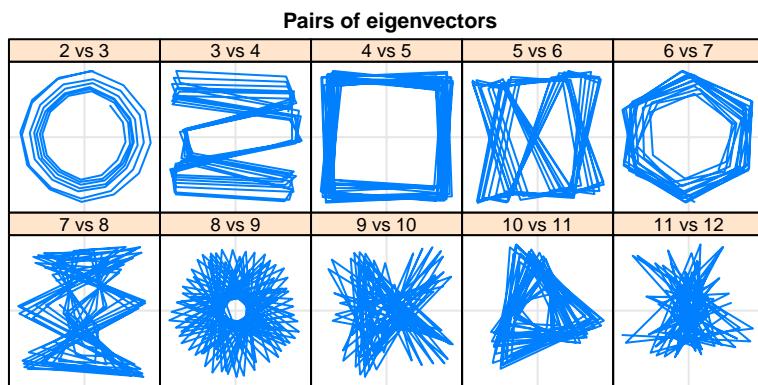


Figure 1.5: ‘Fortified wines’,  $L = 84$ : 2D scatterplots of eigenvectors  $(U_i(k), U_{i+1}(k))$ ,  $k = 1, \dots, L$ .

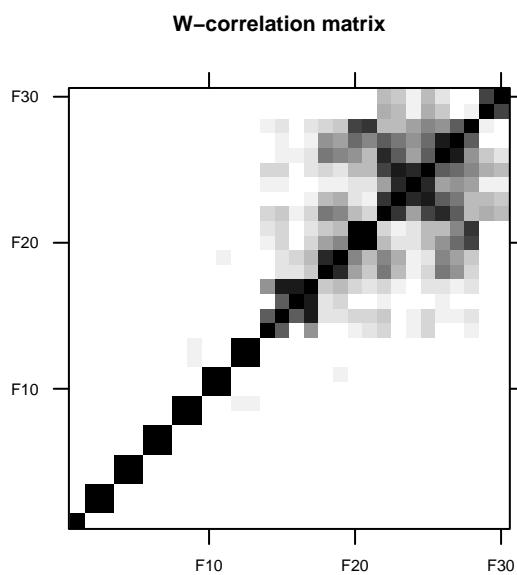


Figure 1.6: ‘Fortified wines’,  $L = 84$ :  $w$ -correlations between elementary RCs.

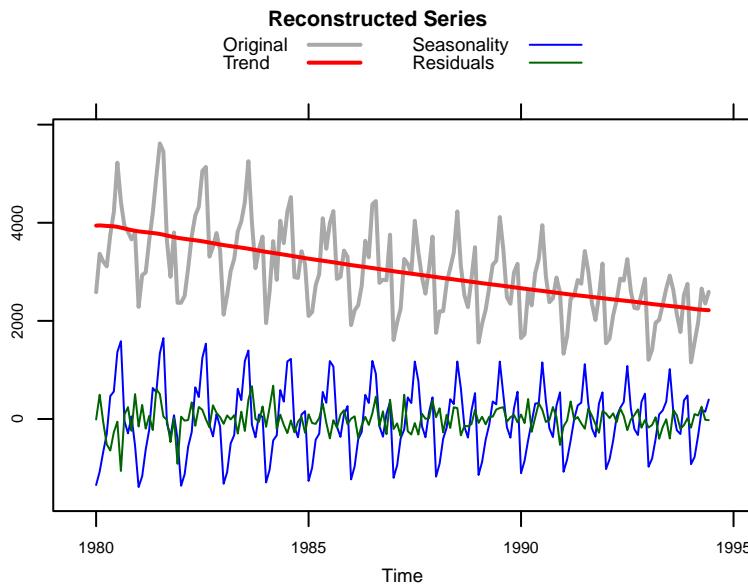


Figure 1.7: ‘Fortified wines’,  $L = 84$ : decomposition for groups ET1, ET2–11 and ET12–84.

#### 1.8.8. Example of problems with separability

In Section 1.8.7, we considered an example with simple trend and periodic components. That example corresponds to a good separability between trend, seasonal and noise components. However, for trends of complex form which are common in time series analysis, there is a big chance that trend components of the SSA decomposition can be mixed with seasonal components. Below we consider a short time series and demonstrate the problem of the lack of separability.

Let us analyze the first five years of the time series ‘MotorVehicle’ (Fig. 1.1). The first decomposition is performed by Basic SSA with  $L = 24$ . Figure 1.8 depicts 11 elementary reconstructed series (RCs), i.e., the time series, which are obtained with the grouping  $\{1, \dots, d\} = \bigcup_{i=1}^d \{i\}$ . These time series are helpful for the grouping procedure, since the reconstruction by eigentriples from a group  $I$  is just the sum of the elementary reconstructed components with numbers from  $I$ .

One can see that the trend is contained in slowly-varying RC 1 and partly in RC 2, 8 and 9; the latter elementary reconstructed time series contain a mixture of the trend and seasonality. This means that the reconstruction of the trend with ET1 is insufficient, while that with, say, ET1,2 will contain seasonality (see Figure 1.9).

Thus, if there is no separability of the trend from the residual, then it is impossible to extract an accurate trend. This example explains, why it is very important to try for separability by different means, such as the choice of the window length, sequential SSA, nested modifications of Basic SSA (e.g., Iterative Oblique SSA, SSA with derivatives; see Section 2.2.1).

Let us apply SSA with derivatives, which will be considered in detail in Section 2.3, to the group of the 11 leading ETs. It is clearly seen in Figure 1.10 that the trend components are RC 9–11. Figure 1.11 shows the extracted accurate trend.

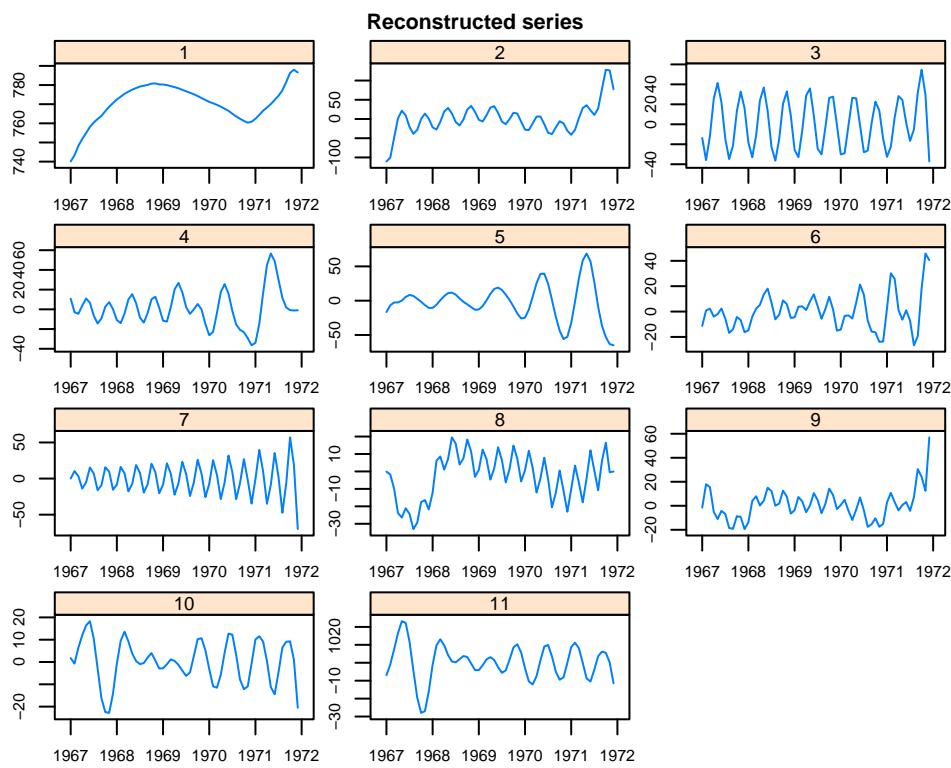


Figure 1.8: ‘MotorVehicle’ (5 years), SSA with  $L = 24$ : elementary RCs; poor separability.

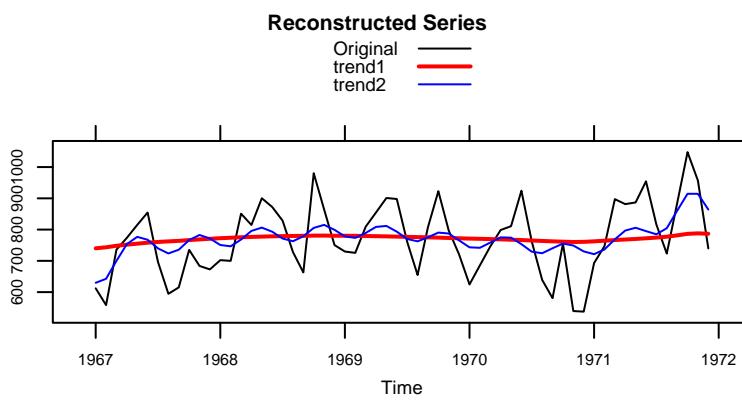


Figure 1.9: ‘MotorVehicle’ (5 years), SSA with  $L = 24$ : two trend reconstructions, ET1 and ET1–2; poor separability.

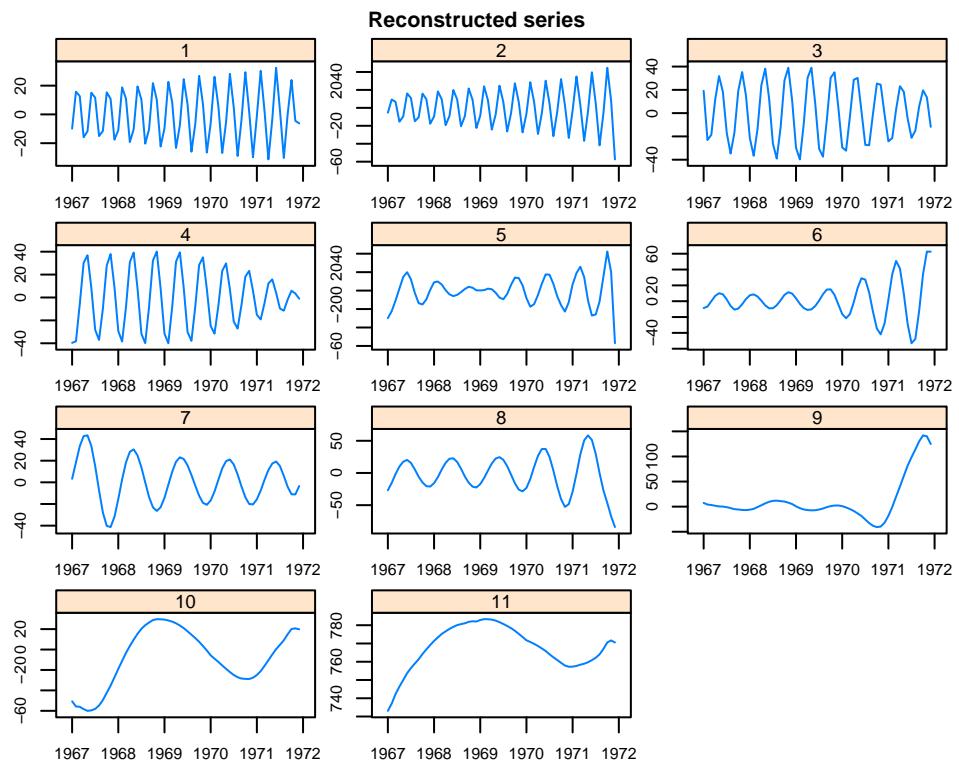


Figure 1.10: ‘MotorVehicle’ (5 years), DerivSSA with  $L = 24$ : elementary RCs; good separability.

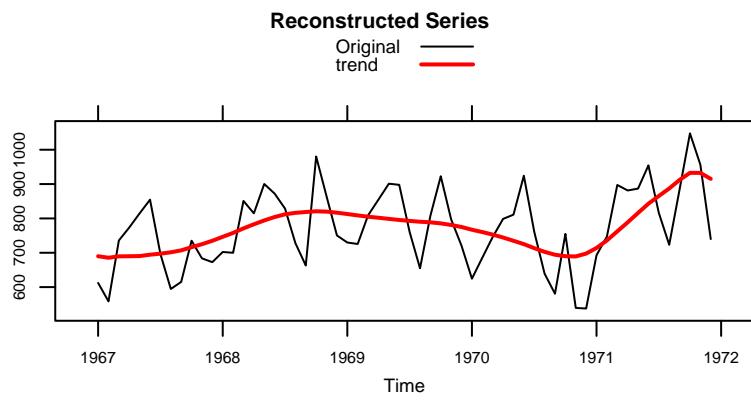


Figure 1.11: ‘MotorVehicle’ (5 years), DerivSSA with  $L = 24$ : trend reconstruction, ET9-11; good separability.

## 1.9. Forecasting and parameter estimation for one-dimensional time series

Introduction and Section 1.8.2 describe the class of signals that is suitable for the SSA method, as well as three forms of its assignment. These are signals whose trajectory matrix has an incomplete rank, or LRR-driven signals, or signals in explicit parametric form (1).

For such signals it is correct to put the problem of prediction and estimation of parameters. In this section, we briefly describe the solution of these problems in the framework of SSA.

### 1.9.1. Signal extraction via projections

Consider a particular case of  $\mathbf{X} = \mathbf{S} + \mathbf{N}$ ,  $\mathbf{X} = (x_1, \dots, x_N)$ , where  $\mathbf{S}$  is a signal of rank  $r$ ,  $\mathbf{N}$  is noise, and set the parameters: the window length  $L$  and the signal rank  $r$ .

Introduce two projections in Frobenius norm:  $\Pi_r : \mathbb{R}^{L \times K} \mapsto \mathcal{M}_r$ , where  $\mathcal{M}_r$  is the set of matrices of rank not larger than  $r$ , and  $\Pi_{\mathcal{H}} : \mathbb{R}^{L \times K} \mapsto \mathcal{H}$ , where  $\mathcal{H}$  is the set of Hankel matrices. Let  $\mathcal{T}$  be defined in (1.1).

**Scheme of SSA for signal extraction:**

$$\mathbf{X} \xrightarrow[L]{\mathcal{T}} \mathbf{X} = \begin{pmatrix} x_1 & x_2 & \dots & x_K \\ x_2 & x_3 & \dots & x_{K+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_L & x_{L+1} & \dots & x_N \end{pmatrix} \xrightarrow[r]{\text{SVD: } (\sqrt{\lambda_m}, U_m, V_m), \Pi_r} \mathbf{X}$$

$$\left\{ \begin{array}{l} \mathcal{L}_r = \text{span}(U_1, \dots, U_r) \\ \text{is the signal space;} \\ \Pi_r \text{ is the projector on } \mathcal{L}_r; \\ \widehat{\mathbf{S}} = \sum_{m=1}^r U_m (\mathbf{X}^T U_m)^T = \Pi_r \mathbf{X}. \end{array} \right. \xrightarrow{\Pi_{\mathcal{H}}} \widetilde{\mathbf{S}} = \begin{pmatrix} \tilde{s}_1 & \tilde{s}_2 & \dots & \tilde{s}_K \\ \tilde{s}_2 & \tilde{s}_3 & \dots & \tilde{s}_{K+1} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{s}_L & \tilde{s}_{L+1} & \dots & \tilde{s}_N \end{pmatrix} \xrightarrow{\mathcal{T}^{-1}} \widetilde{\mathbf{S}}.$$

Thus, a concise form of the SSA algorithm for the signal extraction is

$$\widetilde{\mathbf{S}} = \mathcal{T}^{-1} \Pi_{\mathcal{H}} \Pi_r \mathcal{T} \mathbf{X}.$$

The estimate thus obtained, in general, is not of rank  $r$ . To obtain, in the limit, a signal estimate of rank  $r$ , one can use the Cadzow method, which can be written through alternating projections as

$$\widetilde{\mathbf{S}}_k = \mathcal{T}^{-1} (\Pi_{\mathcal{H}} \Pi_r)^k \mathcal{T} \mathbf{X}. \quad (1.22)$$

### 1.9.2. Parameter estimation

In this section, we describe how to estimate the LRR coefficients and parameters of a series components governed by an LRR  $s_n = \sum a_i s_{n-i}$ . We will assume that the SSA method is able to

approximately extract the investigated series component; that is, the component of interest is approximately separated and the window length together with the SSA modification are chosen appropriately.

Although a series governed by an LRR can be expressed in the general parametric form (1.10), a common particular case is a series  $\mathbf{S} = (s_n)$  with  $s_n = \sum_{i=1}^r C_i \mu_i^n$ ,  $\mu_i \in \mathbb{C}$ , or, in the real-valued form,  $s_n = \sum_{i=1}^p A_i \exp(\alpha_i n) \cos(2\pi n \omega_i + \phi_i)$ , where  $A_i$ ,  $\alpha_i$ ,  $\omega_i$  and  $\phi_i$  ( $i = 1, \dots, p$ ) are unknown parameters whose values may be (and often are) of interest to the investigator. Thus, the problem of parameter estimation arises.

## Method

We describe the so-called subspace-based methods of parameter estimation, where only the estimated subspace of the series components is of concern but Reconstruction step of the SSA algorithm is of no importance.

Let a set of indices  $I$  correspond to the component of interest in the constructed decomposition of the trajectory matrix  $\mathbf{X} = \sum_{i=1}^d \sigma_i P_i Q_i^T$ . For simplicity of notation we assume  $I = \{1, 2, \dots, r\}$ . Then the estimated subspace is  $\tilde{\mathcal{S}} = \text{span}(P_1, \dots, P_r)$ . We always consider the generating set  $\{P_i\}$  of  $\tilde{\mathcal{S}}$  to be orthonormal as otherwise we can orthonormalize it. Since the original vectors  $P_i$  may be linearly dependent (for example, in the method of SSA with projection), the procedure of orthogonalization may reduce the number of vectors. We will consider  $r$  to be equal to the number of vectors after orthogonalization.

We consider two kinds of parametrization; first, in the form of a governing LRR and, second, in the form (1.10). Correspondingly, we describe how to estimate the coefficients of a governing LRR and the parameters of (1.10).

**Estimation of the governing LRR** The trajectory space  $\mathcal{S}$  of a signal  $\mathbf{S}$  governed by a particular LRR corresponds to many LRRs. More precisely, any vector from  $\mathcal{S}^\perp$  with the last coordinate  $-1$  produces such an LRR; in other words, any such vector from  $\mathcal{S}^\perp$  provides a set of coefficients for a linear combination of the first  $L - 1$  coordinates of a vector from  $\mathcal{S}$  to obtain the last coordinate; see [1, Section 5] and [29, Chapter 3] for detailed explanations.

Among all these LRRs (generating the same trajectory space  $\mathcal{S}$ ) there is the best LRR with minimal sum of squared coefficients (the so-called min-norm LRR). The min-norm LRR suppresses possible perturbations in the initial data as much as possible, which is important if we use this LRR for series generation or continuation, on the base of SSA approximation of the initial data.

For a chosen window length  $L$ , the signal subspace  $\mathcal{S} \subset \mathbb{R}^L$  and therefore the min-norm LRR has order  $L - 1$ . For each column vector  $P_i$  of  $\mathbf{P}_r$ , denote  $\pi_i$  the last coordinate of  $P_i$ ,  $\underline{P}_i \in \mathbb{R}^{L-1}$  the vector  $P_i$  with the last coordinate removed, and  $v^2 = \sum_{i=1}^r \pi_i^2$ . Then the elements of the vector

$$\mathcal{R} = (a_{L-1}, \dots, a_1) = \frac{1}{1-v^2} \sum_{i=1}^r \pi_i \underline{P}_i \quad (1.23)$$

provide the coefficients of the *min-norm* governing LRR:  $s_n = \sum_{i=1}^{L-1} a_i s_{n-i}$ .

For an estimated subspace  $\tilde{\mathcal{S}}$ , the estimated LRR is calculated in the same way, on the base of an orthonormal basis of  $\tilde{\mathcal{S}}$ .

**Estimation of frequencies** Let  $X_N = S_N + R_N$ , where  $s_n = \sum_{j=1}^r c_j \mu_j^n$  and the series  $S_N$  and  $R_N$  are approximately separable for a given window length  $L$ . Generally, the signal roots of the characteristic polynomial of a governing LRR allow estimation of the signal parameters  $\mu_j$ ,  $j = 1, \dots, r$  (see Section 1.8.2). However, the min-norm LRR is not minimal and therefore we should somehow distinguish between the signal roots and the extraneous roots. Usually, the signal roots of the min-norm LRR have maximal moduli (e.g. see [107]). Therefore, one can find roots of the min-norm LRR, arrange them in the order of decrease and take the first  $r$  roots.

However, the ordering is never guaranteed. Therefore, the methods that are able to separate the signal and extraneous roots could be very useful.

Let us describe one of these methods called ESPRIT [108]. This method is implemented in two versions, LS-ESPRIT and TLS-ESPRIT, where LS means least squares, TLS means total least squares (see e.g. the method description in [29, Section 3.8.2]). Other names are HSVD [109] and HTLS [110]. Here we describe the LS version (HSVD).

Denote  $\{P_1, \dots, P_r\}$  an orthonormal basis of the estimated subspace of the component under interest. Set  $\mathbf{P}_r = [P_1 : \dots : P_r]$  and let  $\underline{\mathbf{P}}_r$  be the matrix with the last row removed and  $\overline{\mathbf{P}}_r$  be the matrix with the first row removed. Then  $\mu_i$  can be estimated by the eigenvalues of the matrix  $\underline{\mathbf{P}}_r^\dagger \overline{\mathbf{P}}_r$ , where  $\dagger$  denotes pseudo-inversion. Correspondingly, the estimated frequencies are the arguments of  $\mu_i$ .

Note that the matrix  $\mathbf{P}_r$  conventionally consists of the chosen eigenvectors  $U_i$  in the Basic SSA algorithm. However, any basis of the subspace, which estimates the signal subspace, is suitable.

Let us mention a simple and fast method of frequency estimation which is used for identification of the eigentriples at Grouping step. Two vectors  $U^{(1)}$  and  $U^{(2)}$  forming an orthogonal basis of the trajectory space of an exponentially-modulated sine wave have similar forms (the same periods) and their phases differ by approximately  $\pi/2$ . Let  $A$  and  $B$  be defined by  $a_n = \rho^n \sin(2\pi\omega n + \phi)$  and  $b_n = \rho^n \cos(2\pi\omega n + \phi)$ . Denote the angle between vectors by  $\angle$ . Then  $\omega = \angle\left(\begin{pmatrix} a_1 \\ b_1 \end{pmatrix}, \begin{pmatrix} a_2 \\ b_2 \end{pmatrix}\right) / (2\pi)$ . Therefore, we can estimate the frequency using the basis vectors  $U^{(1)}$  and  $U^{(2)}$ . Since these vectors do not have exactly the same form as  $A$  and  $B$ , the sequence of angles  $\angle\left(\begin{pmatrix} u_i^{(1)} \\ u_i^{(2)} \end{pmatrix}, \begin{pmatrix} u_{i+1}^{(1)} \\ u_{i+1}^{(2)} \end{pmatrix}\right) / (2\pi)$ ,  $i = 1, \dots, L - 1$ , can be considered and then the mean or median can be taken as an estimate of the frequency; see [1, Section 1.6] for details. In RSSA, the median is considered and the median of absolute deviations from the median is used as a measure of accuracy.

## Algorithms

Although the LRR approximating the time series is usually used for forecasting, it can also be helpful for construction of the signal model. Hence we introduce an algorithm for calculation of

the min-norm LRR coefficients.

ALGORITHM 1.4: Estimation of the signal LRR

*Input:* Matrix  $\mathbf{P}_r \in \mathbb{R}^{L \times r}$  consisting of orthonormal column vectors, which form a basis of the estimated signal subspace.

*Output:* Coefficients  $\mathcal{R} = (a_{L-1}, \dots, a_1)$  of the corresponding LRR.

- 1: For each column vector  $P_i$  of  $\mathbf{P}_r$ , calculate  $\pi_i$  and  $\underline{P}_i$ ,  $v^2 = \sum_{i=1}^r \pi_i^2$ . If  $v^2$  is equal to 1, then STOP with the error message “Verticality coefficient equals 1”.
- 2: Compute  $\mathcal{R} = \frac{1}{1-v^2} \sum_{i=1}^r \pi_i \underline{P}_i$ .

The next algorithm shows how the parameters  $\mu_i$  in  $s_n = \sum_{i=1}^r C_i \mu_i^n$  can be estimated from the roots of the characteristic polynomial of an LRR governing this time series (see Section 1.8.2 for a description of the relation between LRRs and their characteristic polynomials). The given LRR is an estimate of an LRR governing a series of rank  $r$ ; therefore, only  $r$  roots correspond to the signal, while the other roots are extraneous. Since frequently (but not always!) the signal roots for the min-norm LRR have larger moduli than the extraneous roots, we can select signal roots with large absolute values among the whole set of roots.

ALGORITHM 1.5: Estimation of the signal roots through characteristic polynomial of LRR

*Input:* Coefficients  $A = (a_1, \dots, a_m)$  of the LRR  $s_n = \sum_{i=1}^m a_i s_{n-i}$ , rank  $r$ .

*Output:* Signal roots  $\mu_i$ ,  $i = 1, \dots, r$ .

- 1: Construct the characteristic polynomial  $P(\mu) = \mu^d - \sum_{i=1}^m a_i \mu^{n-i}$ .
- 2: Find the roots  $\mu_1, \dots, \mu_m$  of  $P(\mu)$ .
- 3: Order the roots so that  $|\mu_1| \geq \dots \geq |\mu_m|$ .
- 4: The leading roots  $\mu_i$ ,  $i = 1, \dots, r$ , are the candidates for the signal roots.

Algorithm 1.6 is one of the most known high-resolution subspace-based algorithms of estimation of frequencies and damping factors.

ALGORITHM 1.6: ESPRIT

*Input:* Matrix  $\mathbf{P}_r \in \mathbb{R}^{L \times r}$  consisting of orthonormal column vectors, which form a basis of the estimated signal space.

*Output:*  $r$  roots in the form  $(\rho_i, \omega_i)$ .

- 1: Using either LS or TLS method, find a matrix  $\mathbf{M} \in \mathbb{R}^{r \times r}$  satisfying  $\overline{\mathbf{P}_r} \approx \underline{\mathbf{P}_r} \mathbf{M}$ . For the LS-method,  $\mathbf{M} = \underline{\mathbf{P}_r}^\dagger \overline{\mathbf{P}_r}$ .

- 
- 2: Find eigenvalues  $\mu_i$ ,  $i = 1, \dots, r$ , of  $\mathbf{M}$ .  
 3: Set  $\rho_i = \text{Mod}(\mu_i)$ ,  $\omega_i = \text{Arg}(\mu_i)$ .
- 

The next algorithm is a complementary to Decomposition step used for helping to gather sine-waves with similar frequencies.

---

**ALGORITHM 1.7: Fast ('pairs') estimation of frequencies**

---

*Input:* Two orthonormal vectors  $U^{(1)}$  and  $U^{(2)}$  forming an estimated trajectory space of a sine wave.

*Output:* Frequency  $\omega$ , period  $T$ .

- 1: Compute  $\phi_i = \angle \left( \begin{pmatrix} u_i^{(1)} \\ u_i^{(2)} \end{pmatrix}, \begin{pmatrix} u_{i+1}^{(1)} \\ u_{i+1}^{(2)} \end{pmatrix} \right)$ ,  $i = 1, \dots, L - 1$ .
  - 2: Calculate  $\bar{\phi}$  as the mean or median of  $\{\phi_i\}$ .
  - 3:  $\omega = \bar{\phi}/(2\pi)$ ,  $T = 1/\omega$ .
- 

### 1.9.3. Forecasting

The problem of forecasting is the problem of continuation of the signal  $S$  extracted from the observed series  $X = S + R$ . To do that it is sufficient to estimate the trajectory space of  $S$  and then to construct the forecasted series based on the estimated subspace.

An obvious way to perform forecasting would be to estimate series parameters and use them for forecasting. However, the class of series suitable for forecasting is considerably wider than the class of series for parameter estimation and hence the forecasting methods considered below do not estimate series parameters but only use some features of the estimated subspace.

Generally, the forecasted series should have a structure to forecast. In the framework of SSA, we say that  $S$  has a structure if  $S$  is governed by an LRR. However, it is very important to note that SSA forecasts are meaningful for a much wider class of series when an LRR gives an adequate description of the structure of the series only locally rather than globally, which is a requirement for parameter estimation. For example, as a rule, a trend does not satisfy an LRR on the whole time range but it can be locally approximated by a smooth series governed by an LRR. In particular, relatively reliable forecasts can be made for the series which can be approximated by a series of the form  $s_n = \sum_{i=1}^r C_i(n) \mu_i^n$ ,  $\mu_i \in \mathbb{C}$ , where  $C_i(n)$  are slowly varying functions of  $n$ .

In any version of the forecasting algorithm, we should assume that the series  $S$  is approximately separated from  $R$  by a chosen modification of the SSA method.

## Method

The methods of SSA forecasting are closely related to construction of LRRs described in Section 1.9.2. Below we describe two forecasting algorithms. One of them directly uses the constructed LRR for forecasting, whereas the other algorithm does it implicitly. Both algorithms provide the same forecasts of series governed by LRRs, if parameters of SSA are chosen properly.

**Approach** We will use the same notation as in Section 1.9.2.

Let  $\{P_i\}$  be a basis of a subspace  $\tilde{\mathcal{S}}$  of  $\mathbb{R}^L$ . Then we can state the problem of forecasting in this subspace.

Different modifications of SSA described in Chapter 2 (except for SSA with projection) provide an estimate of a basis of the signal subspace.

If an SSA modification is applied and a set of eigentriples  $\{(\sigma_i, P_i, Q_i), i \in I\}$  is chosen for reconstruction, then  $\tilde{\mathcal{S}} = \text{span}\{P_i, i \in I\}$ . The set of vectors  $\{P_i, i \in I\}$  is not necessarily an orthonormal basis. The first mandatory step is therefore the ortho-normalization of the set of vectors  $\{P_i, i \in I\}$ . After making this step we can construct forecasting algorithms considering an orthonormal basis as an input.

The subspace  $\tilde{\mathcal{S}}$  produces coefficients of a linear combination for reconstruction of the last coordinates of vectors from  $\tilde{\mathcal{S}}$  through their first  $L - 1$  coordinates. The linear combination used for forecasting has minimal Euclidean norm of coefficients among all linear combinations that correspond to the subspace  $\tilde{\mathcal{S}}$ . If  $\tilde{\mathcal{S}}$  is exactly the trajectory subspace of a series governed by an LRR, then the linear combination with minimal norm corresponds to the min-norm LRR. Moreover, the continuation of the series in this subspace is unique.

The advantage of the min-norm forecasting LRR is shown in the following proposition.

**Proposition 4.** *Let  $X_N = S_N + P_N$ , where  $P_N$  is a stationary white noise with zero mean and variance  $\sigma^2$ ,  $X$  and  $S$  be  $L$ -lagged vectors of  $X_N$  and  $S_N$  correspondingly and  $C \in \mathbb{R}^{L-1}$ . Then for  $x = C^T \bar{S}$  and  $\tilde{x} = C^T \bar{X}$ , we have  $E\tilde{x} = x$  and  $D\tilde{x} = \|C\|^2 \sigma^2$ .*

The proof directly follows from the equality  $D \sum_{i=1}^{L-1} c_i (y_i + \varepsilon_i) = D \sum_{i=1}^{L-1} c_i \varepsilon_i = \|C\|^2 \sigma^2$ , where  $C = (c_1, \dots, c_{L-1})^T$  and  $\varepsilon_i$ ,  $i = 1, \dots, L - 1$  are i.i.d. random variables with zero mean and variance  $\sigma^2$ .

If  $X = X_K$  is the last lagged vector of  $S_N$ , then  $\tilde{x} = C^T \bar{X}_K$  can be considered as a forecasting formula applied to a noisy signal and  $\|C\|^2$  regulates the variance of this forecast.

The following property of the min-norm LRR, which was derived in [111], is extremely important for forecasting: all extraneous roots of the min-norm LRR lie inside the unit circle of the complex plane. This property gives us hope that in the case of real-life series (when both the min-norm LRR and the related initial data are perturbed) the terms related to the extraneous roots in (1.10) only slightly influence the forecast. Moreover, bearing in mind the results concerning the distribution of the extraneous roots (see [112, 107]), we can expect that the extraneous summands cancel each other out.

However, if the series subspace is estimated approximately, several versions of forecasting can be suggested. If the estimation of the subspace was accurate enough, then different forecasting versions will be close. Otherwise, they can differ considerably.

Now we formally describe the forecasting algorithms. For detailed explanation, see [1, Chapter 2].

**Recurrent forecasting** The recurrent SSA forecasting is performed by means of the min-norm LRR defined in (1.23).

The *recurrent forecasting method* can be formulated as follows.

1. The time series  $\mathbf{Y}_{N+M} = (y_1, \dots, y_{N+M})$  is defined by

$$y_i = \begin{cases} \tilde{x}_i & \text{for } i = 1, \dots, N, \\ \sum_{j=1}^{L-1} a_j y_{i-j} & \text{for } i = N+1, \dots, N+M. \end{cases} \quad (1.24)$$

2. The numbers  $y_{N+1}, \dots, y_{N+M}$  form the  $M$  terms of the recurrent forecast.

Thus, the recurrent forecasting is performed by the direct use of the forecasting LRR with coefficients taken from  $\mathcal{R} = (a_{L-1}, \dots, a_1)$ .

**Remark 1.** Let us define the linear operator  $\mathcal{P}_{\text{Rec}} : \mathbb{R}^L \mapsto \mathbb{R}^L$  by the formula

$$\mathcal{P}_{\text{Rec}} Z = \begin{pmatrix} \bar{Z} \\ \mathcal{R}^T \bar{Z} \end{pmatrix}, \quad (1.25)$$

where  $\bar{Z}$  consists of the last  $L - 1$  coordinates of  $Z$ . Set

$$Y_i = \begin{cases} \tilde{X}_i & \text{for } i = 1, \dots, K, \\ \mathcal{P}_{\text{Rec}} Y_{i-1} & \text{for } i = K+1, \dots, K+M. \end{cases} \quad (1.26)$$

It is easily seen that the matrix  $\mathbf{Y} = [Y_1 : \dots : Y_{K+M}]$  is the trajectory matrix of the series  $\mathbf{Y}_{N+M}$ . Therefore, (1.26) can be regarded as a vector version of (1.24).

**Remark 2.** In recurrent forecasting, the original series can be taken instead of the reconstructed series as the initial data for the forecasting LRR. This may be sensible only if the leading components are chosen for forecasting. This option can reduce the bias caused by the reconstruction inaccuracy but the volatility of forecasts may increase.

If the LRR is not minimal, then only  $r$  of the roots correspond to the signal. Other roots are extraneous and can influence the forecast. Extraneous roots that have moduli larger than 1 can lead to instability.

**Vector forecasting** Let  $\mathcal{L}_r = \text{span}(P_i, i \in I)$  and  $\widehat{X}_i$  be the projection of the lagged vector  $X_i$  on  $\mathcal{L}_r$ . Consider the matrix

$$\Pi = \underline{\mathbf{P}}\underline{\mathbf{P}}^T + (1 - v^2)\mathcal{R}\mathcal{R}^T, \quad (1.27)$$

where  $\underline{\mathbf{P}} = [\underline{P}_1 : \dots : \underline{P}_r]$  and  $\mathcal{R}$  is defined in (1.23). The matrix  $\Pi$  defines the linear operator that performs the orthogonal projection  $\mathbb{R}^{L-1} \mapsto \underline{\mathcal{L}}_r$ , where  $\underline{\mathcal{L}}_r = \text{span}(\underline{P}_i, i \in I)$ . Finally, we define the linear operator  $\mathcal{P}_{\text{Vec}} : \mathbb{R}^L \mapsto \mathcal{L}_r$  by the formula

$$\mathcal{P}_{\text{Vec}} Z = \begin{pmatrix} \Pi \bar{Z} \\ \mathcal{R}^T \bar{Z} \end{pmatrix}. \quad (1.28)$$

The *vector forecasting method* can be formulated as follows.

1. In the notation above, define the vectors

$$Y_i = \begin{cases} \widehat{X}_i & \text{for } i = 1, \dots, K, \\ \mathcal{P}_{\text{Vec}} Y_{i-1} & \text{for } i = K+1, \dots, K+M+L-1. \end{cases} \quad (1.29)$$

2. By constructing the matrix  $\mathbf{Y} = [Y_1 : \dots : Y_{K+M+L-1}]$  and making its diagonal averaging we obtain the series  $y_1, \dots, y_{N+M+L-1}$ .
3. The numbers  $y_{N+1}, \dots, y_{N+M}$  form the  $M$  terms of the vector forecast.

In recurrent forecasting, we perform diagonal averaging to obtain the reconstructed series and then apply the LRR. In the vector forecasting algorithm, these steps are applied in the reverse order. The vector forecast is typically slightly more stable. The current fast implementation of the vector forecasting makes the vector forecasting comparable with recurrent forecasting in terms of the computational cost, see Section 4.3.2.

## Algorithms

Let a modification of SSA be applied to the time series  $\mathbf{X}$  and let an eigentriple group  $\{(\sigma_i, P_i, Q_i), i \in I\}$  be chosen for reconstruction. The suggested forecasting algorithms are formulated for forecasting in the subspace  $\mathcal{L}_r = \text{span}\{P_i, i \in I\} \subset \mathbb{R}^L$ . For simplicity, we assume that  $I = \{1, \dots, r\}$  and the vectors  $P_i, i \in I$ , are orthonormal. Note that the forecasting values do not depend on the choice of basis in  $\mathcal{L}_r$ .

### ALGORITHM 1.8: Recurrent SSA forecasting

*Input:* Time series  $\mathbf{X}$  of length  $N$ , window length  $L$ , orthonormal system of vectors  $\{P_i\}_{i=1}^r$ , forecast horizon  $M$ .

*Output:* Forecast values  $(\tilde{x}_{N+1}, \dots, \tilde{x}_{N+M})$ .

- 1: Construct the vector  $\mathcal{R} = (a_{L-1}, \dots, a_1)^T$  of coefficients of the min-norm LRR by Algorithm 1.4 applied to  $\{P_i; i \in I\}$ .
- 2: Construct the reconstructed matrix  $\widehat{\mathbf{X}} = \mathbf{P}\mathbf{P}^T\mathbf{X}$ , where  $\mathbf{P} = [P_1 : \dots : P_r]$ , and the reconstructed series  $\widetilde{\mathbf{X}} = (\widetilde{x}_1, \dots, \widetilde{x}_N)$  by  $\widetilde{\mathbf{X}} = \mathcal{T}_{\text{SSA}}^{-1} \circ \Pi_{\mathcal{H}}(\widehat{\mathbf{X}})$ .
- 3: Calculate the forecast values by applying the min-norm LRR:

$$\widetilde{x}_n = \sum_{i=1}^{L-1} a_i \widetilde{x}_{n-i}, \quad n = N+1, \dots, N+M$$


---

Algorithm 1.8 is written in the form, when the reconstructed series is taken as a base for forecasting. If the original series is used as the base of forecasting,  $x_n$  are taken instead of  $\widetilde{x}_n$  for  $n = N-L+2, \dots, N$  at Step 3.

Algorithm 1.8 constructs a forward recurrent forecasting. Backward recurrent forecasting is obtained by applying the forward forecasting to the reversed series.

The next algorithm implements the algorithm of vector forecasting, where the application of the min-norm LRR and the hankelization operation are taken in the reverse order.

---

#### ALGORITHM 1.9: Vector SSA forecasting

---

*Input:* Time series  $\mathbf{X}$  of length  $N$ , window length  $L$ , orthonormal system of vectors  $\{P_i\}_{i=1}^r$ , forecast horizon  $M$ .

*Output:* Forecast values  $(\widetilde{x}_{N+1}, \dots, \widetilde{x}_{N+M})$ .

- 1: Obtain the vector  $\mathcal{R} = (a_{L-1}, \dots, a_1)^T$  of coefficients of the min-norm LRR by Algorithm 1.4 applied to  $\{P_i; i \in I\}$ .
- 2: Calculate the matrix  $\Pi$  of projection given in (1.27).
- 3: Construct the reconstructed matrix  $\widehat{\mathbf{X}} = \mathbf{P}\mathbf{P}^T\mathbf{X}$ , where  $\mathbf{P} = [P_1 : \dots : P_r]$ .
- 4: Extend the reconstructed matrix  $\widehat{\mathbf{X}} = [\widehat{X}_1 : \dots : \widehat{X}_K]$  by column vectors:

$$\widehat{X}_n = \mathcal{P}_{\text{Vec}} \widehat{X}_{n-1} \quad \text{for } n = K+1, \dots, K+M+L-1,$$

where  $\mathcal{P}_{\text{Vec}}$  is given in (1.28) and uses  $\Pi$  and  $\mathcal{R}$ . Denote the extended matrix  $\widehat{\mathbf{X}}_{\text{ext}} \in \mathbb{R}^{L \times (K+M+L-1)}$ .

- 5: Obtain the extended reconstructed series  $\widetilde{\mathbf{X}}_{\text{ext}} = (\widetilde{x}_1, \dots, \widetilde{x}_{N+M+L-1})$  as  $\widetilde{\mathbf{X}}_{\text{ext}} = \mathcal{T}_{\text{SSA}}^{-1} \circ \Pi_{\mathcal{H}}(\widehat{\mathbf{X}}_{\text{ext}})$ .
  - 6: Return the forecast values  $(\widetilde{x}_{N+1}, \dots, \widetilde{x}_{N+M})$ .
- 

Additional  $L-1$  vectors  $\widehat{X}_n$  at Step 4 are calculated to make the forecast values independent on the forecast horizon.

In Algorithm 1.9, the reconstructed series is taken as the base for forecasting. For vector forecasting, it makes little sense to use the original series as the base for forecasting.

Note that in this straightforward form, Algorithm 1.9 has a much larger computational cost than Algorithm 1.8. However, a fast implementation described in Section 4.3.2 and realized in RSSA makes the vector forecasting as fast as the recurrent one.

## Chapter 2

# Decomposition step for one-dimensional time series

In the present chapter and Chapter 3, we thoroughly examine the use of SSA for one-dimensional data. This chapter is fully devoted to the SSA analysis of such data for constructing time series decompositions. Consideration of SSA gap filling, signal estimation and signal detection is delayed until Chapter 3. The main difference between the materials of these two chapters is the use of the models. In the present chapter, the use of models is minimal; on the contrary, the methodologies of Chapter 3 are model-based.

In the terminology of Chapter 1, SSA for one-dimensional data should be referred to as 1D-SSA. However, but for the sake of brevity, in this chapter we will refer to it simply as SSA. The SSA input for all algorithms of this chapter is a collection  $X_N = (x_1, \dots, x_N)$  of  $N$  real numbers, which can be thought of as a time series. Let  $L$  ( $1 < L < N$ ) be a window length and  $K = N - L + 1$ .

The common steps of all modifications of the SSA algorithms discussed in this chapter are the embedding procedure in step 1, grouping in step 3, and diagonal averaging, which makes recovery in step 4 (see Section 1.8.1 and Fig. 1.3).

Section 2.1 contains a variation of the decomposition step, which can be used if an additional information about the time series structure is known. The example is the extraction of polynomial trends.

Sections 2.2, 2.3 and 2.4 propose nested decompositions for improving the separability.

Section 2.5 describes the methods of automation of the grouping step, which works if the separated time series components are (approximate) separable. Therefore, the methods for separability improving should be applied before the grouping.

## 2.1. SSA with projection

We consider the combination of singular spectrum analysis, which does not need a series model given in advance, and of a subspace-based parametric approach, which is incorporated by means of projections to subspaces given in advance. In presenting the results of this section, we follow [16].

The aim is to provide an additional theoretical support to SSA with double centering (ProjSSA(1,1)), which was known before [1], and also to enlarge the range of applications of semi-nonparametric modifications of Basic SSA. The applications are related to extraction of polynomial trends.

### 2.1.1. SSA with centering

There are modifications of SSA called SSA with centering [1, Sections 1.7 and 6.3]. They serve for better separation of constant (SSA with single centering) and linear (SSA with double

centering) trends. Initially, SSA with single centering was created by analogy with Principal Component Analysis, where the columns of a data matrix (which can be interpreted as the rows of the trajectory matrix of a series) are centered before the application of the SVD. Let us describe the approach with centering.

Let us consider a time series  $\mathbf{X}$  of length  $N$ , a window length  $L$ ,  $K = N - L + 1$ , the trajectory matrix  $\mathbf{X}$  of the series  $\mathbf{X}$ .

A general form of the considered modification of Decomposition stage can be expressed as

1. Calculation of a special matrix  $\mathbf{C}^{(\text{center})} = \mathbf{C}(\mathbf{X})$  based on a priori information.
2. Computation of  $\mathbf{X}^* = \mathbf{X} - \mathbf{C}^{(\text{center})}$ .
3. Construction of the SVD:  $\mathbf{X}^* = \sum_{i=1}^{d^*} \sqrt{\lambda_i^*} U_i^* (V_i^*)^T$ .

Thus, we have the decomposition  $\mathbf{X} = \mathbf{C}^{(\text{center})} + \sum_{i=1}^{d^*} \sqrt{\lambda_i^*} U_i^* (V_i^*)^T$ .

Denote  $E_M = (1, \dots, 1)^T \in \mathbb{R}^M$  the vector of units. Centering is considered in the following forms:

- *Single row centering* when  $\mathbf{C}_{\text{row}}^{(\text{center})}(\mathbf{X}) = (\mathbf{X}E_K/K)E_K^T$  corresponds to averaging by rows, that is, each element of a row of  $\mathbf{C}_{\text{row}}^{(\text{center})}$  consists of the average of the corresponding row of the trajectory matrix.
- *Single column centering* when  $\mathbf{C}_{\text{col}}^{(\text{center})}(\mathbf{X}) = E_L(E_L^T \mathbf{X}/L)$  corresponds to averaging by columns.
- *Double centering* when  $\mathbf{C}_{\text{both}}^{(\text{center})} = \mathbf{C}_{\text{row}}^{(\text{center})} + \mathbf{C}_{\text{col}}^{(\text{center})}(\mathbf{X} - \mathbf{C}_{\text{row}}^{(\text{center})}(\mathbf{X}))$  corresponds to averaging by both rows and columns.

Note that the centering can be considered as a projection of rows and/or columns of  $\mathbf{X}$  on  $\text{span}(E_K)$  or  $\text{span}(E_L)$  respectively, since  $E_K E_K^T$  and  $E_L E_L^T$  are exactly the matrices of the projection operators. Therefore, centering in SSA can be considered as a preliminary projection of the trajectory matrix on a given subspace; the residual matrix  $\mathbf{X}^*$  is subsequently expanded by SVD or any other decomposition.

### 2.1.2. SSA with projection

Let us generalize the approach described in Section 2.1.1 involving projections to arbitrary spaces.

**Notation.** Let  $\Pi_{\text{col}} : \mathbb{R}^L \rightarrow \mathcal{L}_{\text{col}}$  and  $\Pi_{\text{row}} : \mathbb{R}^K \rightarrow \mathcal{L}_{\text{row}}$  be orthogonal projectors, where  $\mathcal{L}_{\text{col}} \in \mathbb{R}^L$  is called the column projection space and  $\mathcal{L}_{\text{row}} \in \mathbb{R}^K$  is called the row projection space. For any  $\mathbf{Y} \in \mathcal{M}_{L,t}$ , denote  $\Pi_{\text{col}}(\mathbf{Y})$  the matrix consisting of the columns, which result from projections of the columns of  $\mathbf{Y}$ , while for any  $\mathbf{Y} \in \mathcal{M}_{t,K}$  denote  $\Pi_{\text{row}}(\mathbf{Y})$  the matrix consisting of the rows, which result from projections of the rows of  $\mathbf{Y}$ .

Denote a basis of the column projection space ( $P_i, i = 1, \dots, p$ ) and a basis of the row projection space ( $Q_i, i = 1, \dots, q$ ),  $\mathbf{P} = [P_1 : \dots : P_p]$ ,  $\mathbf{Q} = [Q_1 : \dots : Q_q]$ . Without loss of generality we assume that  $\{P_i, i = 1, \dots, p\}$  and  $\{Q_i, i = 1, \dots, q\}$  are orthonormal bases of  $\mathcal{L}_{\text{col}}$  and  $\mathcal{L}_{\text{row}}$  (otherwise, we can perform ortho-normalization).

In SSA with projection, the scheme of SSA with centering, which is described in Section 2.1.1, is extended to arbitrary projections, that is,  $\mathbf{C} = \Pi_{\text{col}}(\mathbf{X})$  for column projection,  $\mathbf{C} = \Pi_{\text{row}}(\mathbf{X})$  for row projection and  $\mathbf{C} = \Pi_{\text{both}}(\mathbf{X})$  for double projection, where

$$\begin{aligned}\Pi_{\text{both}}(\mathbf{X}) &= \Pi_{\text{row}}(\mathbf{X}) + \Pi_{\text{col}}(\mathbf{X} - \Pi_{\text{row}}(\mathbf{X})) \\ &= \Pi_{\text{col}}(\mathbf{X}) + \Pi_{\text{row}}(\mathbf{X} - \Pi_{\text{col}}(\mathbf{X})) \\ &= \Pi_{\text{row}}(\mathbf{X}) + \Pi_{\text{row}}(\mathbf{X}) - (\Pi_{\text{col}} \circ \Pi_{\text{row}})(\mathbf{X}).\end{aligned}\quad (2.1)$$

If either the column or row basis is absent (that is, the corresponding projection should not be performed), then we formally set the corresponding projector to be the zero operator implying  $\mathbf{C} = \Pi_{\text{both}}(\mathbf{X})$  for any mode.

**Decomposition into elementary matrices.** A general form of the decomposition provided by SSA with projection is

$$\mathbf{X} = \mathbf{C} + \sum_{i=1}^{d^*} \sqrt{\lambda_i^*} U_i^* (V_i^*)^T, \quad (2.2)$$

where  $\mathbf{C} = \Pi_{\text{both}}(\mathbf{X})$  and  $\sum_{i=1}^{d^*} \sqrt{\lambda_i^*} U_i^* (V_i^*)^T$  is the SVD of  $\mathbf{X} - \mathbf{C}$ . Let us demonstrate that the matrix  $\mathbf{C}$  can be presented as a sum of elementary matrices of rank 1. Then (2.2) will be similar to the result of decomposition (1.12) performed by Basic SSA and therefore the reconstruction stage will be also similar to that of Basic SSA.

Note that  $\Pi_{\text{col}}(\mathbf{Y}) = \mathbf{P}\mathbf{P}^T \mathbf{Y} = \sum_{i=1}^p P_i (\mathbf{Y}^T P_i)^T$  and  $\Pi_{\text{row}}(\mathbf{Y}) = \mathbf{Y}\mathbf{Q}\mathbf{Q}^T = \sum_{i=1}^q (\mathbf{Y} Q_i) Q_i^T$  are decompositions into sums of elementary rank-one matrices. Therefore,  $\mathbf{C} = \Pi_{\text{both}}(\mathbf{X})$  also can be expanded to a sum of elementary matrices, since  $\Pi_{\text{both}}$  can be expressed as a sequential application of the projection operators  $\Pi_{\text{row}}$  and  $\Pi_{\text{col}}$ , see (2.1). For double projection, this expansion depends on the order of projections; for definiteness, we will apply the row projector first.

Thus, the matrix  $\mathbf{C}$  can be considered as a sum of  $p + q$  elementary matrices of the forms  $\sigma_i^{(c)} P_i \tilde{Q}_i^T$ ,  $i = 1, \dots, p$ , and  $\sigma_i^{(r)} \tilde{P}_i Q_i^T$ ,  $i = 1, \dots, q$  (some of them can be zero), where the triples  $(\sigma_i^{(c)}, P_i, \tilde{Q}_i)$  and  $(\sigma_i^{(r)}, \tilde{P}_i, Q_i)$  have the same meaning as eigentriples. Therefore, the decomposition (2.2) can be transformed to a decomposition into a sum of  $d^* + p + q$  elementary rank-one matrices, which are orthogonal with respect to the Frobenius norm  $\|\cdot\|$ , by construction. As a consequence, contribution of the projection term  $\mathbf{C}$  into the decomposition is given by  $\|\mathbf{C}\|^2 / \|\mathbf{X}\|^2$ ;

The following lemma describes properties of the decomposition (2.2).

**Lemma 1.** *The decomposition (2.2) satisfies the following properties:*

1.  $d^* \leq \text{rank } \mathbf{X}$ ;

2.  $d^* \geq \text{rank } \mathbf{X} - (p + q)$ ; the equality holds if  $P_i$ ,  $i = 1, \dots, p$ , belong to the column span of  $\mathbf{X}$  and  $Q_i$ ,  $i = 1, \dots, q$ , belong to the row span of  $\mathbf{X}$ .

*Proof.* Due to the definition of orthogonal projection,  $\mathbf{X}^* = \mathbf{X} - \mathbf{C} = (\mathbf{I}_L - \mathbf{P}\mathbf{P}^T)\mathbf{X}(\mathbf{I}_K - \mathbf{Q}\mathbf{Q}^T)$ , where  $\mathbf{I}_M$  denotes the  $M \times M$  identity matrix. Therefore,  $d^* \leq \text{rank } \mathbf{X}$ . On the other hand,  $\text{rank } \mathbf{X}$  does not exceed the number of elementary matrices  $d^* + p + q$ ; the equality is attained if the rows and columns of the projected matrices belong to the row and column spaces of  $\mathbf{X}$  respectively.  $\square$

## Algorithm

Let us summarize the steps of SSA with projection in the form of algorithms, splitting the whole algorithm into decomposition and reconstruction.

### ALGORITHM 2.1: SSA with projection: decomposition

*Input:* The time series  $\mathbf{X}$  of length  $N$ , the window length  $L$ , an orthonormal basis of the column projection space ( $P_i, i = 1, \dots, p$ ) and an orthonormal basis of the row projection space ( $Q_i, i = 1, \dots, q$ ). Either  $p$  or  $q$  can be zero.

*Output:* Decomposition of the trajectory matrix on elementary matrices  $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d$ , where  $\mathbf{X}_i = \sigma_i U_i V_i^T$  are either zero or rank-one matrices.

- 1: Construct the trajectory matrix  $\mathbf{X} = \mathcal{T}_{\text{SSA}}(\mathbf{X})$ .
- 2: Subtract the row projection:  $\mathbf{X}' = \mathbf{X} - \mathbf{C}_{\text{row}}$ , where

$$\mathbf{C}_{\text{row}} = \Pi_{\text{row}}(\mathbf{X}) = \sum_{i=1}^q \sigma_i^{(r)} \tilde{P}_i Q_i^T,$$

$\sigma_i^{(r)} = \|\mathbf{X} Q_i\|$ ,  $\tilde{P}_i = \mathbf{X} Q_i / \sigma_i^{(r)}$  if  $\sigma_i^{(r)} > 0$ ; otherwise,  $\tilde{P}_i$  is the zero vector.

- 3: Subtract the column projection:  $\mathbf{X}^* = \mathbf{X}' - \mathbf{C}_{\text{col}}$ , where

$$\mathbf{C}_{\text{col}} = \Pi_{\text{col}}(\mathbf{X}') = \sum_{i=1}^p \sigma_i^{(c)} P_i \tilde{Q}_i^T,$$

$\sigma_i^{(c)} = \|\mathbf{X}'^T P_i\|$ ,  $\tilde{Q}_i = \mathbf{X}'^T P_i / \sigma_i^{(c)}$  if  $\sigma_i^{(c)} > 0$ ; otherwise,  $\tilde{Q}_i$  is the zero vector.

- 4: Construct an SVD of the matrix  $\mathbf{X}^*$ :  $\mathbf{X}^* = \sum_{i=1}^{d^*} \mathbf{X}_i^*$ , where  $\mathbf{X}_i^* = \sqrt{\lambda_i^*} U_i^* (V_i^*)^T$ .
- 5: As a result,  $\mathbf{X} = \sum_{i=1}^d \mathbf{X}_i$ , where  $d = p + q + d^*$ ,  $\mathbf{X}_i = \sigma_i^{(r)} \tilde{P}_i Q_i^T$  for  $i = 1, \dots, q$ ,  $\mathbf{X}_{i+q} = \sigma_i^{(c)} P_i \tilde{Q}_i^T$  for  $i = 1, \dots, p$ , and  $\mathbf{X}_{i+p+q} = \sqrt{\lambda_i^*} U_i^* (V_i^*)^T$  for  $i = 1, \dots, d^*$ .

To complete the algorithm of SSA with projection, let us describe the algorithm of Reconstruction stage.

---

ALGORITHM 2.2: SSA with projection: reconstruction

---

*Input:* Decomposition  $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d$  and grouping  $\{1, \dots, d\} = \bigsqcup_{j=1}^m I_j$ , which does not split the first  $p+q$  projection components, where  $q$  and  $p$  are the numbers of row and column projection components.

*Output:* Decomposition of time series on identifiable components  $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_m$ .

- 1: Construct the grouped matrix decomposition  $\mathbf{X} = \mathbf{X}_{I_1} + \dots + \mathbf{X}_{I_m}$ , where  $\mathbf{X}_I = \sum_{i \in I} \mathbf{X}_i$ .
  - 2: Compute  $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_m$ , where  $\mathbf{X}_i = \mathcal{T}^{-1}\mathcal{H}(\mathbf{X}_{I_i})$ .
- 

The only essential difference with the reconstruction by Basic SSA is that the set of the matrices  $\mathbf{X}_i$ ,  $i = 1, \dots, p+q$ , which is produced by projections, should be included in the same group. Then the resultant series decomposition does not depend on the selected bases  $\{P_i, i = 1, \dots, p\}$  and  $\{Q_i, i = 1, \dots, q\}$ .

### Appropriate class of time series

For SSA with projection, the natural question is what series are preserved with projection; that is, for what kinds of series with a trajectory matrix  $\mathbf{X}$  we have  $\Pi_{\text{col}}(\mathbf{X}) = \mathbf{X}$  for column projection,  $\Pi_{\text{row}}(\mathbf{X}) = \mathbf{X}$  for row projection and  $\Pi_{\text{both}}(\mathbf{X}) = \mathbf{X}$  for double projection.

The following lemma is a direct consequence of the definition of projection.

**Lemma 2.** *Let  $\mathcal{L}_{\text{row}}$  contain the row space of a matrix  $\mathbf{X}$  and  $\mathcal{L}_{\text{col}}$  contain the column space of a matrix  $\mathbf{X}$ . Then  $\Pi_{\text{row}}(\mathbf{X}) = \mathbf{X}$  and  $\Pi_{\text{col}}(\mathbf{X}) = \mathbf{X}$ .*

For example, it follows from Lemma 2 that to preserve an exponential series with  $s_n = C\mu^n$  by SSA with column projection, the column projection should be performed to a space which contains  $\text{span}((\mu, \mu^2, \dots, \mu^L)^T)$ , while to preserve a linear function with  $s_n = an + b$  for any  $b$  and non-zero  $a$ , the column projection should be performed to a space which contains  $\text{span}((1, 1, \dots, 1)^T, (1, 2, \dots, L)^T)$ .

Let us derive a condition sufficient for  $\Pi_{\text{both}}(\mathbf{X}) = \mathbf{X}$  to hold for the general case of the double projection.

**Lemma 3.** *Let the columns of a matrix  $\mathbf{W} \in \mathcal{M}_{K,q}$  belong to  $\mathcal{L}_{\text{row}}$  and the columns of a matrix  $\mathbf{S} \in \mathcal{M}_{L,p}$  belong to  $\mathcal{L}_{\text{col}}$ . Then  $\Pi_{\text{both}}(\mathbf{X}) = \mathbf{X}$  for*

$$\mathbf{X} = \tilde{\mathbf{S}}\mathbf{W}^T + \widetilde{\mathbf{W}}\mathbf{S}^T \quad (2.3)$$

for any  $\tilde{\mathbf{S}} \in \mathcal{M}_{L,q}$  and  $\widetilde{\mathbf{W}} \in \mathcal{M}_{K,p}$ .

*Proof.* By the assumption,  $\Pi_{\text{row}}(\mathbf{A}\mathbf{W}^T) = \mathbf{A}\mathbf{W}^T$  for any matrix  $\mathbf{A} \in \mathcal{M}_{t,q}$ , where  $t \geq 1$ , while

$\Pi_{\text{col}}(\mathbf{S}\mathbf{B}^T) = \mathbf{S}\mathbf{B}^T$  for any matrix  $\mathbf{B} \in \mathcal{M}_{t,p}$ . Then

$$\Pi_{\text{both}}(\mathbf{X}) = \widetilde{\mathbf{S}}\mathbf{W}^T + \Pi_{\text{row}}(\widetilde{\mathbf{S}}\mathbf{W}^T) + \mathbf{S}\widetilde{\mathbf{W}}^T + \Pi_{\text{col}}(\widetilde{\mathbf{S}}\mathbf{W}^T) - \Pi_{\text{col}}(\Pi_{\text{row}}(\widetilde{\mathbf{S}}\mathbf{W}^T + \widetilde{\mathbf{S}}\mathbf{W}^T)) = \mathbf{X},$$

since  $\Pi_{\text{col}} \circ \Pi_{\text{row}} \equiv \Pi_{\text{row}} \circ \Pi_{\text{col}}$  by the associativity of matrix multiplication.  $\square$

It is easy to check that the trajectory matrix of a linear series satisfies the conditions of Lemma 3 for the case of double centering. However, for a general case an approach based on characteristic roots is more convenient. We start with a technical lemma.

**Lemma 4.** *For any polynomial  $P_d$  of degree  $d$  and for any  $l$  and  $k$  such that  $l+k=d-1$  the following expansion can be constructed:*

$$P_d(i+j) = P_{l,d}(i,j) + P_{d,k}(i,j),$$

where  $P_{\tau,\varkappa}(i,j)$  denotes a polynomial of  $i$  and  $j$  of bidegree  $(\tau, \varkappa)$ .

*Proof.* This lemma is proved by an appropriate grouping of the monomials  $C_{u,v}i^u j^v$ ,  $u+v \leq d$ , of  $P_d(i+j)$ .  $\square$

Recall that a series governed by an LRR, whose characteristic polynomial has the given set of roots called characteristic roots, is of the form (1.10).

**Theorem 1.** *Let series  $\mathbf{Y}^{(m)}$ ,  $m=1,2$ , be governed by minimal LRRs of orders  $r_m$ ,  $\mathbf{Y}^{(m)}$  be their trajectory matrices. Let  $\{\mu_j; j=1,\dots,s\}$  be the set containing the characteristic roots of both series. Assume that  $\mathbf{Y}^{(m)}$ ,  $m=1,2$ , have the characteristic roots  $\mu_j$ ,  $j=1,\dots,s$ , with multiplicities  $d_j^{(m)} \geq 0$ ,  $\sum_{j=1}^s d_j^{(m)} = r_m$ . Let  $\Pi_{\text{col}}$  be the projector on the column space  $\mathcal{C}$  of  $\mathbf{Y}^{(1)}$ ,  $\Pi_{\text{row}}$  be the projector on the row space  $\mathcal{R}$  of  $\mathbf{Y}^{(2)}$ ,  $\Pi_{\text{both}}$  be given in (2.1). Then  $\Pi_{\text{both}}(\mathbf{X}) = \mathbf{X}$  if and only if the set of characteristic roots of the series  $\mathbf{X}$  consists of the roots  $\mu_j$ ,  $j=1,\dots,s$ , of multiplicities  $d_j \leq d_j^{(1)} + d_j^{(2)}$ .*

*Proof.* Due to linearity of projectors and linear dependence of  $\Pi_{\text{both}}$  on  $\Pi_{\text{row}}$  and  $\Pi_{\text{col}}$ , it is sufficient to prove the theorem for the case of one root  $\mu$ . Let  $\mathbf{Y}^{(1)}$  have the characteristic root  $\mu$  of multiplicity  $p$ ,  $\mathbf{Y}^{(2)}$  have the characteristic root  $\mu$  of multiplicity  $q$ .

Thus, we should prove that  $\Pi_{\text{both}}(\mathbf{X}) = \mathbf{X}$  if and only if the series  $\mathbf{X}$  has the form  $x_k = P_t(k)\mu^k$ , where  $t \leq p+q-1$ . It is sufficient to consider  $t = p+q-1$ .

By Lemma 4

$$P_{p+q-1}(i+j)\mu^{i+j} = P_{p-1,p+q-1}(i,j)\mu^i\mu^j + P_{p+q-1,q-1}(i,j)\mu^i\mu^j.$$

This means that (2.3) holds for  $\mathbf{W} \in \mathcal{M}_{K,q}$  and  $\mathbf{S} \in \mathcal{M}_{L,p}$  such that the column space of  $\mathbf{W}$  coincides with  $\mathcal{R}$  and the column space of  $\mathbf{S}$  coincides with  $\mathcal{C}$ .

Since the dimension of the space of trajectory matrices that are preserved by the projector  $\Pi_{\text{both}}$  is equal to  $r = r_1 + r_2$ , we found all such matrices. This completes the proof.  $\square$

**Corollary 1.** Let  $\mathbf{Y}$  be a series of dimension  $r$ ,  $\mathbf{Y}$  be its trajectory matrix,  $\Pi_{\text{row}}$  be the projector on its row trajectory space,  $\Pi_{\text{col}}$  be the projector on its column trajectory space. Consider the series  $\mathbf{X}$  with  $x_n = (an+b)y_n$ . Then  $\Pi_{\text{both}}(\mathbf{X}) = \mathbf{X}$ .

**Remark 3.** Note that multiplication of a time series by  $an+b$ , where  $a \neq 0$ , means that the multiplicities of the series characteristic roots increase by 1.

Note that formally the sets  $\{P_i, i = 1, \dots, p\}$  and  $\{Q_i, i = 1, \dots, q\}$  can be arbitrary. However, if the model of the series is partly known, then in the context of SSA this means that a time series component satisfies an LRR and we know its characteristic roots (see Section 1.9.2). Therefore, to extract, for example, a sine wave using projections, we should know its period, and to extract an exponential trend, we should know its rate. Such conditions are often too restrictive. A clear exception is extraction of polynomial trends of a degree  $k$ , when there is the unique characteristic root equal to 1 of multiplicity  $k+1$  and we should assume only the degree of the polynomial trend to obtain its trajectory space.

**Corollary 2.** Let  $\Pi_{\text{row}}$  be the projector on the row trajectory space of a polynomial of degree  $l$ ,  $\Pi_{\text{col}}$  be the projection on the column trajectory space of a polynomial of degree  $k$ . Then for any polynomial  $\mathbf{X} = P_{l+k+1}$  of degree  $l+k+1$  we have  $\Pi_{\text{both}}(\mathbf{X}) = \mathbf{X}$ .

**Remark 4.** It immediately follows from Lemma 2 that in the conditions of Corollary 2, for any polynomial  $\mathbf{X} = P_l$  of degree  $l$  we have  $\Pi_{\text{row}}(\mathbf{X}) = \mathbf{X}$  and for any polynomial  $\mathbf{X} = P_k$  of degree  $k$  we have  $\Pi_{\text{col}}(\mathbf{X}) = \mathbf{X}$ .

## Separability

We expect that if a time series component is governed by a minimal LRR and this LRR is known, then the series component can be separated by a suitable version of SSA with projection better than it can be done by Basic SSA.

Using the notion of separability, we can formulate this improvement as follows. Let  $\mathbf{X} = \mathbf{X}^{(1)} + \mathbf{X}^{(2)}$ . We will say that a time series component  $\mathbf{X}^{(1)}$  is separated by SSA with projection if  $\mathbf{X}^{(1)} = \mathbf{C}$ , where  $\mathbf{C}$  is equal to  $\Pi_{\text{row}}(\mathbf{X})$ ,  $\Pi_{\text{col}}(\mathbf{X})$  or  $\Pi_{\text{both}}(\mathbf{X})$ , in dependence on the type of projection.

Let  $\mathbf{X}^{(1)}$  be a series of finite rank,  $\mathbf{X} = \mathbf{X}^{(1)} + \mathbf{X}^{(2)}$ . Similarly to [1, Section 6.3], where conditions for separability by SSA with centering are considered, the following conditions of separability can be obtained.

### 1. Basic SSA:

$\mathbf{X}^{(1)}$  and  $\mathbf{X}^{(2)}$  are separable if (if and only if, by definition) their row and column spaces are orthogonal.

### 2. SSA with row projection on the row space of $\mathbf{X}^{(1)}$ :

$\mathbf{X}^{(1)}$  and  $\mathbf{X}^{(2)}$  are separable if their row spaces are orthogonal.

3. SSA with column projection on the column space of  $\mathbf{X}^{(1)}$ :  
 $\mathbf{X}^{(1)}$  and  $\mathbf{X}^{(2)}$  are separable if their column spaces are orthogonal.
4. SSA with double projection on the row and column space of  $\mathbf{Y}$ , where  $\mathbf{X}^{(1)}$  is expressed through  $\mathbf{Y}$  as  $x_n^{(1)} = (an + b)y_n$ ,  $a \neq 0$ :  
 $\mathbf{X}^{(1)}$  and  $\mathbf{X}^{(2)}$  are separable by SSA with double projection if  $\mathbf{Y}$  and  $\mathbf{X}^{(2)}$  are separable by Basic SSA.

Note that the separability by SSA with projection is always strong, since projections on linear spaces are uniquely defined.

For the approximate separability, where  $\mathbf{X}^{(1)} \approx \mathbf{C}$ , the approximate orthogonality is necessary. Also, the asymptotic separability can be considered by analogy with the conventional separability for Basic SSA and SSA with centering.

Recall that the usual double centering in SSA corresponds to a constant series  $\mathbf{Y}$  and therefore to a linear series  $\mathbf{X}^{(1)}$ . Orthogonality to a constant series is a much weaker condition than that to a linear series (moreover, the condition of orthogonality to a linear series can never be exactly satisfied). In particular, any sinusoid with frequency  $\omega$  is asymptotically separable from the linear trend and the exact separability by SSA with projection takes place if  $L\omega$  and  $K\omega$  are integers, that is, if  $L$  and  $K$  are divisible by the period of the sinusoid. Therefore, for extraction of linear trends, the double centering is recommended.

In the case of a polynomial trend of degree larger than 1, the conditions of exact separability cannot be satisfied at all, even for SSA with double projection. However, we still can expect that in the case of polynomial trends, SSA with double projection also will work better than SSA with only row or column projection and also better than Basic SSA. This will be checked in the next section.

### 2.1.3. Examples

The presented examples are related to finding polynomial trends. For convenience, if the row and column projections are performed on the subspace generated by polynomials of degree  $q - 1$  and  $p - 1$  respectively, then we denote the method as ProjSSA( $q,p$ ). Recall (see Corollary 2 and Remark 4) that the choice ProjSSA( $q,p$ ) corresponds to extraction of a polynomial trend of degree  $q + p - 1$ . The zero value for  $p$  or  $q$  means that the corresponding projection is not performed. For example, both ProjSSA(2,0) and ProjSSA(1,1) can be used for extraction of a linear trend. In ProjSSA( $q,p$ ), the projection part of the decomposition, i.e., the decomposition of the matrix  $\mathbf{C} = \Pi_{\text{both}}(\mathbf{X})$ , consists of  $p + q$  rank-one matrices.

Note that the implementation of SSA with projection in RSSA is efficient, since it uses the approach described in [113]. The computational cost of Decomposition stage for the common case is  $\mathcal{O}((q + p)N \log N + qpN)$  for the projection computation using fast convolution and  $\mathcal{O}((q + p + r + \log N)Nr)$  for the SVD decomposition itself. Here  $r$  is the number of required leading SVD

eigentriples, which is small in practice. The computational cost of Reconstruction stage is exactly the same as that for Basic SSA.

### SSA with projection and regression

Let us demonstrate that the conventional linear regression and SSA with double centering, i.e., ProjSSA(1,1), use different statements of the solved problem and therefore can yield different results. It is clearly seen in short time series. For long time series the results are very close. Also, in the model of linear regression with white noise, the least-squares regression solution is the best linear unbiased estimate, see the Gauss-Markov theorem. Therefore, to demonstrate the difference, we consider a time series, which contains a seasonal component.

Here we examine the time series ‘Gasoline’ taken from [114] and containing the data ‘Gasoline’, which contains gasoline demand, monthly, Jan 1960 – Jun 1967, Ontario, gallon millions.

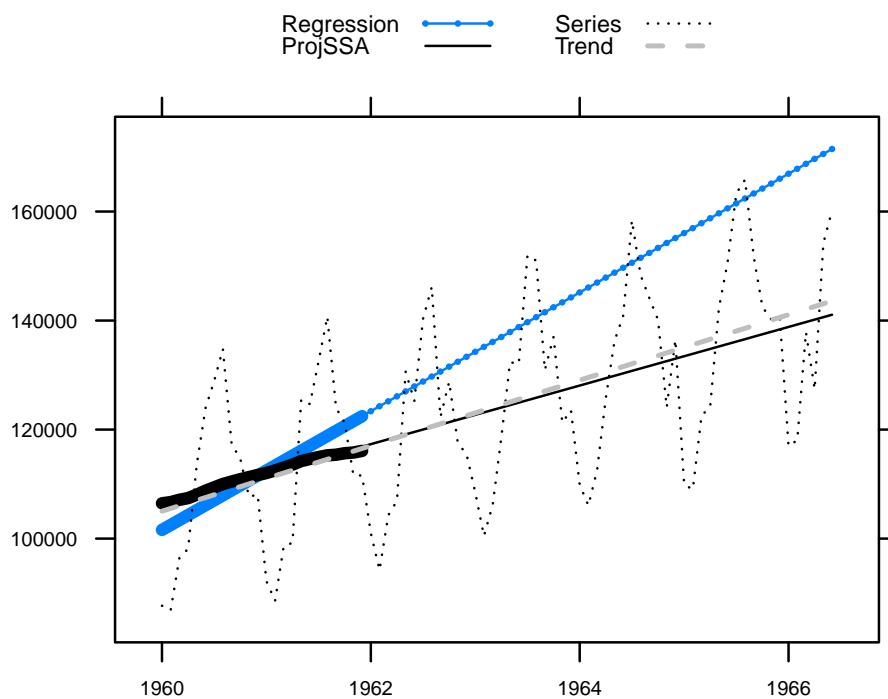


Figure 2.1: ‘Gasoline’: SSA with projection, linear trend detection.

Let us consider the first two years and apply the linear regression and ProjSSA(1,1) with  $L = 12$ . To show the difference, we continue the linear regression line with the help of the estimated coefficients. In the RSSA, a method of forecasting for SSA with projection is implemented. However, since this forecasting method is not thoroughly tested and proved, we do not use it. We will construct the forecast by a linear regression applied to the reconstruction, which is performed by ProjSSA(1,1). Note that the forecasting procedure from RSSA provides a similar prediction. As a benchmark, the linear regression constructed by the whole series is considered.

One can see in Figure 2.1 that the ProjSSA(1,1) linear trend (the black thin line) is very

close to a linear trend constructed by the whole long time series (the grey dash line). The linear regression line (a line with circle points) gives a much worse approximation of the trend. This is explained by the following reasons. The least-squares approach to the linear regression estimation minimizes the prediction error and therefore the seasonal component can shift the linear regression trend. For ProjSSA(1,1), the seasonal component is well separated from the linear trend, since for the chosen parameters  $L = K = 12$  are divisible by the seasonal period 12.

### SSA with projection and Basic SSA

The example introduced in this section demonstrates that both SSA with projection and Basic SSA can extract trends in a similar manner. Let us consider the example ‘co2’ (Mauna Loa Atmospheric CO<sub>2</sub> Concentration, 468 observations, monthly from 1959 to 1997 [115]).

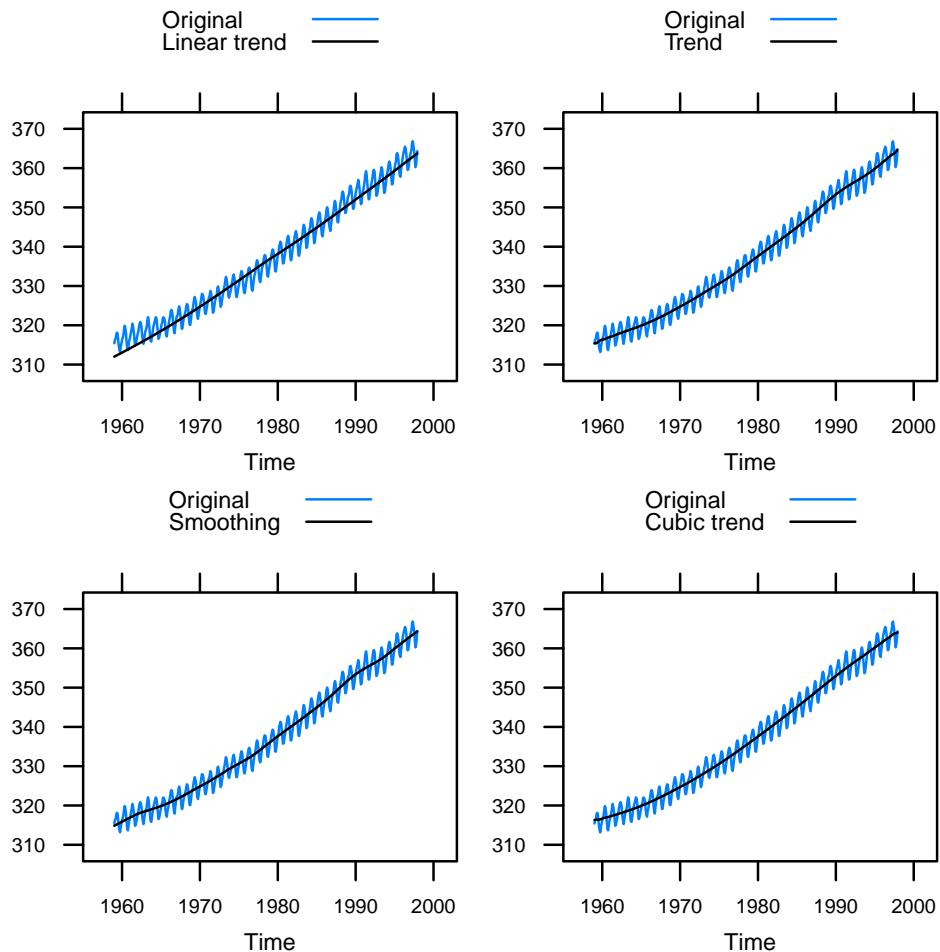


Figure 2.2: ‘co2’: Reconstructions of the trend. Left-top: ProjSSA(1,1),  $L = 228$ ; right-top: ProjSSA(1,1),  $L = 228$ , complemented by the ET 5 and 8; left-bottom: ProjSSA(1,1),  $L = 36$ ; right-bottom: ProjSSA(2,2),  $L = 228$ .

We start with extraction of the linear trend and therefore choose ProjSSA(1,1) to perform SSA with double centering.

By analogy with SSA, large window lengths help to extract separable series components, while small window lengths correspond to smoothing. Therefore, we take  $L = 228$ , which is divisible by 12 and is close to half of the time series length to obtain better separability, and a small value  $L = 36$  to smooth the series. Three of four versions of the extracted trends presented in Figure 2.2 almost coincide.

For the choice  $L = 228$ , the extracted trend is close to linear, see Figure 2.2 (left-top). Certainly, the accurate trend of ‘co2’ series is not linear. However, the projection components can be supplemented by the third and sixth SVD components (ET5,8) to improve the trend (Figure 2.2 (right-top)). Figure 2.2 (left-bottom) shows the result of smoothing with  $L = 36$ . Finally, the result of ProjSSA(2,2) with  $L = 228$ , which is designed for extraction of a cubic trend, is depicted in Figure 2.2 (right-bottom). The extracted trend is very similar to that in [8], which was extracted by Basic SSA (not depicted).

Identification of the components in the decomposition produced by SSA with projection is exactly the same as it is performed in Basic SSA.

## Numerical comparison

The real-life examples presented in Sections 2.1.3 and 2.1.3 show that the results of Basic SSA, SSA with projection and linear regression can be either different or similar. To understand, which method is better, let us perform a numerical study.

We consider a time series of length  $N = 199$  with the common term

$$x_n = t_n + s_n + \varepsilon_n, \quad (2.4)$$

where  $t_n$  is a trend,  $s_n = A \sin(2\pi n \omega + \phi)$ ,  $\varepsilon_n$  is a Gaussian white noise with standard deviation  $\sigma$ .

For obtained estimates  $\hat{t}_n^{(i)}$ , where  $i$  is the number of series with  $i$ th realization of noise  $\varepsilon_n^{(i)}$ ,  $i = 1, \dots, M$ , we will calculate the root-mean-square error (RMSE) as  $\sqrt{\frac{1}{MN} \sum_{i=1}^M \sum_{n=1}^N (\hat{t}_n^{(i)} - t_n)^2}$ .

*Linear trend and sine wave.* Let us start with the noiseless case ( $\sigma = 0$ ) and therefore take  $M = 1$ . Let  $t_n = an + b$ . We fix  $a = 1$ ,  $b = -100$ ,  $A = 1$  and change  $\omega$  from 0.02 to 0.1 (that is, the period is changed from 50 to 10).

Since the result of the least-squares method strongly depends on the form of the residual, we consider the values of the phase,  $\phi = 0$  and  $\phi = \pi/2$ .

Figure 2.3 (left) contains the RMSE values in the case  $\phi = 0$  for Basic SSA with reconstruction by ET1–2, ProjSSA(2,0), ProjSSA(1,1) with  $L = 100$ , and for the linear regression. One can see that the worse cases for ProjSSA(1,1) are approximately equal to the best cases for the linear regression.

In Section 2.1.3, we performed forecasting by the linear regression applied to the trend reconstruction. The same linear regression, which is applied to the trend reconstruction, can be considered as a different trend estimate. Figure 2.3 (right) contains the RMSE for the linear

regression lines constructed in this way; ‘regr’ is added to the legend. Note that the ‘linear regression’ thick line is the same on the left and right plots, which are depicted in different scales. The ordering of the SSA methods is generally the same, while the SSA methods become better than the linear regression. Probably, 0 is one of the worst values of  $\phi$  for linear regression.

Now consider  $\phi = \pi/2$  as one of the best cases for the linear regression. The behavior of the errors is quite different (Figure 2.4 (left)). However, the accuracy of ProjSSA(1,1) is still better than that of the linear regression. Linear least-squares approximation of the SSA reconstructions considerably improves the accuracy of the SSA methods (Figure 2.4 (right)).

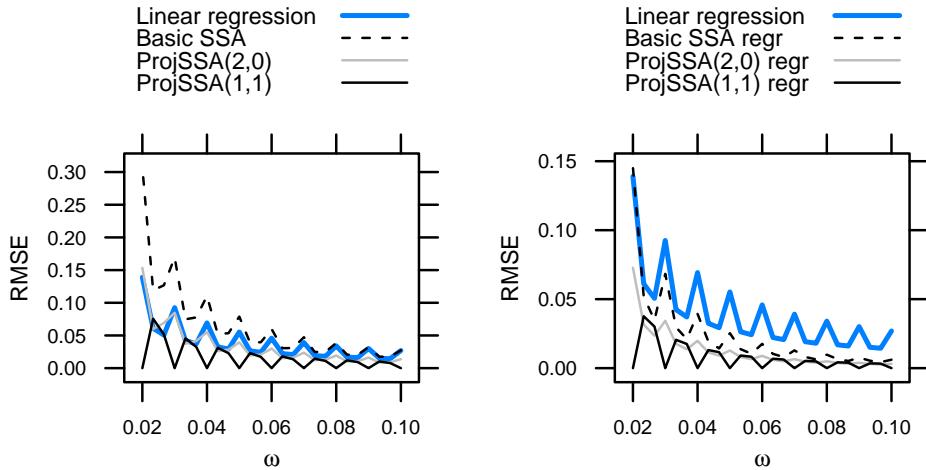


Figure 2.3: Dependence of the RMSE of linear-trend estimates on frequency of the periodic component,  $\phi = 0$ .

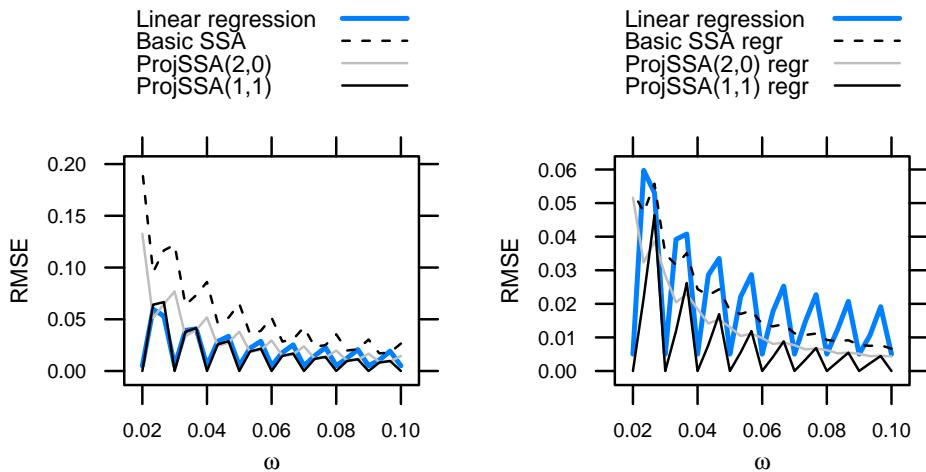


Figure 2.4: Dependence of the RMSE of linear-trend estimates on frequency of the periodic component,  $\phi = \pi/2$ .

Note that zero values of the RMSE for ProjSSA(1,1) for frequencies  $\omega = 0.01k$  are explained by the theory, since then  $L\omega$  and  $K\omega$  are integers. The errors for ProjSSA(2,0) lie between that for

Basic SSA and ProjSSA(1,1). It is interesting that the minimal errors for Basic SSA are achieved for the middle points, when  $L\omega + 0.5$  and  $K\omega + 0.5$  are integers.

**Remark 5.** *The accuracy of separation of a periodic component from a trend depends on (a) is the window length  $L$  small or close to half of the time series length and (b) is  $L$  divisible by the period of the periodic component [1, Section 6.1]. The accuracy is more sensitive to the divisibility (see Section 3.4). Therefore, taking different periods, we check the stability of the comparison with respect to the choice of the window length.*

*Cubic trend and sine wave.* Let us consider a more complex case of the cubic trend  $t_n = 0.0001n^3$ ,  $\sigma = 0$ . Since there is no exact separability for any choice of parameters, the results are unpredictable. Figures 2.5 (left) and 2.6 (left) contain the RMSE values for Basic SSA with reconstruction by ET1–4, ProjSSA(4,0), ProjSSA(2,2) with  $L = 100$  and for the cubic regression. One can see that ProjSSA(2,2) is the best method for  $\phi = 0$ , while it is just comparable with the linear regression for  $\phi = \pi/2$ . Note that here the best parameters for ProjSSA(2,2) do not correspond to the case when  $L\omega$  and  $K\omega$  are integers. The cubic least-squares approximation of the reconstructed trend again improves the estimates (Figures 2.5 (right) and 2.6 (right)).

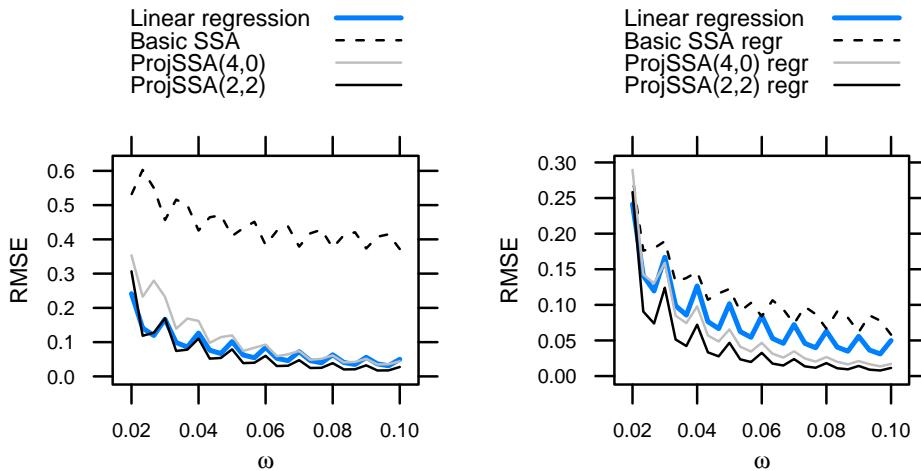


Figure 2.5: Dependence of the RMSE of cubic-trend estimates on frequency of the periodic component,  $\phi = 0$ .

Basic SSA fails for the chosen parameters because of lack of strong separability: the fourth trend component has a contribution comparable with the contribution of the periodic components that causes their mixture.

Note that the modifications Iterative O-SSA described in Section 2.2 can be used to get strong exact separability for the considered noiseless examples. However, we do not involve this modification into the comparison, since Iterative O-SSA is not able to remove noise and should be applied after denoising in the nested manner, while the compared methods are able to extract the trend without denoising.

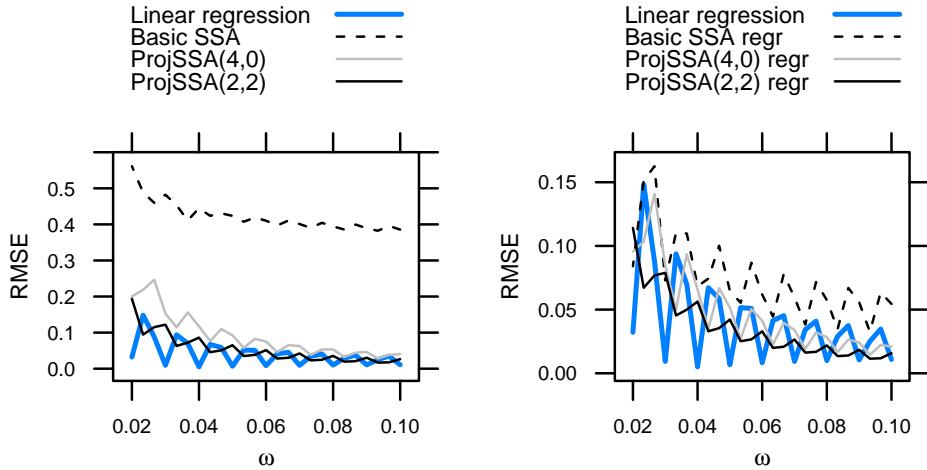


Figure 2.6: Dependence of the RMSE of cubic-trend estimates on frequency of the periodic component,  $\phi = \pi/2$ .

*Linear trend and noise.* For the data which satisfy the model of the linear regression with white Gaussian noise, that is, for the amplitude  $A$  equal to zero, we take  $\sigma = 1$  and use  $M = 1000$ . As expected, the smallest error 0.10 is achieved for the regression estimate. However, the RMSE of the ProjSSA(1,1) estimate equal to 0.12 is very close to 0.10. The error of the Basic SSA is equal to 0.17. Application of linear regression to the results of SSA reconstruction improves the SSA estimates. The RMSE for ProjSSA(1,1) and Basic SSA become equal to 0.115 and 0.104 respectively.

We do not show the errors of trend estimates when the series has both periodic component and noise, since the comparison result (ordering of error values) is intermediate between the cases of a noisy trend with no periodic component and a noiseless trend with added periodicity. The general conclusion is that to keep the advantage of SSA with projection, the noise standard deviation  $\sigma$  should be considerably smaller than the amplitude  $A$  of the periodic component.

#### 2.1.4. Summary of results

The considered applications of SSA with projection were related to the extraction of a polynomial trend, since its trajectory space is determined by the polynomial degree only.

We showed on the example ‘Gasoline’ that the linear regression approach can be inadequate for short series and large oscillations, in comparison with ProjSSA(1,1). Comparison of different SSA versions applied to the ‘co2’ data demonstrates that even if the model of a series component used for projection is wrong, the non-parametric part of SSA with projection can correct the bias.

A numerical study was performed for a better understanding of the difference between SSA with projection and the linear regression approach. First, it appears that if we extract a polynomial trend by SSA with projection, then the polynomial least-squares approximation of the trend reconstruction can considerably improve the accuracy.

The second found effect is related to the influence of the residual geometry on the estimate accuracy. In the considered example, we changed the phase of a sinusoid. The SSA estimates slightly depend on the phase, while the regression estimates demonstrate a considerable dependence.

Numerical experiments confirm that for a linear trend and a sine wave residual, ProjSSA(1,1) is more accurate than the linear regression estimate. For a noisy linear trend, when the model of the linear regression is fulfilled, the linear regression estimate is slightly more accurate than SSA. Thus, we can formulate conditions, when SSA with double projection can be recommended for use: series has a linear or polynomial trend (the polynomial degree is not large) and the regular oscillations are considerably larger than the noise level.

Note that another application of ProjSSA is to involve the structure of a supporting series to improve the reconstruction and prediction of the main time series.

## 2.2. Iterative Oblique SSA

In this section we extend the notion of orthogonality following [9], which allows us to weaken the conditions of weak separability.

### 2.2.1. Oblique SVD and SSA

In this section, we consider the technique of oblique decompositions, which helps to extend the SSA algorithms and improve the separability.

Orthogonality of subseries can be a strong limitation on the separated series. However, if we consider orthogonality with respect to non-standard Euclidean inner product, conditions of separability can be considerably weakened. This yields the first method called Oblique SSA (O-SSA) with the SVD step performed in a non-orthogonal coordinate system.

#### Inner products and related matrix decompositions

Here we provide the necessary information about matrix decompositions with respect to given inner products in the row and column spaces (see e.g. [116, Th.3]), which are called in [117] Restricted SVD (RSVD).

**Inner products** Usually, orthogonality of vectors in  $\mathbb{R}^M$  is considered in a conventional manner:  $X_1$  and  $X_2$  in  $\mathbb{R}^M$  are orthogonal if their Euclidean inner product is equal to 0, i.e.  $(X_1, X_2)_M = 0$ , where  $(\cdot, \cdot)_M$  is the standard inner product in  $\mathbb{R}^M$ . Sometimes we will omit the dimension in denotation if it is clear from the context. It is well-known that any inner product in  $\mathbb{R}^M$  can be defined as  $\langle X_1, X_2 \rangle_{\mathbf{A}} = X_1^T \mathbf{A} X_2$  for a symmetric positive-definite matrix  $\mathbf{A}$ . For any  $\mathbf{O}_{\mathbf{A}}$  such that  $\mathbf{O}_{\mathbf{A}}^T \mathbf{O}_{\mathbf{A}} = \mathbf{A}$  we have  $\langle X_1, X_2 \rangle_{\mathbf{A}} = (\mathbf{O}_{\mathbf{A}} X_1, \mathbf{O}_{\mathbf{A}} X_2)_M$ . Evidently,  $\mathbf{O}_{\mathbf{A}}$  is defined up to multiplication by an orthogonal matrix.

The inner product yields the notion of orthogonality. We will say that two vectors are **A-orthogonal** if  $\langle X_1, X_2 \rangle_{\mathbf{A}} = 0$ .

Let the matrix  $\mathbf{A}$  be symmetric positive semi-definite,  $\text{rank } \mathbf{A} = r$ . Then  $\mathbf{A}$  can be decomposed as  $\mathbf{A} = \mathbf{O}_{\mathbf{A}}^T \mathbf{O}_{\mathbf{A}}$  with  $\mathbf{O}_{\mathbf{A}} \in \mathcal{M}_{r,M}$  and  $\langle X_1, X_2 \rangle_{\mathbf{A}} = (\mathbf{O}_{\mathbf{A}} X_1, \mathbf{O}_{\mathbf{A}} X_2)_r$ . Note that the row space of  $\mathbf{O}_{\mathbf{A}}$  is the same for any choice of  $\mathbf{O}_{\mathbf{A}}$  and coincides with the column space of  $\mathbf{A}$ . If the matrix  $\mathbf{A}$  is not positive definite, then we obtain a degenerate inner product, that is, if  $\langle X, X \rangle_{\mathbf{A}} = 0$ , then it is not necessary that  $X = 0_M$ . However, for vectors belonging to the column space of  $\mathbf{A}$  the equality  $\langle X, X \rangle_{\mathbf{A}} = 0$  yields  $X = 0_M$ . Thus, if we consider inner product generated by a rank-deficient matrix  $\mathbf{A}$ , then we should consider it only on the column space of  $\mathbf{A}$ . In particular, we can correctly define **A-orthogonality** of vectors from the column space of  $\mathbf{A}$ .

The following evident proposition shows that any basis can be considered as **A-orthonormal** for some choice of  $\mathbf{O}_{\mathbf{A}}$ .

**Proposition 5.** *Let  $P_1, \dots, P_r$  be a set of linearly independent vectors in  $\mathbb{R}^M$ . Then  $P_1, \dots, P_r$  are **A-orthonormal** for  $\mathbf{O}_{\mathbf{A}} = \mathbf{P}^\dagger$ , where  $\mathbf{P} = [P_1 : \dots : P_r]$ .*

Note that the column space of  $\mathbf{P}$  coincides with the row space of  $\mathbf{O}_{\mathbf{A}}$ . We call a matrix  $\mathbf{O}_{\mathbf{A}}$  that makes a set  $P_1, \dots, P_r$  **A-orthonormal orthonormalizing matrix** of this set. Certainly, the orthonormalizing matrix is not uniquely defined.

Let us present an elementary example. Let  $X = (1, 2)^T$  and  $Y = (1, 1)^T$ . Certainly, these vectors are not orthogonal in the usual sense:  $(X, Y) = 3$ . However, if we define

$$\mathbf{A} = \begin{pmatrix} 5 & -3 \\ -3 & 2 \end{pmatrix}, \quad (2.5)$$

then  $\langle X, Y \rangle_{\mathbf{A}} = (\mathbf{A}X, Y) = 0$  and  $(\mathbf{O}_{\mathbf{A}}X, \mathbf{O}_{\mathbf{A}}Y) = 0$  for any  $\mathbf{O}_{\mathbf{A}}$  such that  $\mathbf{O}_{\mathbf{A}}^T \mathbf{O}_{\mathbf{A}} = \mathbf{A}$ , e.g.

$$\mathbf{O}_{\mathbf{A}} = \begin{pmatrix} 1 & -1 \\ -2 & 1 \end{pmatrix}.$$

This means that  $\{X, Y\}$  is an orthogonal basis with respect to the **A-inner product**  $\langle \cdot, \cdot \rangle_{\mathbf{A}}$  and  $\mathbf{O}_{\mathbf{A}}$  corresponds to an orthogonalizing map. The matrix  $\mathbf{A}$  can be chosen such that  $X$  and  $Y$  have any **A-norm**. The choice (2.5) corresponds to **A-orthonormality**.

**Oblique decompositions** Let us consider minimal decompositions of  $\mathbf{Y} \in \mathcal{M}_{L,K}$  of rank  $r$  in the form

$$\mathbf{Y} = \sum_{i=1}^r \sigma_i P_i Q_i^T, \quad (2.6)$$

where  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$ ,  $\{P_i\}_{i=1}^r$  and  $\{Q_i\}_{i=1}^r$  are linearly independent (therefore,  $\{P_i\}_{i=1}^r$  is a basis of the column space of  $\mathbf{Y}$ ,  $\{Q_i\}_{i=1}^r$  is a basis of the row space of  $\mathbf{Y}$ ). It is convenient to write (2.6) in the matrix form:  $\mathbf{Y} = \mathbf{P} \boldsymbol{\Sigma} \mathbf{Q}^T$ , where  $\mathbf{P} = [P_1 : \dots : P_r]$ ,  $\mathbf{Q} = [Q_1 : \dots : Q_r]$  and  $\boldsymbol{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_r)$ .

**Proposition 6.** Let  $\mathbf{O}_L$  be an orthonormalizing matrix of  $\{P_i\}_{i=1}^r$  and  $\mathbf{O}_R$  be an orthonormalizing matrix of  $\{Q_i\}_{i=1}^r$ . Then

$$\mathbf{O}_L \mathbf{Y} \mathbf{O}_R^T = \sum_{i=1}^r \sigma_i (\mathbf{O}_L P_i) (\mathbf{O}_R Q_i)^T \quad (2.7)$$

is an SVD of  $\mathbf{O}_L \mathbf{Y} \mathbf{O}_R^T \in \mathcal{M}_{r,r}$  with the left singular vectors  $\mathbf{O}_L P_i \in \mathbb{R}^r$  and the right singular vectors  $\mathbf{O}_R Q_i \in \mathbb{R}^r$ .

This proposition follows from the fact that any bi-orthogonal decomposition is an SVD.

**Definition 5.** If the column space of  $\mathbf{L}$  contains the column space of  $\mathbf{Y}$  and the column space of  $\mathbf{R}$  contains the row space of  $\mathbf{Y}$ , then we will call such a pair  $(\mathbf{L}, \mathbf{R})$  consistent with the matrix  $\mathbf{Y}$ .

**Definition 6.** For  $(\mathbf{L}, \mathbf{R})$  consistent with  $\mathbf{Y}$ , we say that (2.6) is an  $(\mathbf{L}, \mathbf{R})$ -SVD, if the system  $\{P_i\}_{i=1}^r$  is  $\mathbf{L}$ -orthonormal and the system  $\{Q_i\}_{i=1}^r$  is  $\mathbf{R}$ -orthonormal.

In a matrix statement of problem [117], the  $(\mathbf{L}, \mathbf{R})$ -SVD is called Restricted SVD of  $\mathbf{Y}$  with respect to  $(\mathbf{L}, \mathbf{R})$ .

It follows from Definition 6 that (2.7) is an SVD if and only if (2.6) is an  $(\mathbf{L}, \mathbf{R})$ -SVD, where  $\mathbf{L} = \mathbf{O}_L^T \mathbf{O}_L$  and  $\mathbf{R} = \mathbf{O}_R^T \mathbf{O}_R$ ,  $\mathbf{O}_L$  and  $\mathbf{O}_R$  are orthonormalizing.

Proposition 6 says that any minimal decomposition into a sum of matrices of rank 1 in the form (2.6) is the  $(\mathbf{L}, \mathbf{R})$ -SVD for some matrices  $\mathbf{L}$  and  $\mathbf{R}$ .

**Proposition 7.** Let

$$\mathbf{O}_L \mathbf{Y} \mathbf{O}_R^T = \sum_{i=1}^r \sqrt{\lambda_i} U_i V_i^T \quad (2.8)$$

be the ordinary SVD of  $\mathbf{O}_L \mathbf{Y} \mathbf{O}_R^T$ . Then the decomposition (2.6) with  $\sigma_i = \sqrt{\lambda_i}$ ,  $P_i = \mathbf{O}_L^\dagger U_i$  and  $Q_i = \mathbf{O}_R^\dagger V_i$  is the  $(\mathbf{L}, \mathbf{R})$ -SVD.

Proposition 7 follows from Proposition 6 and provides the method how the  $(\mathbf{L}, \mathbf{R})$ -SVD can be calculated (see Algorithm 2.3).

Let us show how we can change the set of  $\sigma_i$  in the  $(\mathbf{L}, \mathbf{R})$ -SVD (2.6) without change of directions of  $P_i$  and  $Q_i$ , that is, of  $P_i/\|P_i\|$  and  $Q_i/\|Q_i\|$ .

**Proposition 8.** Let (2.6) be the  $(\mathbf{L}, \mathbf{R})$ -SVD with  $\mathbf{O}_L = \mathbf{P}^\dagger$  and  $\mathbf{O}_R = \mathbf{Q}^\dagger$ . Then

$$\mathbf{Y} = \sum_{i=1}^r \tilde{\sigma}_i \tilde{P}_i \tilde{Q}_i^T, \quad (2.9)$$

where  $\tilde{\sigma}_i = \sigma_i / (\mu_i v_i)$ ,  $\tilde{P}_i = \mu_i P_i$  and  $\tilde{Q}_i = v_i Q_i$  for some positive  $v_i$  and  $\mu_i$ , is (after reordering of addends according order of  $\tilde{\sigma}_i$ ) the  $(\tilde{\mathbf{L}}, \tilde{\mathbf{R}})$ -SVD with  $\mathbf{O}_{\tilde{\mathbf{L}}} = \tilde{\mathbf{P}}^\dagger$  and  $\mathbf{O}_{\tilde{\mathbf{R}}} = \tilde{\mathbf{Q}}^\dagger$ .

The case of one-side non-orthogonal decompositions, when one of the matrices,  $\mathbf{R}$  or  $\mathbf{L}$ , is identical, is of special concern. It is shown in [117] that then Restricted SVD is Quotient SVD (often

called Generalized SVD [118]). If  $\mathbf{L}$  is the identity matrix, then  $P_i$ ,  $i = 1, \dots, r$ , are orthonormal in the conventional sense and form an orthonormal basis of the column space of  $\mathbf{Y}$ . If  $\mathbf{R}$  is the identity matrix, then  $Q_i$ ,  $i = 1, \dots, r$ , are orthonormal and constitute an orthonormal basis of the row space.

**Matrix scalar products and approximations** Let  $\mathbf{X}, \mathbf{Y} \in \mathcal{M}_{L,K}$ ,  $(\mathbf{L}, \mathbf{R})$  be consistent with both  $\mathbf{X}$  and  $\mathbf{Y}$ ,  $\mathbf{L} = \mathbf{O}_L^T \mathbf{O}_L$  and  $\mathbf{R} = \mathbf{O}_R^T \mathbf{O}_R$ .

Define the induced Frobenius inner product as

$$\langle \mathbf{X}, \mathbf{Y} \rangle_{F,(\mathbf{L},\mathbf{R})} = \langle \mathbf{O}_L \mathbf{X} \mathbf{O}_R^T, \mathbf{O}_L \mathbf{Y} \mathbf{O}_R^T \rangle_F.$$

Note that the definition does not depend on the choice of  $\mathbf{O}_L$  and  $\mathbf{O}_R$ , since  $\langle \mathbf{C}, \mathbf{D} \rangle_F = \text{tr}(\mathbf{C}^T \mathbf{D}) = \text{tr}(\mathbf{C} \mathbf{D}^T)$ .

For two matrices  $\mathbf{C}$  and  $\mathbf{D}$  we say that they

1.  $(\mathbf{L}, \mathbf{R})$  F-orthogonal if  $\langle \mathbf{C}, \mathbf{D} \rangle_{F,(\mathbf{L},\mathbf{R})} = 0$ ,
2.  $(\mathbf{L}, \mathbf{R})$  left-orthogonal if  $\mathbf{C}^T \mathbf{L} \mathbf{D} = \mathbf{0}$ ,
3.  $(\mathbf{L}, \mathbf{R})$  right-orthogonal if  $\mathbf{C} \mathbf{R} \mathbf{D}^T = \mathbf{0}$ ,
4.  $(\mathbf{L}, \mathbf{R})$  bi-orthogonal if the left and right orthogonalities hold.

Left or right orthogonality is the sufficient condition for F-orthogonality. The matrix components of an  $(\mathbf{L}, \mathbf{R})$ -SVD are  $(\mathbf{L}, \mathbf{R})$  bi-orthogonal and therefore  $(\mathbf{L}, \mathbf{R})$  F-orthogonal.

The measure of  $(\mathbf{L}, \mathbf{R})$  orthogonality is

$$\rho_{(\mathbf{L},\mathbf{R})}(\mathbf{X}, \mathbf{Y}) = \frac{\langle \mathbf{X}, \mathbf{Y} \rangle_{F,(\mathbf{L},\mathbf{R})}}{\|\mathbf{X}\|_{F,(\mathbf{L},\mathbf{R})} \|\mathbf{Y}\|_{F,(\mathbf{L},\mathbf{R})}}. \quad (2.10)$$

Let  $\mathbf{X} = \sum_i \mathbf{X}_i$ , where  $\mathbf{X}_i = \sigma_i P_i Q_i^T$ , be the  $(\mathbf{L}, \mathbf{R})$ -SVD. Then  $\|\mathbf{X}_i\|_{F,(\mathbf{L},\mathbf{R})} = \sigma_i$  and  $\|\mathbf{X}\|_{F,(\mathbf{L},\mathbf{R})}^2 = \sum_i \sigma_i^2$ . The  $(\mathbf{L}, \mathbf{R})$ -contribution of  $\mathbf{X}_k$  is equal to  $\sigma_k^2 / \sum_i \sigma_i^2$ .

The following proposition follows from the representation of the Frobenius scalar product through the trace of matrix multiplication.

**Proposition 9.** *If  $\mathbf{X}$  and  $\mathbf{Y}$  are  $(\mathbf{L}, \mathbf{R})$  left-orthogonal, then  $\mathbf{X}$  and  $\mathbf{Y}$  are  $(\mathbf{L}, \widetilde{\mathbf{R}})$  F-orthogonal for any  $\widetilde{\mathbf{R}}$ .*

**Corollary 3.** *Let  $\mathbf{L}$  be the identity matrix and  $\mathbf{X}$  and  $\mathbf{Y}$  be  $(\mathbf{L}, \mathbf{R})$  left-orthogonal for some matrix  $\mathbf{R}$ . Then the conventional F-orthogonality of  $\mathbf{X}$  and  $\mathbf{Y}$  holds and  $\|\mathbf{X} + \mathbf{Y}\|_F^2 = \|\mathbf{X}\|_F^2 + \|\mathbf{Y}\|_F^2$ .*

Corollary 3 shows that if at least in either row or column matrix spaces the conventional inner product is given, that is, vectors are orthogonal in the ordinary sense, then the conventional F-orthogonality can be considered and F-norm and F-inner product can be used to measure the approximation accuracy and the component orthogonality.

**Remark 6.** *The introduced definitions and statements are appropriate if  $\mathbf{L}$  and  $\mathbf{R}$  are consistent with the matrices  $\mathbf{X}$  and  $\mathbf{Y}$  (see Definition 5). Otherwise, e.g., (2.10) can be formally calculated,*

but this measure will reflect only the correlation between projections of columns and rows of  $\mathbf{X}$  and  $\mathbf{Y}$  on the row spaces of  $\mathbf{L}$  and  $\mathbf{R}$  correspondingly.

Let us remark that the conventional Frobenius norm is an interpretable characteristic of approximation, while the norm based on  $\langle \cdot, \cdot \rangle_{F,(\mathbf{L},\mathbf{R})}$  is much worse interpretable, since it is equivalent to approximation by the Frobenius norm of the matrix  $\mathbf{O}_L \mathbf{X} \mathbf{O}_R^T$ .

## Oblique SSA

Although many interpretable series components like trend (a slowly varying component) and seasonality are asymptotically orthogonal, for the given time series length the orthogonality can be not reached even approximately. Therefore, it would be helpful to weaken the orthogonality condition. The suggested approach consists in using an orthogonality, which still means the equality of an inner product to 0, but this is a non-ordinary inner product which is adapted to time series components, which we want to separate.

Consider a scalar scalar product  $\langle X_1, X_2 \rangle_A = (\mathbf{A}X_1, X_2)$  with a symmetric non-negative-defined matrix. Below, considering  $\langle X_1, X_2 \rangle_A$ , we will always assume that the vectors  $X_i$ ,  $i = 1, 2$ , belong to the column space  $\mathbf{A}$ .

To describe a so-called Oblique SSA, we will consider the SVD of a matrix  $\mathbf{X}$  produced by two oblique bases,  $\mathbf{L}$ -orthonormal and  $\mathbf{R}$ -orthonormal correspondingly, in the row and column spaces (Definition 6).

*Oblique SSA (O-SSA)* is the modification of the Basic SSA algorithm described in Section 1.8, where the SVD step is changed by the  $(\mathbf{L}, \mathbf{R})$ -SVD for some matrices  $\mathbf{L}$  and  $\mathbf{R}$  consistent with  $\mathbf{X}$  (see Definition 5). We will use the notions introduced in the algorithm of Basic SSA also for its oblique modification.

Proposition 7 provides Algorithm 2.3 which reduces the  $(\mathbf{L}, \mathbf{R})$ -SVD to the ordinary SVD.

### ALGORITHM 2.3: $(\mathbf{L}, \mathbf{R})$ -SVD.

*Input:*  $\mathbf{Y}$  of rank  $r$ ,  $(\mathbf{L}, \mathbf{R})$  consistent with  $\mathbf{Y}$ .

*Output:* The  $(\mathbf{L}, \mathbf{R})$ -SVD in the form  $\mathbf{Y} = \sum_{i=1}^r \sigma_i P_i Q_i^T$ .

- 1: Calculate  $\mathbf{O}_L$  and  $\mathbf{O}_R$  such that  $\mathbf{O}_L^T \mathbf{O}_L = \mathbf{L}$  and  $\mathbf{O}_R^T \mathbf{O}_R = \mathbf{R}$ .
- 2: Calculate  $\mathbf{O}_L \mathbf{Y} \mathbf{O}_R^T$ .
- 3: Find the ordinary SVD decomposition  $\mathbf{O}_L \mathbf{Y} \mathbf{O}_R^T = \sum_{i=1}^r \sqrt{\lambda_i} U_i V_i^T$ .
- 4:  $\sigma_i = \sqrt{\lambda_i}$ ,  $P_i = \mathbf{O}_L^\dagger U_i$  and  $Q_i = \mathbf{O}_R^\dagger V_i$ . where  $\dagger$  denotes pseudo-inverse.

Note that if  $\mathbf{L}$  and  $\mathbf{R}$  are the identity matrices, then Oblique SSA coincides with Basic SSA,  $\sigma_i = \sqrt{\lambda_i}$ ,  $P_i = U_i$  and  $Q_i = V_i$ .

**Separability** The notion of weak and strong  $(\mathbf{L}, \mathbf{R})$ -separability, which is similar to conventional separability described in Section 1.8.3, can be introduced. Again, let  $\mathbf{X} = \mathbf{X}^{(1)} + \mathbf{X}^{(2)}$ ,  $\mathbf{X}$  be its trajectory matrix,  $\mathbf{X}^{(m)}$  be the trajectory matrices of the series components,  $\mathbf{X}^{(m)} = \sum_{i=1}^{r_m} \sigma_{m,i} P_{m,i} Q_{m,i}^T$  be their  $(\mathbf{L}, \mathbf{R})$ -SVDs,  $m = 1, 2$ . We assume that  $\mathbf{L}$  and  $\mathbf{R}$  are consistent with  $\mathbf{X}$ ,  $\mathbf{X}^{(1)}$  and  $\mathbf{X}^{(2)}$ .

**Definition 7.** Let  $L$  be fixed. Two series  $X_N^{(1)}$  and  $X_N^{(2)}$  are called weakly  $(\mathbf{L}, \mathbf{R})$ -separable, if their column trajectory spaces are  $\mathbf{L}$ -orthogonal and their row trajectory spaces are  $\mathbf{R}$ -orthogonal, that is,  $(\mathbf{X}^{(1)})^T \mathbf{L} \mathbf{X}^{(2)} = \mathbf{0}_{\mathbf{K}, \mathbf{K}}$  and  $\mathbf{X}^{(1)} \mathbf{R} (\mathbf{X}^{(2)})^T = \mathbf{0}_{\mathbf{L}, \mathbf{L}}$ .

**Definition 8.** Two series  $X_N^{(1)}$  and  $X_N^{(2)}$  are called strongly  $(\mathbf{L}, \mathbf{R})$ -separable, if they are weakly  $(\mathbf{L}, \mathbf{R})$ -separable and  $\sigma_{1,i} \neq \sigma_{2,j}$  for any  $i$  and  $j$ .

The  $(\mathbf{L}, \mathbf{R})$ -separability of two series components means  $\mathbf{L}$ -orthogonality of their subseries of length  $L$  and  $\mathbf{R}$ -orthogonality of the subseries of length  $K = N - L + 1$ .

The following theorem shows that the  $(\mathbf{L}, \mathbf{R})$ -separability is in a sense much less restrictive than the ordinary one.

**Theorem 2.** Let  $X = X^{(1)} + X^{(2)}$  be the series of length  $N$ ,  $L$  be the window length and the  $L$ -rank of  $X$  be equal to  $r$ . Let  $X^{(m)}$  be the series of  $L$ -rank  $r_m$ ,  $m = 1, 2$ ,  $r_1 + r_2 = r$ . Then there exist separating matrices  $\mathbf{L} \in \mathcal{M}_{L,L}$  and  $\mathbf{R} \in \mathcal{M}_{K,K}$  of rank  $r$  such that the series  $X^{(1)}$  and  $X^{(2)}$  are strongly  $(\mathbf{L}, \mathbf{R})$ -separable.

*Proof.* Denote  $\{P_i^{(m)}\}_{i=1}^{r_m}$  a basis of the column space of  $\mathbf{X}^{(m)}$  and  $\{Q_i^{(m)}\}_{i=1}^{r_m}$  a basis of the row space of  $\mathbf{X}^{(m)}$ ,  $m = 1, 2$ ; e.g.,  $P_i^{(m)} = P_{m,i} \in \mathbb{R}^L$ ,  $Q_i^{(m)} = Q_{m,i} \in \mathbb{R}^K$ . Define

$$\begin{aligned}\mathbf{P} &= [P_1^{(1)} : \dots : P_{r_1}^{(1)} : P_1^{(2)} : \dots : P_{r_2}^{(2)}], \\ \mathbf{Q} &= [Q_1^{(1)} : \dots : Q_{r_1}^{(1)} : Q_1^{(2)} : \dots : Q_{r_2}^{(2)}].\end{aligned}$$

By the theorem conditions, the matrices  $\mathbf{P}$  and  $\mathbf{Q}$  are of full rank. Since  $\mathbf{P}^\dagger$  and  $\mathbf{Q}^\dagger$  orthonormalize the columns of the matrices  $\mathbf{P}$  and  $\mathbf{Q}$  (Proposition 5), the trajectory matrices  $\mathbf{X}^{(1)}$  and  $\mathbf{X}^{(2)}$  are  $(\mathbf{L}, \mathbf{R})$  bi-orthogonal for  $\mathbf{L} = (\mathbf{P}^\dagger)^T \mathbf{P}^\dagger$  and  $\mathbf{R} = (\mathbf{Q}^\dagger)^T \mathbf{Q}^\dagger$ . Therefore the series  $X^{(1)}$  and  $X^{(2)}$  are  $(\mathbf{L}, \mathbf{R})$ -separable.

Proposition 8 shows that it is possible to change  $\sigma_{m,i}$  keeping bi-orthogonality; that is, this proposition explains how to get strong separability not corrupting the weak one.  $\square$

**Remark 7.** Consider two time series governed by minimal LRRs of orders  $r_1$  and  $r_2$ ,  $r_1 + r_2 \leq \min(L, K)$ . The conditions of Theorem 2 fulfill if and only if the sets of characteristic roots of the series are disjoint. Really, the sets of characteristic roots are disjoint if and only if the column and row spaces of  $L$ -trajectory matrices intersect only in  $\{0\}$ , that is,  $\mathbf{P}$  and  $\mathbf{Q}$  are of full rank.

**Remark 8.** Theorem 2 together with Remark 7 shows that any two times series governed by LRRs with different characteristic roots can be separated by some  $(\mathbf{L}, \mathbf{R})$ -SVD for sufficiently large series and window lengths.

Note that Theorem 2 is not constructive, since the trajectory spaces of the separated series should be known for exact separation. However, we can try to estimate these spaces and thereby to improve the separability.

**Measures of oblique separability.** If Oblique SSA does not separate the components exactly, a measure of separability is necessary. We can consider the analogue of  $\mathbf{w}$ -correlations described in Section 1.8.3, since they are defined through the Frobenius inner products of trajectory matrices and therefore can be generalized; see Section 2.2.1 for definition of  $\rho_{\mathbf{L}, \mathbf{R}}$  in (2.10). Define  $(\mathbf{L}, \mathbf{R})$   $\mathbf{w}$ -correlation between the reconstructed series  $\tilde{\mathbf{X}}^{(1)}$  and  $\tilde{\mathbf{X}}^{(2)}$  as  $\rho_{(\mathbf{L}, \mathbf{R})}(\tilde{\mathbf{X}}^{(1)}, \tilde{\mathbf{X}}^{(2)})$ . Note that due to diagonal averaging, the column and row spaces of  $\tilde{\mathbf{X}}^{(m)}$  do not necessarily belong to the column spaces of  $\mathbf{L}$  and  $\mathbf{R}$  correspondingly, that is, matrices  $\mathbf{L}$  and  $\mathbf{R}$  can be not consistent with  $\tilde{\mathbf{X}}^{(m)}$ ,  $m = 1, 2$ . Therefore,  $\rho_{\mathbf{L}, \mathbf{R}}$  takes into consideration only projections of columns and rows of  $\tilde{\mathbf{X}}^{(1)}$  and  $\tilde{\mathbf{X}}^{(2)}$  on the column spaces of  $\mathbf{L}$  and  $\mathbf{R}$  (Remark 6). This means that  $\rho_{\mathbf{L}, \mathbf{R}}$  can overestimate the separability accuracy.

For Oblique SSA, when only one of the coordinate systems (left or right) is oblique, the conventional  $\mathbf{w}$ -correlations between series are more appropriate measures of separability, since in the case of exact oblique separability we have orthogonal (in the Frobenius inner product) matrix components (Corollary 3).

Another important measure of proper separability is the closeness of the reconstructed series components to time series of finite rank. This can be measured by the contribution of the leading  $r_m = |I_m|$  eigentriples into the SVD of the trajectory matrix  $\tilde{\mathbf{X}}^{(m)}$  of the  $m$ th reconstructed series component  $\tilde{\mathbf{X}}^{(m)}$ . If we denote  $\tilde{\lambda}_{m,i}$  the eigenvalues of the ordinary SVD of  $\tilde{\mathbf{X}}^{(m)}$ , then  $\tau_{r_m}(\tilde{\mathbf{X}}^{(m)}) = 1 - \sum_{i=1}^{r_m} \tilde{\lambda}_{m,i} / \|\tilde{\mathbf{X}}^{(m)}\|^2$  reflects the closeness of the  $m$ th series to the series of rank  $r_m$ .

**Nested Oblique SSA** Rather than the ordinary SVD, the SVD with respect to non-orthogonal coordinate systems provides approximation in an inappropriate way. That is why Oblique SSA cannot be used for extraction of the leading components, in particular, for extraction of the signal and for denoising.

Therefore, the nested way of using Oblique SSA is suggested. The approach is somewhat similar to factor analysis, where a factor space can be estimated by principal component analysis and then interpretable factors are extracted from the factor space.

Suppose that Basic SSA can extract the signal but cannot separate the signal components. For example, let the time series consist of a noisy sum of two sinusoids. Then Basic SSA can perform denoising but probably cannot separate these sinusoids, if their frequencies are close. Thus, Basic SSA is used for estimation of the subspace of the sum of sinusoids and then some other method can be used to separate the sinusoids themselves. The choice of parameters for better separation is thoroughly investigated in Section 3.4.

Thus, let us apply Basic SSA with proper parameters and let a matrix decomposition  $\mathbf{X} = \mathbf{X}_{I_1} + \dots + \mathbf{X}_{I_p}$  be obtained at Grouping step of Basic SSA; each group corresponds to a separated time series component. Let the  $s$ th group  $I = I_s$  be chosen for a refined decomposition.

Denote  $\mathbf{Y} = \mathbf{X}_I$ ,  $r = \text{rank } \mathbf{Y}$ ,  $\mathbf{Y} = \mathcal{T}^{-1}\mathcal{H}\mathbf{Y}$  the series obtained from  $\mathbf{Y}$  by diagonal averaging.

---

ALGORITHM 2.4: Nested Oblique SSA.

---

*Input:* The matrix  $\mathbf{Y}$ , matrices  $(\mathbf{L}, \mathbf{R})$ , which are consistent with  $\mathbf{Y}$  (see Definition 5).

*Output:* a refined series decomposition  $\mathbf{Y} = \tilde{\mathbf{Y}}^{(1)} + \dots + \tilde{\mathbf{Y}}^{(l)}$ .

- 1: Construct an  $(\mathbf{L}, \mathbf{R})$ -SVD of  $\mathbf{Y}$  by Algorithm 2.3 in the form

$$\mathbf{Y} = \sum_{i=1}^r \sigma_i P_i Q_i^T.$$

- 2: Partition the set  $\{1, \dots, r\} = \bigsqcup_{m=1}^l J_m$  and perform grouping to obtain a refined matrix decomposition  $\mathbf{Y} = \mathbf{Y}_{J_1} + \dots + \mathbf{Y}_{J_l}$ .
  - 3: Obtain a refined series decomposition  $\mathbf{Y} = \tilde{\mathbf{Y}}^{(1)} + \dots + \tilde{\mathbf{Y}}^{(l)}$ , where  $\tilde{\mathbf{Y}}^{(m)} = \mathcal{T}^{-1}\mathcal{H}\mathbf{Y}_{J_m}$ .
- 

Thus, after application of Algorithm 2.4 to the group  $I_s$ , we obtain the following decomposition of the series  $\mathbf{X}$ :

$$\mathbf{X} = \tilde{\mathbf{X}}^{(1)} + \dots + \tilde{\mathbf{X}}^{(p)}, \text{ where } \tilde{\mathbf{X}}^{(s)} = \tilde{\mathbf{Y}}^{(1)} + \dots + \tilde{\mathbf{Y}}^{(l)}.$$

For simplicity, below we will consider the case  $l = 2$ .

For reasonably long time series lengths and moderate noise levels, interpretable components such as trends, oscillations and noise are approximately separable by Basic SSA [1, Sections 1.5 and 6.1]. However, the conditions of approximate separability can be restrictive, especially, for short time series.

Orthogonality of subseries, which is the main condition for separability in Basic SSA, see Section 1.8.3, can be a strong limitation on the series which we want to separate. However, if we consider orthogonality with respect to a non-standard Euclidean inner product, conditions of separability are considerably weaker. This approach yields the method called Oblique SSA (O-SSA) with the SVD performed in a non-orthogonal coordinate system at Decomposition step. The idea of Oblique SSA is similar to that of prewhitening which is frequently used in statistics as preprocessing: if we know covariances between components, then we can perform linear transformation and obtain uncorrelated components. Since the ‘covariances’ of the components are not known in advance, an iterative method called Iterative Oblique SSA can be used. Also, the method is able to change contributions of the components in a specific way so that their strong separability will most likely to be improved.

## 2.2.2. Method

### Iterative approach to O-SSA

Let us describe an iterative version of Nested O-SSA; that is, an iterative algorithm for obtaining appropriate matrices  $\mathbf{L}$  and  $\mathbf{R}$  for the  $(\mathbf{L}, \mathbf{R})$ -SVD of  $\mathbf{X}_I$ . For proper use of nested decompositions, we should expect that the matrix  $\mathbf{X}_I$  is close to a rank-deficient trajectory matrix of rank  $r$ .

To explain the main principle of the method, assume that  $\mathbf{X}_I = \mathbf{Y}$  is the trajectory matrix of  $\mathbf{Y}$ . Let  $\mathbf{Y} = \mathbf{Y}^{(1)} + \mathbf{Y}^{(2)}$  and the trajectory matrices  $\mathbf{Y}_1$  and  $\mathbf{Y}_2$  be of ranks  $r_1$  and  $r_2$ ,  $r_1 + r_2 = r$ . Then by Theorem 2, there exist  $r$ -rank separating matrices  $\mathbf{L}^*$ ,  $\mathbf{R}^*$  of sizes  $L \times L$  and  $K \times K$  correspondingly and a partition  $\{1, \dots, r\} = J_1 \sqcup J_2$  such that we can perform the proper grouping in the  $(\mathbf{L}^*, \mathbf{R}^*)$ -SVD and thereby obtain  $\mathbf{Y}_{J_1} = \mathbf{Y}_1$  and  $\mathbf{Y}_{J_2} = \mathbf{Y}_2$ .

The separating matrices  $\mathbf{L}^*$  and  $\mathbf{R}^*$  are unknown as they are determined by unknown trajectory spaces of  $\mathbf{Y}^{(1)}$  and  $\mathbf{Y}^{(2)}$ . Therefore, we want to construct a sequence of  $(\mathbf{L}, \mathbf{R})$ -SVD decompositions (2.6), which in the limit gives the required separating decomposition.

Let us have an initial  $(\mathbf{L}^{(0)}, \mathbf{R}^{(0)})$ -SVD decomposition of  $\mathbf{Y}$  and group its components to obtain some initial estimates  $\tilde{\mathbf{Y}}^{(1,0)}$  and  $\tilde{\mathbf{Y}}^{(2,0)}$  of  $\mathbf{Y}^{(1)}$  and  $\mathbf{Y}^{(2)}$ . Then we can use the trajectory spaces of  $\tilde{\mathbf{Y}}^{(1,0)}$  and  $\tilde{\mathbf{Y}}^{(2,0)}$  to construct the new inner product which is expected to be closer to the separating one. Therefore, we can expect that  $\tilde{\mathbf{Y}}^{(1,1)}$  and  $\tilde{\mathbf{Y}}^{(2,1)}$  will be closer to  $\mathbf{Y}^{(1)}$  and  $\mathbf{Y}^{(2)}$  and therefore we take their trajectory spaces to construct a new inner product; and so on. Of course, if the initial decomposition is strongly separating, then we obtain  $\tilde{\mathbf{Y}}^{(m,1)} = \tilde{\mathbf{Y}}^{(m,0)} = \mathbf{Y}^{(m)}$ ,  $m = 1, 2$ .

### Basic iterative algorithm

We call the iterative version of Nested Oblique SSA *Iterative Oblique SSA* or *Iterative O-SSA*.

As before, we consider a nested O-SSA whose input is the matrix  $\mathbf{Y} = \mathbf{X}_I$  of rank  $r$ . For Basic SSA and for nested O-SSA, a partition of eigentriple numbers for grouping is made after Decompositions stage. For Iterative O-SSA, a partition  $I = \tilde{J}_1 \sqcup \tilde{J}_2$ ,  $r_m = |\tilde{J}_m|$ , should be specified in advance, since iterations are performed in an automatic mode. Certainly, the choice of partition is not made in dark since before the use of a nested version, we have a full decomposition which we use for choosing the group  $I$  and its partition.

Iterative O-SSA is made of repeated application of nested O-SSA with recalculation of  $(\mathbf{L}^{(k)}, \mathbf{R}^{(k)})$ . As any iterative algorithm, Iterative O-SSA should have initial data  $(\mathbf{L}^{(0)}, \mathbf{R}^{(0)})$  and a stopping rule. Standard stopping rule includes the maximum number of iterations  $M$  and the precision threshold  $\varepsilon$ . In Iterative O-SSA, the algorithm stops if the reconstructed series components  $\tilde{\mathbf{Y}}^{(m,k)}$ ,  $m = 1, 2$ , change very little. For a function  $\rho(\cdot)$  defining a vector norm, the iterations stop under the condition  $\max(\rho(\tilde{\mathbf{Y}}^{(m,k)} - \tilde{\mathbf{Y}}^{(m,k-1)}), m = 1, 2) < \varepsilon$ .

Note that since the consistence of  $(\mathbf{L}, \mathbf{R})$  with  $\mathbf{Y}$  is needed for a correct  $(\mathbf{L}, \mathbf{R})$ -SVD, the initial data  $(\mathbf{L}^{(0)}, \mathbf{R}^{(0)})$  should also be consistent.

**Remark 9.** The initial matrices  $(\mathbf{L}^{(0)}, \mathbf{R}^{(0)})$  together with grouping can be specified so that the initial decomposition is a part of the SVD (1.12) given by the set of indices  $I$  and  $I = \tilde{J}_1 \sqcup \tilde{J}_2$ , where  $\tilde{J}_1$  and  $\tilde{J}_2$  are chosen on the base of analysis of the components in (1.12). Since the expansion (1.12) is biorthogonal,  $\mathbf{L}^{(0)}$  and  $\mathbf{R}^{(0)}$  are the identity matrices. It is convenient to denote by  $J_1$  and  $J_2$  the sets consisting of ordinal indices of the elements of  $\tilde{J}_1$  and  $\tilde{J}_2$  in  $I$ . Thereby,  $\{1, \dots, r\} = J_1 \sqcup J_2$ . For example, if  $\tilde{J}_1 = \{11, 14\}$  and  $\tilde{J}_2 = \{12, 18\}$ , then  $I = \{11, 12, 14, 18\}$ ,  $J_1 = \{1, 3\}$  and  $J_2 = \{2, 4\}$ .

To finalize the Iterative O-SSA method, we present a formal description of iterations. The separating decomposition  $\mathbf{Y} = \mathbf{Y}_1 + \mathbf{Y}_2$  should satisfy the following properties:

- (a)  $\mathbf{Y}_1$  and  $\mathbf{Y}_2$  are Hankel;
- (b)  $\text{rank } \mathbf{Y}_1 = r_1$ ,  $\text{rank } \mathbf{Y}_2 = r_2$ ;
- (c) the column and row spaces of  $\mathbf{Y}_1$  and  $\mathbf{Y}_2$  lie in the column and row spaces of  $\mathbf{Y}$ ;
- (d)  $\mathbf{Y}_1$  and  $\mathbf{Y}_2$  are  $(\mathbf{L}, \mathbf{R})$ -biorthogonal for  $\mathbf{L} = \mathbf{L}^*$  and  $\mathbf{R} = \mathbf{R}^*$ .

Define  $\Pi_{\text{col}}$  the orthogonal projection operator (for the Euclidean norm) on the column space of  $\mathbf{Y}$ ,  $\Pi_{\text{row}}$  the projection operator on the row space of  $\mathbf{Y}$ . The nested group is the ordered union  $I = \tilde{J}_1 \sqcup \tilde{J}_2$ ,  $r_m = |\tilde{J}_m|$ ,  $J_1$  and  $J_2$  are defined in Remark 9; the pair of matrices  $(\mathbf{L}^{(k-1)}, \mathbf{R}^{(k-1)})$  is the input for the  $k$ th iteration.

Let us formulate the  $k$ th iteration steps.

- (A) To obtain Hankel matrices, we perform hankelization of the input decomposition  $\tilde{\mathbf{Y}}_m = \Pi_{\mathcal{H}} \mathbf{Y}_{J_m}^{(k-1)}$ ,  $m = 1, 2$ .
- (B) Then, to obtain a low-rank approximation of ranks  $r_1$  and  $r_2$  correspondingly, we construct the ordinary SVDs  $\tilde{\mathbf{Y}}_m = \sum_{i=1}^{d_m} \sqrt{\lambda_i^{(m)}} U_i^{(m)} (V_i^{(m)})^T$ ,  $m = 1, 2$ , and take the leading  $r_m$  terms.
- (C) Since we should not fall outside the column space of the input matrix  $\mathbf{Y}$  (we consider a nested decomposition), we find the projections  $\hat{U}_i^{(m)} = \Pi_{\text{col}} U_i^{(m)}$  and  $\hat{V}_i^{(m)} = \Pi_{\text{row}} V_i^{(m)}$  for  $i = 1, \dots, r_m$ ,  $m = 1, 2$ . Denote

$$\hat{\mathbf{U}}^{(m)} = [\hat{U}_1^{(m)} : \dots : \hat{U}_{r_m}^{(m)}], \quad \hat{\mathbf{V}}^{(m)} = [\hat{V}_1^{(m)} : \dots : \hat{V}_{r_m}^{(m)}].$$

For the algorithm correctness, we assume that the matrices  $\hat{\mathbf{U}}^{(m)}$  and  $\hat{\mathbf{V}}^{(m)}$  are of full rank; otherwise, the algorithm may not work.

- (D) Finally, calculate  $\mathbf{L}^{(k)} = (\hat{\mathbf{U}}^\dagger)^T \hat{\mathbf{U}}^\dagger$  and  $\mathbf{R}^{(k)} = (\hat{\mathbf{V}}^\dagger)^T \hat{\mathbf{V}}^\dagger$ , where  $\hat{\mathbf{U}} = [\hat{\mathbf{U}}^{(1)} : \hat{\mathbf{U}}^{(2)}]$  and  $\hat{\mathbf{V}} = [\hat{\mathbf{V}}^{(1)} : \hat{\mathbf{V}}^{(2)}]$ , to achieve the  $(\mathbf{L}^{(k)}, \mathbf{R}^{(k)})$ -biorthogonality.

The convergence of  $\tilde{\mathbf{Y}}^{(1,k)}$  and  $\tilde{\mathbf{Y}}^{(2,k)}$  to a proper decomposition is not proved theoretically. However, looking at the construction scheme, which resembles the alternating projections, we do expect

this convergence, at least if the case chosen is not too unusual. Numerical experiments confirm the convergence in the majority of examples. Note also that Iterative O-SSA does not change the separating decomposition; that is, the separating decomposition is a fixed point of the algorithm.

### Modification with sigma-correction

If the initial point  $(\mathbf{L}^{(0)}, \mathbf{R}^{(0)})$  for iterations is not far from the separating pair  $(\mathbf{L}^*, \mathbf{R}^*)$ , then we can expect that the convergence in the algorithm above will take place, since we are close to the fixed-point value and we can expect that  $\sigma_i^{(k)}$  in the  $(\mathbf{L}^{(k)}, \mathbf{R}^{(k)})$ -SVDs  $\mathbf{Y} = \sum_{i=1}^r \sigma_i^{(k)} P_i^{(k)} (Q_i^{(k)})^\top$  are just slightly changed during iterations. In general, however, a possible reordering of the decomposition components between iterations of Iterative O-SSA can interfere with convergence. The case of  $J_1 = \{1, \dots, r_1\}$ , when the minimal singular value  $\sigma_{r_1}$  of the first series is kept significantly larger than the maximal singular value  $\sigma_{r_1+1}$  of the second series, would prevent the component reordering and hence improve the convergence.

Let us describe a modification of Iterative O-SSA that provides reordering of the components, moves them apart and thereby relaxes the problem of component mixing. In this modification, an adjustment is made for calculation of  $\widehat{\mathbf{U}}^{(2)}$  and  $\widehat{\mathbf{V}}^{(2)}$  at Step (C) of iterations.

Let us choose a parameter  $\varkappa > 1$ . If  $\lambda_{r_1}^{(1)} < \varkappa^2 \lambda_1^{(2)}$  at Step (C), then define  $\mu = \varkappa \sqrt{\lambda_1^{(2)} / \lambda_{r_1}^{(1)}}$  and change  $\widehat{\mathbf{U}}^{(2)} \leftarrow \sqrt{\mu} \widehat{\mathbf{U}}^{(2)}$ ,  $\widehat{\mathbf{V}}^{(2)} \leftarrow \sqrt{\mu} \widehat{\mathbf{V}}^{(2)}$ . To be consistent with the reordering, set  $J_1 = \{1, \dots, r_1\}$ ,  $J_2 = \{r_1 + 1, \dots, r\}$ .

Note that if  $\lambda_{r_1}^{(1)} < \varkappa^2 \lambda_1^{(2)}$ , then the adjustment above makes a change in the order of the matrix components in (2.11), since they are ordered by  $\sigma_i^{(k)}$ . Hence we force an increase of the matrix components related to the first series component. For explanation of how this sigma-correction works, see Proposition 8.

**Remark 10.** *The reordering procedure is made by sequential adjustment of the component weights and therefore depends on the component enumeration.*

Summarizing, the described correction can help to improve convergence and to provide strong separability of components in the case when only weak separability takes place.

#### 2.2.3. Algorithms

Let us present the method described in Section 2.2 in the form of algorithms. The method consists of different parts and therefore we describe it as several algorithms.

Now we formulate the Iterative O-SSA algorithm. Denote  $\mathbf{Y} = \mathbf{X}_I$ ,  $r = \text{rank } \mathbf{Y}$ ,  $\mathbf{Y} = \mathcal{T}^{-1} \circ \Pi_{\mathcal{H}}(\mathbf{Y})$  the series obtained from  $\mathbf{Y}$  by the diagonal averaging.

We separate the whole algorithm into two parts. Algorithm 2.5 shows a general scheme of Iterative O-SSA, but it does not show how to calculate the pair of matrices  $(\mathbf{L}^{(k)}, \mathbf{R}^{(k)})$  at each iteration. Algorithm 2.6 covers this gap.

---

ALGORITHM 2.5: Iterative O-SSA

---

*Input:* Decomposition of the  $L$ -trajectory matrix  $\mathbf{X} = \sum_{i=1}^d \sigma_i P_i Q_i^T$  of the series  $X$ ; disjoint sets of indices  $\tilde{J}_1$  and  $\tilde{J}_2$  from  $\{1, \dots, d\}$ ; the accuracy tolerance  $\varepsilon$ ; function  $\rho$  for calculating the accuracy; the maximal number of iterations  $M$ ; initial matrices  $(\mathbf{L}^{(0)}, \mathbf{R}^{(0)})$  consistent with  $\mathbf{Y} = \mathbf{X}_I$ . The set  $I = \{i_1, \dots, i_r\}$  is defined as  $I = \tilde{J}_1 \sqcup \tilde{J}_2$ ,  $r_m = |\tilde{J}_m|$ ,  $r = |I| = r_1 + r_2$ , the sets  $J_1$  and  $J_2$  are defined in Remark 9. This partition produces the decompositions for the matrices and series:  $\mathbf{Y} = \mathbf{Y}_{J_1}^{(0)} + \mathbf{Y}_{J_2}^{(0)}$  and  $\mathbf{Y} = \tilde{\mathbf{Y}}^{(1,0)} + \tilde{\mathbf{Y}}^{(2,0)}$ .

*Output:*  $\mathbf{Y} = \tilde{\mathbf{Y}}^{(1)} + \tilde{\mathbf{Y}}^{(2)}$ .

- 1: Set  $k = 1$ .
- 2: Call Algorithm 2.6 for calculation of  $(\mathbf{L}^{(k)}, \mathbf{R}^{(k)})$  consistent with  $\mathbf{Y}$ .
- 3: Compute the  $(\mathbf{L}^{(k)}, \mathbf{R}^{(k)})$ -SVD of  $\mathbf{Y}$  by Algorithm 2.3:

$$\mathbf{Y} = \sum_{i=1}^r \sigma_i^{(k)} P_i^{(k)} (Q_i^{(k)})^T = \mathbf{Y}_{J_1}^{(k)} + \mathbf{Y}_{J_2}^{(k)}. \quad (2.11)$$

- 4: Obtain the decomposition of the series  $\mathbf{Y} = \tilde{\mathbf{Y}}^{(1,k)} + \tilde{\mathbf{Y}}^{(2,k)}$ , where  $\tilde{\mathbf{Y}}^{(m,k)} = \mathcal{T}^{-1} \circ \Pi_{\mathcal{H}}(\mathbf{Y}_{J_m}^{(k)})$ ,  $m = 1, 2$ .
  - 5: If  $k \geq M$  or  $\max(\rho(\tilde{\mathbf{Y}}^{(m,k)} - \tilde{\mathbf{Y}}^{(m,k-1)}), m = 1, 2) < \varepsilon$ , then  $\tilde{\mathbf{Y}}^{(m)} \leftarrow \tilde{\mathbf{Y}}^{(m,k)}$ ,  $m = 1, 2$ , and STOP; else  $k \leftarrow k + 1$  and go to Step 2.
- 

Algorithm 2.6 presents the iteration itself, including the sigma-correction, which may be useful for achieving the strong separability.

---

ALGORITHM 2.6: Calculation of  $(\mathbf{L}^{(k)}, \mathbf{R}^{(k)})$ 


---

*Input:* Partition  $\{1, \dots, r\} = J_1 \sqcup J_2$ ;  $r_m = |J_m|$ ; pair of matrices  $(\mathbf{L}^{(k-1)}, \mathbf{R}^{(k-1)})$ ; parameter for sigma-correction  $\varkappa > 1$  (if  $\varkappa = 0$ , then the sigma-correction is not performed).

*Output:* Pair of matrices  $(\mathbf{L}^{(k)}, \mathbf{R}^{(k)})$  for  $k$ th iteration.

- 1: Calculate  $\tilde{\mathbf{Y}}_m = \Pi_{\mathcal{H}} \mathbf{Y}_{J_m}^{(k-1)}$ ,  $m = 1, 2$ .
- 2: Construct the ordinary SVDs:

$$\tilde{\mathbf{Y}}_m = \sum_{i=1}^{d_m} \sqrt{\lambda_i^{(m)}} U_i^{(m)} (V_i^{(m)})^T, \quad m = 1, 2,$$

(we need the first  $r_m$  terms only).

- 3: Sigma-correction (if  $\varkappa \neq 0$ ): If  $\lambda_{r_1}^{(1)} < \varkappa^2 \lambda_1^{(2)}$ , then define  $\mu = \varkappa \sqrt{\lambda_1^{(2)} / \lambda_{r_1}^{(1)}}$  and change  $\widehat{\mathbf{U}}^{(2)} \leftarrow \sqrt{\mu} \widehat{\mathbf{U}}^{(2)}$ ,  $\widehat{\mathbf{V}}^{(2)} \leftarrow \sqrt{\mu} \widehat{\mathbf{V}}^{(2)}$ . In view of reordering, set  $J_1 = \{1, \dots, r_1\}$ ,  $J_2 = \{r_1 + 1, \dots, r\}$ .

- 4: Find the projections  $\widehat{U}_i^{(m)} = \Pi_{\text{col}} U_i^{(m)}$  and  $\widehat{V}_i^{(m)} = \Pi_{\text{row}} V_i^{(m)}$  for  $i = 1, \dots, r_m$ ,  $m = 1, 2$ . Denote  

$$\widehat{\mathbf{U}}^{(m)} = [\widehat{U}_1^{(m)} : \dots : \widehat{U}_{r_m}^{(m)}], \quad \widehat{\mathbf{V}}^{(m)} = [\widehat{V}_1^{(m)} : \dots : \widehat{V}_{r_m}^{(m)}].$$
  
5: Calculate  $\mathbf{L}^{(k)} = (\widehat{\mathbf{U}}^\dagger)^T \widehat{\mathbf{U}}^\dagger$  and  $\mathbf{R}^{(k)} = (\widehat{\mathbf{V}}^\dagger)^T \widehat{\mathbf{V}}^\dagger$ , where  $\widehat{\mathbf{U}} = [\widehat{\mathbf{U}}^{(1)} : \widehat{\mathbf{U}}^{(2)}]$  and  $\widehat{\mathbf{V}} = [\widehat{\mathbf{V}}^{(1)} : \widehat{\mathbf{V}}^{(2)}]$ .
- 

**Remark 11.** Algorithm 2.5, which uses the sigma-correction, may change the groups of indices. The new groups in Algorithm 2.6 are constructed in such a way that  $J_1$  and  $J_2$  partition the set  $\{1, \dots, r\}$ . The new partition of  $I$  is obtained as  $\tilde{J}_1 = \{i_k \in I : k \in J_1\}$  and  $\tilde{J}_2 = \{i_k \in I : k \in J_2\}$ .

**Remark 12.** Algorithm 2.5 describes a refined decomposition of the matrix  $\mathbf{X}_I$ . However, we can consider Iterative O-SSA as an algorithm, where the full decomposition of the trajectory matrix  $\mathbf{X}$  of an original series  $X$  is used (which changes components from the group  $I$ ). The result would also be a full decomposition.

#### 2.2.4. Example. Separability of sine waves with close frequencies

**Noise-free cases** Let us consider the sum of two sinusoids  $x_n = \sin(2\pi\omega_1 n) + A \sin(2\pi\omega_2 n)$ ,  $n = 1, \dots, N$ ,  $N = 150$ , with close frequencies  $\omega_1 = 0.065$  and  $\omega_2 = 0.06$  and unequal amplitudes, 1 and  $A = 1.2$ . Let the window length  $L = 70$ . Since sinusoids with such close frequencies are far from being orthogonal for the considered window and series lengths, Basic SSA cannot separate them, see Fig. 2.7 (top) where the result of the Basic SSA decomposition is depicted.

To separate the sinusoids we apply the Iterative O-SSA algorithm (Algorithm 2.5) with no sigma-correction,  $\epsilon = 10^{-5}$  and two groups ET1–2 and ET3–4. The maximal number  $M$  of iterations was taken to be very large and therefore was not reached. Decomposition after Iterative O-SSA is depicted in Fig. 2.7 (bottom).

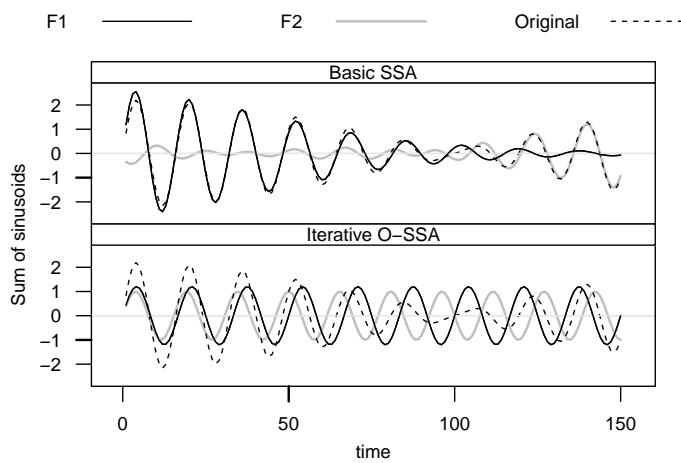


Figure 2.7: Sum of two sinusoids with close frequencies: decomposition by Basic SSA and Iterative O-SSA.

Let us apply the measures of separability described in Section 2.2.1. Note that the conventional

**w**-correlations do not reflect the quality of decomposition. For the initial decomposition we have 0.08. After Iterative O-SSA the **w**-correlation becomes to be equal to  $-0.44$ , while  $(\mathbf{L}, \mathbf{R})$  **w**-correlation is almost 0. The last result confirms that the method separates harmonics exactly. Other measure of true decomposition is the closeness of the components to series of finite ranks. Since the ranks should be equal to the number of the components in the chosen groups, we can calculate the proportion of the corresponding number of the leading components in their SVD decompositions. The mean proportion  $0.5(\tau_{r_1}(\mathbf{X}^{(1)}) + \tau_{r_2}(\mathbf{X}^{(2)}))$  is changed from 0.06 to almost 0.

Let us fix  $\omega_2 = 0.06$ . Then for  $\omega_1 = 0.065$  the algorithm stops after 113 iterations, for  $\omega_1 = 0.07$  the number of iterations is equal to 26, for  $\omega_1 = 0.08$  it is equal to just 6; see thick gray line in Fig. 2.9 (top).

Note that we do not need to use the sigma-correction, since the sinusoids have different amplitudes.

If we consider equal amplitudes with  $A = 1$  and take  $\varkappa = 2$  (Algorithms 2.5 and 2.6), then Iterative O-SSA still converges even for  $\omega_2 = 0.065$  (191 iterations) to the true solution.

**Nested separability in presence of noise** Let us add noise to the sum of two sinusoids and take  $x_n = \sin(2\pi\omega_1 n) + A \sin(2\pi\omega_2 n) + \delta \varepsilon_n$  with close frequencies  $\omega_1 = 0.07$  and  $\omega_2 = 0.06$  and unequal amplitudes, 1 and  $A = 1.2$ . Here  $\varepsilon_n$  is white Gaussian noise with variance 1,  $\delta = 1$ . Let again  $N = 150$ ,  $L = 70$ .

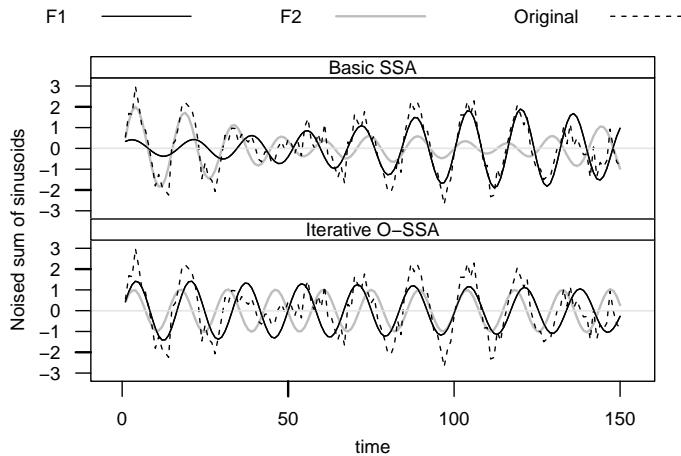


Figure 2.8: Noisy sum of two sinusoids with close frequencies: decomposition by Basic SSA and Iterative O-SSA.

Basic SSA well separates the sinusoids from noise, but cannot separate these sinusoids themselves. Thus, Iterative O-SSA applied to the estimated signal subspace should be used. We use the sigma-correction with  $\varkappa = 2$ , since the difference between amplitudes, 1 and 1.2, appears to be small for strong separability in presence of noise. As before, we set the initial grouping ET1–2 and ET3–4.

The decomposition by Basic SSA at top and by Iterative O-SSA at bottom is depicted in Fig. 2.8. The number of iterations is equal to 32, what is just slightly larger than 26 in the noiseless

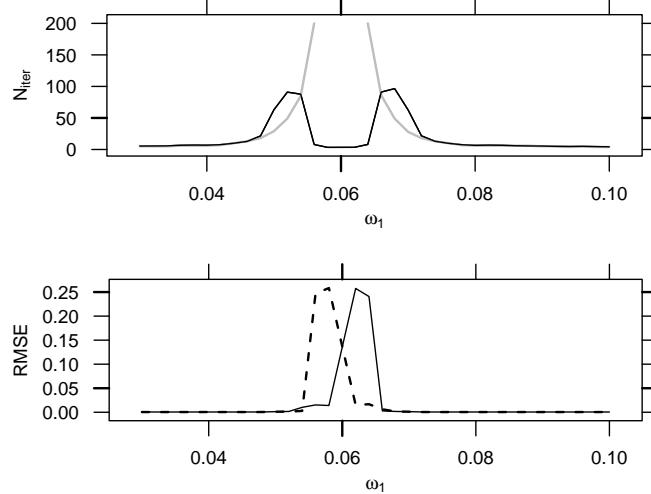


Figure 2.9: Dependence of number of iterations (top) and RMSE errors of frequency estimations (bottom) on  $\omega_1$  for  $\omega_2 = 0.6$ .

case.

Let us investigate the dependence of number of iterations on  $\omega_1$  with the fixed  $\omega_2 = 0.06$ . We change  $\omega_1$  from 0.03 to 0.059 and from 0.061 to 0.1. Fig. 2.9 (top) shows the number of iterations for noiseless signal (thick gray line) and the estimated mean number of iterations for the noisy signal (thin black line); the number of repetitions equals 1000, 5% winsorized estimates of means were calculated. Note that the number of iterations was limited by 200, although for the pure signal convergence held for each  $\omega_1$  from the considered set. A surprisingly small number of iterations for the noisy signal and close frequencies is explained by convergence to a wrong limit, see Fig. 2.9 (bottom) with root mean square errors of LS-ESPRIT estimates for  $\omega_1$  and  $\omega_2$  based on the subspaces spanned by eigenvectors from ET1–2 and ET3–4 (see, e.g., [108] or [29, Section 2.8.2] for the ESPRIT algorithms; also, ESPRIT is briefly described in Section 1.9.2). Since we use the nested decomposition, the noise slightly influences the reconstruction accuracy for frequencies that are quite different ( $\omega_1$  smaller than 0.048 and larger than 0.072).

## 2.3. Filter-adjusted O-SSA and SSA with derivatives

In this section, we describe a variation of SSA that helps to overcome the problem of lack of strong separability if weak separability holds, following [9].

### 2.3.1. SSA with derivatives. Variation for strong separability

Recall that the lack of strong separability of two series components is caused by equal singular values in the sets of the singular values generated by each of time series. In turn, the singular values depends on coefficients  $A_1$  and  $A_2$  before the series components in the sum  $A_1 s_n^{(1)} + A_2 s_n^{(2)}$ . The question is how to change the coefficients  $A_1$  and  $A_2$  in conditions of unknown  $s_n^{(1)}$  and  $s_n^{(2)}$  to

make the singular values different.

It seems that the most natural approach is to use the derivative of the time series in order to change the coefficients and not to change the component subspaces. For example, if  $x_n = A \sin(2\pi\omega n + \phi)$ , then  $x'_n = 2\pi\omega A \cos(2\pi\omega n + \phi)$ , that is, the coefficient  $A' = 2\pi\omega A$ . If we take two sinusoids with different frequencies, then derivation changes their amplitudes differently. For  $x_n = Ae^{\alpha n}$ , derivation also changes the coefficient before the exponential, since  $x'_n = \alpha Ae^{\alpha n}$ , and preserves the rate. For most of the series of finite rank, the derivative subspace coincides with the series subspace. The exception is the polynomial series, when the derivative subspace is a subset of the initial subspace.

Certainly, since we deal with discrete time, we consider  $\varphi_n(\mathbf{X}) = x_{n+1} - x_n$  instead of derivative. However, the approach of taking differences works. For example, for series  $\mathbf{X} = \mathbf{X}_N$  of length  $N$  with  $x_n = A \sin(2\pi\omega n + \phi)$ , we obtain the series  $\Phi_{N-1}(\mathbf{X}) = (\varphi_1(\mathbf{X}), \dots, \varphi_{N-1}(\mathbf{X}))$  of length  $N-1$  with  $\varphi_n(\mathbf{X}) = 2 \sin(\pi\omega) A \cos(2\pi\omega n + \pi\omega + \phi)$ ; for  $x_n = Ae^{\alpha n}$ , we obtain  $\varphi_n(\mathbf{X}) = (e^\alpha - 1)Ae^{\alpha n}$ .

Thus, we can combine the initial series and its derivative to imbalance the component contribution and therefore to obtain their strong separability. For sinusoids, the smaller the period, the larger the increase of the sinusoid amplitude. Therefore, derivation increases the contribution of high frequencies. This effect can increase the level of the noise component, if the series is corrupted by noise. Hence, the nested version of the method implementation should be produced; in particular, the noise component should be removed by Basic SSA in advance.

**Remark 13.** *The approach involving derivatives (that is, sequential differences) can be naturally extended to considering an arbitrary linear filtration  $\varphi$  instead of taking sequential differences. In this paper we deal with derivatives, since this particular case is simple and has very useful applications.*

In the next section, we consider the initial series and its derivative together as two series, regulating the contribution of the derivative, and then apply the multivariate version of SSA. After that, this approach is transformed to a special nested version of Oblique SSA called DerivSSA.

## SSA with derivatives as MSSA

Let us consider the system of two time series  $(\mathbf{X}_N, \gamma\Phi_{N-1}(\mathbf{X}))$  and apply Multivariate SSA (MSSA). The MSSA algorithm was briefly mentioned in Section 1.3.1 and will be described in detail in Chapter 4.

In MSSA, the embedding operator  $\mathcal{T}$  transfers two time series  $(\mathbf{X}_{N_1}, \mathbf{Y}_{N_2})$  to the stacked  $L$ -trajectory matrix  $[\mathbf{X} : \mathbf{Y}]$ . That is, the only difference with Basic SSA consists in the construction of the embedding operator  $\mathcal{T}$ .

Let  $\mathbf{X}_N = \mathbf{X}_N^{(1)} + \mathbf{X}_N^{(2)}$ , where  $\mathbf{X}_N^{(1)}$  and  $\mathbf{X}_N^{(2)}$  are of finite rank and approximately separable. Therefore their row and column trajectory spaces are approximately orthogonal. Then the same is valid for  $\Phi_{N-1}(\mathbf{X}^{(1)})$  and  $\Phi_{N-1}(\mathbf{X}^{(2)})$  in view of the fact that their column spaces belong to the column spaces of  $\mathbf{X}_N^{(1)}$  and  $\mathbf{X}_N^{(2)}$ , while their row spaces are spanned by vectors of the same

structure as the vectors constituting bases of the row spaces of  $\mathbf{X}_N^{(1)}$  and  $\mathbf{X}_N^{(2)}$ , except for these basis vectors have length  $K - 1$  instead of  $K$ . Therefore, approximate orthogonality still holds. Since  $\Phi_{N-1}(\mathbf{X}) = \Phi_{N-1}(\mathbf{X}^{(1)}) + \Phi_{N-1}(\mathbf{X}^{(2)})$ , MSSA applied to  $(\mathbf{X}_N, \gamma\Phi_{N-1}(\mathbf{X}))$  will approximately separate the time series  $\mathbf{X}_N^{(1)}$  and  $\mathbf{X}_N^{(2)}$ . Certainly, we will not have exact separability; however, it is not so important for practice.

As it was mentioned before, a drawback of the described approach is that the method cannot be applied to noisy series, since it intensifies high-frequency harmonics and therefore strengthens noise. Therefore, denoising should be applied as preprocessing. Also, SSA involving derivatives changes component contributions (this is what we want) but simultaneously the method loses approximation features. These reasons lead to the necessity to use the nested way of decomposition introduced in Section 2.2.1.

### Nested SSA with derivatives (DerivSSA)

Let us formulate the nested version of SSA with derivatives called DerivSSA. As well as in Section 2.2.1, let  $\mathbf{Y} = \mathbf{X}_I$  be one of matrices in the decomposition  $\mathbf{X} = \mathbf{X}_{I_1} + \dots + \mathbf{X}_{I_p}$  obtained at Grouping step of Basic SSA; each group corresponds to a separated time series component and we want to construct a refined decomposition of  $\mathbf{Y}$ . As before, denote  $r = \text{rank } \mathbf{Y}$ ,  $\mathbf{Y} = \mathcal{T}^{-1}\mathcal{H}\mathbf{Y}$ .

#### ALGORITHM 2.7: DerivSSA.

*Input:* The matrix  $\mathbf{Y}$ , the weight of derivative  $\gamma > 0$ .

*Output:* a refined series decomposition  $\mathbf{Y} = \tilde{\mathbf{Y}}^{(1)} + \dots + \tilde{\mathbf{Y}}^{(l)}$ .

- 1: Denote  $\Phi(\mathbf{Y}) = [Y_2 - Y_1 : \dots : Y_K - Y_{K-1}]$ . Construct the matrix  $\mathbf{Z} = [\mathbf{Y} : \gamma\Phi(\mathbf{Y})]$ .
- 2: Perform the SVD of  $\mathbf{Z}$ :  $\mathbf{Z} = \sum_{i=1}^r \sqrt{\lambda_i} U_i V_i^T$ .
- 3: Construct the following decomposition of  $\mathbf{Y} = \mathbf{X}_I$  into the sum of elementary matrices:  $\mathbf{Y} = \sum_{i=1}^r U_i U_i^T \mathbf{Y}$ .
- 4: Partition the set  $\{1, \dots, r\} = \bigsqcup_{m=1}^l J_m$  and perform grouping to obtain a refined matrix decomposition  $\mathbf{Y} = \mathbf{Y}_{J_1} + \dots + \mathbf{Y}_{J_l}$ .
- 5: Obtain a refined series decomposition  $\mathbf{Y} = \tilde{\mathbf{Y}}^{(1)} + \dots + \tilde{\mathbf{Y}}^{(l)}$ , where  $\tilde{\mathbf{Y}}^{(m)} = \mathcal{T}^{-1}\mathcal{H}\mathbf{Y}_{J_m}$ .

Note that steps 2 and 3 of the algorithm are correct, since the column space of  $\mathbf{Z}$  coincides with the column space of  $\mathbf{Y}$ . Therefore,  $\text{rank } \mathbf{Z} = r$  and  $\{U_i\}_{i=1}^r$  is the orthonormal basis of the column space of  $\mathbf{Y}$ .

The following proposition shows that Algorithm 2.7 is exactly Algorithm 2.4 with a specific pair of matrices  $(\mathbf{L}, \mathbf{R})$ , where  $P_i = U_i$ ,  $Q_i$  are normalized vectors  $\mathbf{Y}^T U_i$  in (2.11).

**Proposition 10.** *The left singular vectors of the ordinary SVD of  $\mathbf{Z}$  coincide with the left singular vectors of the  $(\mathbf{L}, \mathbf{R})$ -SVD of the input matrix  $\mathbf{Y}$ , where  $\mathbf{L} \in \mathcal{M}_{L,L}$  is the identity matrix and  $\mathbf{R}$  is*

defined by the equality  $\mathbf{R} = \mathbf{I}_K + \gamma^2 \mathbf{F}^T \mathbf{F}$ ,  $\mathbf{I}_K \in \mathcal{M}_{K,K}$  is the identity matrix and

$$\mathbf{F} = \begin{pmatrix} -1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -1 & 1 & 0 \\ 0 & \cdots & 0 & 0 & -1 & 1 \end{pmatrix} \in \mathcal{M}_{K-1,K}.$$

*Proof.* Note that the standard inner product in the row space of  $\mathbf{Z}$  can be expressed as  $(\mathbf{Z}_1, \mathbf{Z}_2)_{2K-1} = (\mathbf{Q}_1, \mathbf{Q}_2)_K + \gamma^2 (\Phi(\mathbf{Q}_1), \Phi(\mathbf{Q}_2))_{K-1}$ , where  $\mathbf{Q}_1$  and  $\mathbf{Q}_2$  consist of the first  $K$  components of  $\mathbf{Z}_1$  and  $\mathbf{Z}_2$ ,  $\Phi(Q) \in \mathbb{R}^{K-1}$  applied to a vector  $Q = (q_1, \dots, q_K)^T$  consists of successive differences of vector components  $q_{i+1} - q_i$ . Thus, if we introduce the inner product  $\langle \mathbf{Q}_1, \mathbf{Q}_2 \rangle_{\mathbf{R}} = (\mathbf{R}\mathbf{Q}_1, \mathbf{Q}_2)_K$ , then the ordinary SVD of  $\mathbf{Z}$  can be reduced to the  $(\mathbf{L}, \mathbf{R})$ -SVD of  $\mathbf{Y}$  with the corresponding matrices  $\mathbf{L}$  and  $\mathbf{R}$ .  $\square$

**Remark 14.** If  $\mathbf{Y}$  is the trajectory matrix of a series  $\mathbf{Y}_N$ , then the nested SSA with derivatives is equivalent to the MSSA implementation described in Section 2.3.1. Indeed, the trajectory matrix of the derivative time series  $\Phi_{N-1}(\mathbf{Y})$  coincides with the matrix  $\Phi(\mathbf{Y})$ . Although, if  $\mathbf{Y}$  is not Hankel, there is no MSSA analogue.

## Separability

Let  $\mathbf{X}_N = \mathbf{X}_N^{(1)} + \mathbf{X}_N^{(2)}$ , where  $\mathbf{X}_N^{(1)}$  and  $\mathbf{X}_N^{(2)}$  are of finite rank and approximately weakly separable which implies that their row and column trajectory spaces are approximately orthogonal. The same is then true for  $\Phi_{N-1}(\mathbf{X}^{(1)})$  and  $\Phi_{N-1}(\mathbf{X}^{(2)})$ , due to the fact that their column trajectory spaces belongs to the column trajectory spaces of  $\mathbf{X}_N^{(1)}$  and  $\mathbf{X}_N^{(2)}$ , while their row trajectory spaces are spanned by the vectors of the same structure that the vectors constituting bases of the row spaces of  $\mathbf{X}_N^{(1)}$  and  $\mathbf{X}_N^{(2)}$ , except that these basis vectors have length  $K-1$  and not  $K$ . Therefore, approximate orthogonality still holds. Since  $\Phi_{N-1}(\mathbf{X}) = \Phi_{N-1}(\mathbf{X}^{(1)}) + \Phi_{N-1}(\mathbf{X}^{(2)})$ , DerivSSA applied to  $(\mathbf{X}_N, \gamma \Phi_{N-1}(\mathbf{X}))$  will approximately separate the time series  $\mathbf{X}_N^{(1)}$  and  $\mathbf{X}_N^{(2)}$ .

Thus, DerivSSA does not worsen weak separability and can achieve strong separability. It is important to always keep in mind that DerivSSA increases the contribution of high-frequency components and decreases that for low-frequency components.

### 2.3.2. Filter-adjusted O-SSA

Note that sequential differences, which are taken for each row of the matrix  $\mathbf{Y}$ , can be extended to an arbitrary linear filter of the rows. That is, we can chose coefficients of a linear filter  $A = (a_1, \dots, a_t)^T$  and define  $\Phi(\mathbf{Y}) = [Y_1^*, \dots, Y_{K-t+1}^*]$ , where  $Y_i^* = a_1 Y_i + \dots + a_t Y_{i+t-1}$ . The rest of the respective version of SSA is the same as DerivSSA. DerivSSA corresponds to  $t = 2$  and  $A = (-1, 1)^T$ .

## Algorithm

The general algorithm of Filter-adjusted O-SSA is described in two equivalent forms in Algorithms 2.8 and 2.9. Algorithm 2.8 directly follows the description of the method given in Section 2.3.1, while Algorithm 2.9 is more appropriate for an effective implementation and for a modification implemented in Algorithm 2.10.

### ALGORITHM 2.8: Filter-adjusted O-SSA: decomposition

*Input:* Decomposition of the  $L$ -trajectory matrix  $\mathbf{X} = \sum \sigma_i P_i Q_i^T$ , group of components  $I$ ,  $|I| = r$ , filter coefficients  $(a_1, \dots, a_t)$ , weight  $\gamma > 0$ .

*Output:* Decomposition of  $\mathbf{Y} = \mathbf{X}_I$  on elementary matrices  $\mathbf{Y} = \mathbf{Y}_1 + \dots + \mathbf{Y}_r$ , where  $\mathbf{Y}_i = \sigma'_i P'_i (Q'_i)^T$ .

- 1: Form the matrix  $\mathbf{Y} = \mathbf{X}_I = \sum_{i \in I} \sigma_i P_i Q_i^T$ .
- 2: Denote  $\Phi(\mathbf{Y}) = [Y_1^*, \dots, Y_{K-t+1}^*]$ , where  $Y_i^* = a_1 Y_i + \dots + a_t Y_{i+t-1}$ . Construct the matrix  $\mathbf{Z} = [\mathbf{Y} : \gamma \Phi(\mathbf{Y})]$ .
- 3: Compute the SVD of  $\mathbf{Z}$ :  $\mathbf{Z} = \sum_{i=1}^r \sqrt{\lambda_i} U_i V_i^T$ .
- 4: Construct the following decomposition of  $\mathbf{Y} = \mathbf{X}_I$  into a sum of elementary matrices:  $\mathbf{Y} = \sum_{i=1}^r U_i U_i^T \mathbf{Y}$ .
- 5: Obtain the decomposition  $\mathbf{Y} = \sum_{i=1}^r \sigma'_i P'_i (Q'_i)^T$ , where  $\sigma'_i = \|U_i^T \mathbf{Y}\|$ ,  $P'_i = U_i$ ,  $Q'_i = U_i^T \mathbf{Y} / \sigma'_i$ .

### ALGORITHM 2.9: Filter-adjusted O-SSA: decomposition (equivalent)

*Input:* Decomposition of the trajectory matrix  $\mathbf{X} = \sum \sigma_i P_i Q_i^T$ , group of components  $I$ ,  $|I| = r$ , filter coefficients  $(a_1, \dots, a_t)$ , weight  $\gamma$ .

*Output:* Decomposition of  $\mathbf{Y} = \mathbf{X}_I$  on elementary matrices  $\mathbf{Y} = \mathbf{Y}_1 + \dots + \mathbf{Y}_r$ , where  $\mathbf{Y}_i = \sigma'_i P'_i (Q'_i)^T$ .

- 1: Form the matrix  $\mathbf{Y} = \mathbf{X}_I = \sum_{i \in I} \sigma_i P_i Q_i^T$  and compute its thin SVD  $\mathbf{Y} = \mathbf{U}_r \Lambda_r^{1/2} \mathbf{V}_r^T$ . Set  $\mathbf{S} = [S_1 : \dots : S_K] = \Lambda_r^{1/2} \mathbf{V}_r^T \in \mathbb{R}^{r \times K}$ .
- 2: Denote  $\Phi(\mathbf{S}) = [S_1^*, \dots, S_{K-t+1}^*]$ , where  $S_i^* = a_1 S_i + \dots + a_t S_{i+t-1}$ . Construct the matrix  $\mathbf{Z} = [\mathbf{S} : \gamma \Phi(\mathbf{S})] \in \mathbb{R}^{r \times (2K-t+1)}$ .
- 3: Calculate the rotation matrix  $\tilde{\mathbf{U}} \in \mathbb{R}^{r \times r}$  consisting of the eigenvectors of  $\mathbf{Z} \mathbf{Z}^T$  as columns.
- 4: Set  $\tilde{\mathbf{P}} = [\tilde{P}_1 : \dots : \tilde{P}_r] = \mathbf{U}_r \tilde{\mathbf{U}}$  and  $\tilde{\mathbf{Q}} = [\tilde{Q}_1 : \dots : \tilde{Q}_r] = \mathbf{S}^T \tilde{\mathbf{U}}$ .
- 5: Obtain the decomposition  $\mathbf{Y} = \sum_{i=1}^r \sigma'_i P'_i (Q'_i)^T$ , where  $\sigma'_i = \|\tilde{Q}_i\|$ ,  $P'_i = \tilde{P}_i$ ,  $Q'_i = \tilde{Q}_i / \sigma'_i$ .

The method introduced in Section 2.3.1 has a modification implemented in RSSA, which can slightly worsen the separability but has an advantage that it orders the eigentriples corresponding to sine-waves exactly by the decrease of their frequencies, independently of the values of the sine-wave amplitudes. We will call this modification ‘Filter-adjusted O-SSA with normalization’.

The main difference between Algorithms 2.9 and 2.10 is in the construction of the matrix  $\mathbf{S}$  at step 1.

---

**ALGORITHM 2.10:** Filter-adjusted O-SSA with normalization: decomposition

---

*Input:* Decomposition of the trajectory matrix  $\mathbf{X} = \sum \sigma_i P_i Q_i^T$ , group of components  $I$ ,  $|I| = r$ , filter coefficients  $(a_1, \dots, a_t)$ , weight  $\gamma$ .

*Output:* Decomposition of  $\mathbf{Y} = \mathbf{X}_I$  on elementary matrices  $\mathbf{Y} = \mathbf{Y}_1 + \dots + \mathbf{Y}_r$ , where  $\mathbf{Y}_i = \sigma'_i P'_i (Q'_i)^T$ .

- 1: Form the matrix  $\mathbf{Y} = \mathbf{X}_I = \sum_{i \in I} \sigma_i P_i Q_i^T$  and construct its thin SVD  $\mathbf{Y} = \mathbf{U}_r \Lambda_r^{1/2} \mathbf{V}_r^T$ . Set  $\mathbf{S} = [S_1 : \dots : S_K] = \mathbf{V}_r^T \in \mathbb{R}^{r \times K}$ .
  - 2: Denote  $\Phi(\mathbf{S}) = [S_1^*, \dots, S_{K-t+1}^*]$ , where  $S_i^* = a_1 S_i + \dots + a_t S_{i+t-1}$ . Construct the matrix  $\mathbf{Z} = [\mathbf{S} : \gamma \Phi(\mathbf{S})] \in \mathbb{R}^{r \times (2K-t+1)}$ .
  - 3: Calculate the rotation matrix  $\tilde{\mathbf{U}} \in \mathbb{R}^{r \times r}$  consisting of the eigenvectors of  $\mathbf{Z} \mathbf{Z}^T$  as columns.
  - 4: Set  $\tilde{\mathbf{P}} = [\tilde{P}_1 : \dots : \tilde{P}_r] = (\mathbf{U}_r \Lambda_r^{1/2}) \tilde{\mathbf{U}}$  and  $\tilde{\mathbf{Q}} = [\tilde{Q}_1 : \dots : \tilde{Q}_r] = \mathbf{S}^T \tilde{\mathbf{U}}$ .
  - 5: Obtain the decomposition  $\mathbf{Y} = \sum_{i=1}^r \sigma'_i P'_i (Q'_i)^T$ , where  $\sigma'_i = \|\tilde{P}_i\|$ ,  $P'_i = \tilde{P}_i / \sigma'_i$ ,  $Q'_i = \tilde{Q}_i$ .
- 

**Remark 15.** Algorithms 2.8–2.10 can be extended to the case when several filters in a stacked manner are applied. For example, if filters  $\Phi_1$  and  $\Phi_2$  are given, then the matrix  $\mathbf{Z}$  at Step 2 has the forms  $\mathbf{Z} = [\mathbf{Y} : \gamma \Phi_1(\mathbf{Y}) : \gamma \Phi_2(\mathbf{Y})]$  and  $\mathbf{Z} = [\mathbf{S} : \gamma \Phi_1(\mathbf{S}) : \gamma \Phi_2(\mathbf{S})]$  respectively.

The reconstruction algorithm is the same as for most versions of SSA. Since Filter-adjusted O-SSA (as well as DerivSSA) is a nested method, the result is a decomposition of a chosen series component rather than a decomposition of the original series.

---

**ALGORITHM 2.11:** Filter-adjusted O-SSA: reconstruction

---

*Input:* Decomposition  $\mathbf{Y} = \sum_{i=1}^r \mathbf{Y}_i$ , where  $\mathbf{Y}_i = \sigma'_i P'_i (Q'_i)^T$ , grouping  $I = \bigsqcup_{k=1}^l J_k$ .

*Output:* Refined series decomposition  $\mathbf{Y} = \tilde{\mathbf{Y}}^{(1)} + \dots + \tilde{\mathbf{Y}}^{(l)}$ .

- 1: Obtain the grouped matrix decomposition  $\mathbf{Y} = \mathbf{Y}_{J_1} + \dots + \mathbf{Y}_{J_l}$ , where  $\mathbf{Y}_J = \sum_{j \in J} \mathbf{Y}_j$ .
  - 2: Obtain a refined series decomposition  $\mathbf{Y} = \tilde{\mathbf{Y}}^{(1)} + \dots + \tilde{\mathbf{Y}}^{(l)}$ , where  $\tilde{\mathbf{Y}}^{(m)} = \mathcal{T}^{-1} \circ \Pi_{\mathcal{H}}(\mathbf{Y}_{J_m})$ ,  $m = 1, \dots, l$ .
- 

**Remark 16.** Algorithm 2.11 describes a refined decomposition of the matrix  $\mathbf{X}_I$ . However, we can consider Filter-adjusted O-SSA as an algorithm, which we apply to the full decomposition of the trajectory matrix  $\mathbf{X}$  of an original series  $X$ , which changes components from the group  $I$ . Then the result is also a full decomposition of  $\mathbf{X}$ .

### 2.3.3. Examples

#### Separation of sine waves with equal amplitudes

Consider the series  $x_n = \sin(2\pi n/10) + \sin(2\pi n/15)$ ,  $n = 1, \dots, N$ ,  $N = 150$ ,  $L = 70$ , which is depicted in Fig. 2.10.

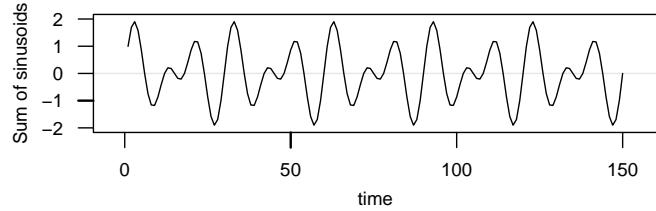


Figure 2.10: Sum of two sinusoids with equal amplitudes.

Sinusoids with periods 10 and 15 are approximately separable for such series and window lengths. However, since the sinusoid amplitudes are equal, there is no strong separability and therefore after Basic SSA we obtain an unsatisfactory decomposition, an arbitrary mixture of the sinusoids (top picture of Fig. 2.11) with  $\mathbf{w}$ -correlation between reconstructed by ET1–2 and ET3–4 series equal to 0.92.

The decomposition performed by DerivSSA with  $\gamma = 10$  applied to the group ET1–4 with  $J_1 = \{1, 2\}$  and  $J_2 = \{3, 4\}$  (Algorithm 2.7) is depicted in the bottom graph of Fig. 2.11 and demonstrates the very accurate separability,  $\mathbf{w}$ -correlation is equal to 0.01. The second measure, the mean proportion  $0.5(\tau_{r_1}(X^{(1)}) + \tau_{r_2}(X^{(2)}))$ , is diminished from 0.3266 to 0.0003. For this example, the obtained decomposition practically does not depend on  $\gamma$  for all  $\gamma > 2$ .

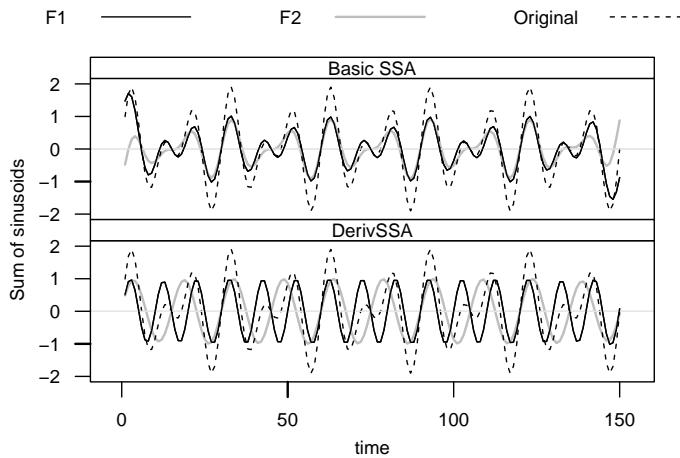


Figure 2.11: Sum of two sinusoids with equal amplitudes: reconstruction by Basic SSA (top) and DerivSSA (bottom).

#### Real-life time series

In this section, we apply Iterative O-SSA (Algorithms 2.5 and 2.6) and DerivSSA (Algorithm 2.7) to real-life time series. The role of the methods for separability of sine-waves was

demonstrated in Sections 2.2.4 and 2.3.3 with the help of simulated data. The obtained conclusions are generally valid for real-life series: DerivSSA adds to Basic SSA the ability to separate sine waves with close amplitudes, while Iterative O-SSA can help in separation of sine waves, which are not orthogonal, that is, their frequencies are insufficiently far one from another. Note that since in real-life series with seasonality there are no close frequencies, DerivSSA can be very useful for seasonality decomposition.

In this section, we consider the problem of trend extraction. The choice of examples is explained by the following considerations.

If a time series is long enough, then the oscillations are weakly separated from the trend and only strong separability is under question. Therefore, we expect that DerivSSA will work for trends of complex forms.

For short series, the trend can be not orthogonal to a periodic component like seasonality; therefore, DerivSSA can even worsen the separability; moreover, derivation suppresses low-frequency components. On the other hand, Iterative O-SSA is specially designed to separate non-orthogonal series components.

We will take only one iteration in Iterative O-SSA method, since this is sufficient to obtain appropriate decompositions in the considered examples and also makes the methods comparable by computational cost.

**Improving of strong separability** Let us consider US Unemployment data (monthly, 1948–1981, thousands) for male (20 years and over). Data are taken from [119], the series length  $N$  is equal to 408, see Fig. 2.12. Since the series is long, we can expect weak separability of the trend and the seasonality. For better weak separability we choose the window length equal to  $L = N/2 = 204$ , which is divisible by 12.

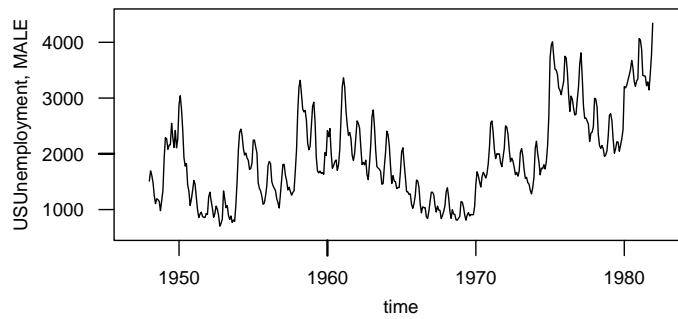


Figure 2.12: US unemployment: initial series.

Basic SSA does not separate the trend and seasonality for this time series (see Fig. 2.13 and Fig. 2.17 (left)), likely due to lack of strong separability. This is the typical situation when the trend has a complex form, trend components are mixed with the seasonality components and therefore the so-called Sequential SSA was recommended [1, Section 1.7.3]. However, this is the case when DerivSSA should help.

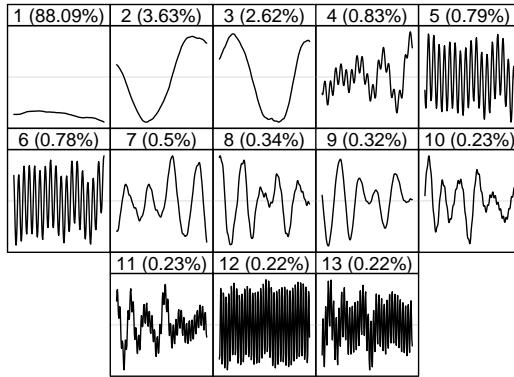


Figure 2.13: US unemployment: eigenvectors obtained by Basic SSA.

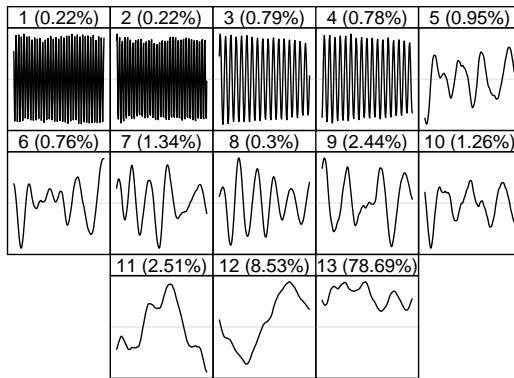


Figure 2.14: US unemployment: eigenvectors obtained by DerivSSA.

We apply DerivSSA to the group ET1–13 that can be related to the signal. DerivSSA separates different frequencies so that components with higher frequencies become leading ones. Since the low-frequency components in the considered series have a large contribution, the weight of derivatives should be large to make the seasonal components leading; we take  $\gamma = 1000$ .

The resulting eigenvectors are depicted in Fig. 2.14. One can see that the first 4 components contain seasonality, while the eigenvectors 5–13 contains components of the trend. The mixture of the components within the trend group is not important. Fig. 2.14 demonstrates that the seasonal components are now separated from the residual. Fig. 2.15 depicting the DerivSSA reconstructions of the trend and the seasonality confirms that DerivSSA visibly improves the reconstruction accuracy, especially at the ends of the series.

Since Iterative O-SSA has a possibility of sigma-correction, it also can help to move apart the decomposition components, and therefore we can apply Iterative O-SSA to the group ET1–13 with the refined groups ET1–4,7–11 (trend) and ET5,6,12,13 (seasonality). Since the components of the Basic SSA decomposition are mixed, we refer the components that contain mostly trend and slow cycles to the first group and the components that contain mostly seasonality to the second group. As eigenvectors reflect forms of the corresponding time series components, we can use the graph of eigenvectors shown in Fig. 2.13 for the initial grouping. For example, the fourth eigenvector looks like slow oscillations corrupted by seasonality and therefore we refer it to the trend group, while the fifth eigenvector looks like seasonal component corrupted by something slow varying and

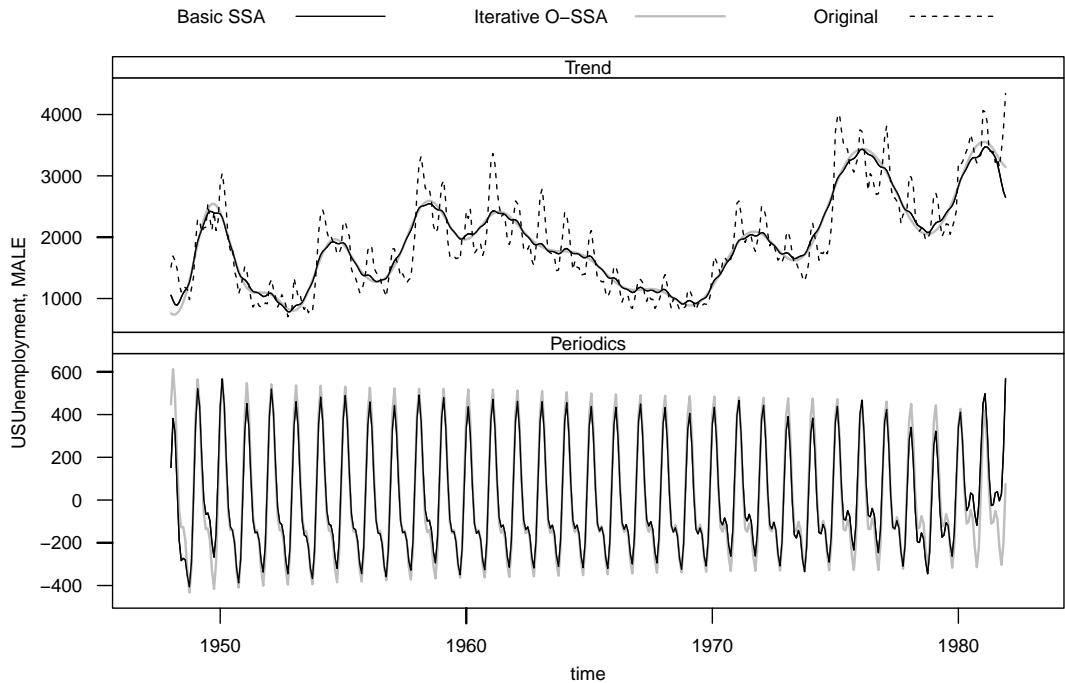


Figure 2.15: US unemployment: Decompositions by Basic SSA and Iterative O-SSA, which coincides with that by DerivSSA.

we refer it to the seasonality group. We apply one iteration with sigma-correction, taking  $\varkappa = 2$ . After reordering caused by the sigma-correction, the first trend group consists of the first eight components 1–8, while the second seasonality group consists of 9–13 components, see Fig. 2.16.

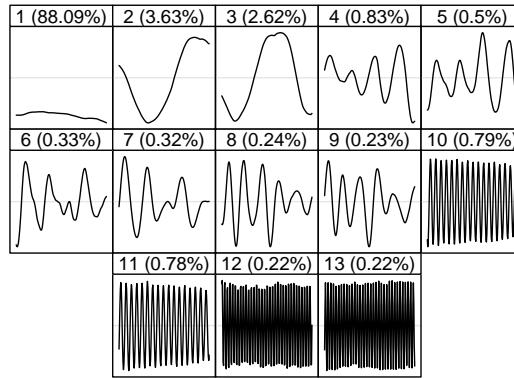


Figure 2.16: US unemployment: Iterative O-SSA eigenvectors.

The trend eigenvectors of the DerivSSA decomposition (Fig. 2.14, ET5–13) differ from that of the Iterative O-SSA decomposition (Fig. 2.16, ET1–8), the seasonality components are almost the same. Nevertheless, the result of Iterative O-SSA reconstruction is visibly the same as that of DerivSSA shown in Fig. 2.15 and therefore we do not depict this reconstruction.

Fig. 2.17 contains the  $\mathbf{w}$ -correlations between the elementary components provided by Basic SSA (left) and the  $\mathbf{w}$ -correlations between the elementary components reconstructed by Iterative O-SSA (right). The figure confirms the improving of separability. Note that although an oblique decomposition was formally obtained, this decomposition is almost F-orthogonal (the maximal F-



Figure 2.17: US unemployment:  $\mathbf{w}$ -correlations before (left) and after (right) Iterative O-SSA.



Figure 2.18: US unemployment: 2D plots of periodic eigenvectors before (left) and after (right) Iterative O-SSA.

correlation between elementary matrix components, which is calculated as  $\langle \mathbf{X}_i, \mathbf{X}_j \rangle_F / (\|\mathbf{X}_i\|_F \|\mathbf{X}_j\|_F)$ , is equal to 0.00368); therefore, conventional  $\mathbf{w}$ -correlations are appropriate, see Section 2.2.1. For trend extraction, it is important that correlations between trend and seasonality groups are close to zero. Really, correlations between ET1–8 and ET9–13 are small. The mixture of the components within the trend group is not important. One can see that the trend components are still slightly mixed with the noise components. This can be explained by a mixture with the residual before iterations (left); therefore, it cannot be corrected by Iterative O-SSA (right), since the nested version is used. Fig. 2.18 shows the improvement of separability with the help of scatterplots of seasonal eigenvectors. After one iteration, plots of seasonal eigenvectors form almost regular polygons.

Figures for the decomposition of DerivSSA analogous to Fig. 2.17 and 2.18 are very similar and we do not present them in this work. Note that in DerivSSA we group components after their separation, what is easier than to group mixing components for Iterative O-SSA before separation. That is, in the considered example the resultant decomposition is the same, but application of DerivSSA is much easier.

**Improving of weak separability** Let us consider the series ‘Fortified wine’ (fortified wine sales, Australia, monthly, from January 1980, thousands of litres) taken from [120]. The first 120 points of the series are depicted in Fig. 2.19.

The series length is long enough to obtain weak separability; therefore, we will consider short subseries to demonstrate the advantage of Iterative O-SSA for improving of weak separability.

We take here the window length  $L = 18$  to make the difference between methods clearly visible on the figures, although the relation between accuracies of the considered methods is the same for other choices of window lengths. Let us consider two subseries, from 30th to 78th points and from 36th to 84th points. The difference consists in behavior of the seasonality at the ends of the subseries.

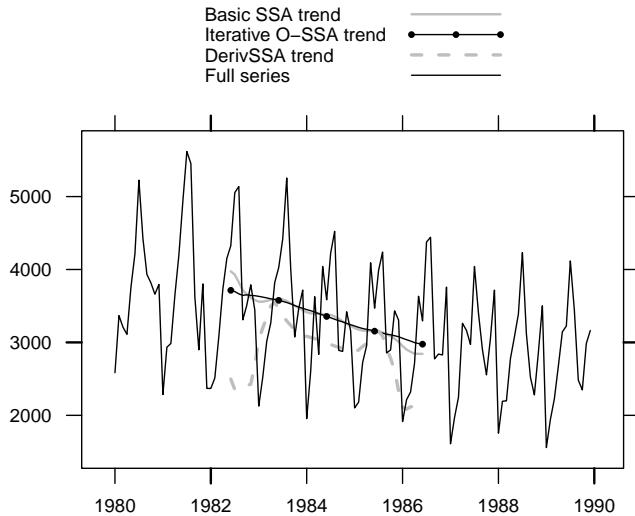


Figure 2.19: Fortified wines: trend reconstruction by DerivSSA and Iterative O-SSA for subseries of points 30–78.

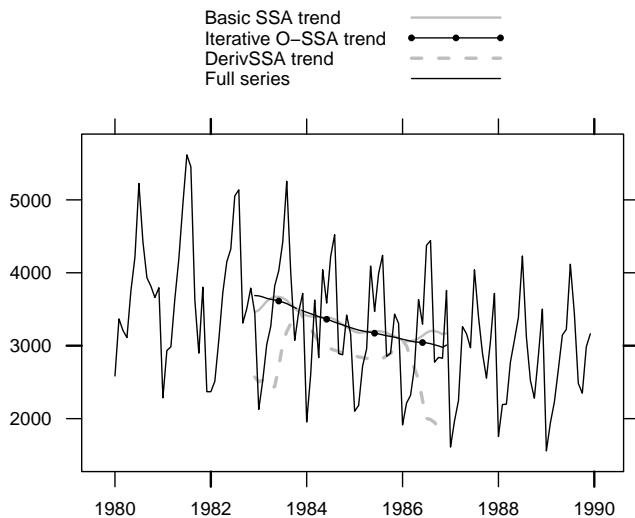


Figure 2.20: Fortified wines: trend reconstruction by DerivSSA and Iterative O-SSA for subseries of points 36–84.

As well as in the previous example, we start with Basic SSA. ET1 is identified as corresponding to trend, other components are produced by seasonality and noise (we do not include their pictures). One can see in Fig. 2.19 and 2.20 (thick gray line) that the reconstructed trend is slightly mixed with the seasonality and steps after the seasonality at the ends of the series.

To apply Iterative O-SSA, we should choose a group of elementary components containing the trend components and approximately separated from the residual. Let it be ET1–7. Thus, we apply one iteration of O-SSA to the refined groups ET1 and ET2–7. Since the trend has the contribution much larger than the residual, we consider Iterative O-SSA with no sigma-correction. The result of reconstruction is much more relevant, see Fig. 2.19 and 2.20 (solid line with filled circles). The gray dashed line in the same figures shows that DerivSSA gives more poor reconstruction than Basic SSA in this example.

## 2.4. SSA–ICA

This section is devoted to one more nested method of improving separability.

Independent Component Analysis (ICA) [121, 122] is a statistical approach, which is similar to PCA but provides a decomposition into independent random variables instead of uncorrelated components. The aim of the ICA is finding statistically independent components  $\{\eta_i; i = 1, \dots, p\}$  from observations of their linear combinations  $\{\xi_i; i = 1, \dots, p\}$ . Certainly, for normally distributed random variables it is the same.

In this section, we consider deterministic optimization problems that are solved within the sample versions of ICA to construct a decomposition into rank-one matrices; the trajectory matrix is considered as a sample.

The ICA method does not have approximation properties, in particular, it is not intended to separate the signal from the noise. Therefore, the SSA-ICA method is used as a nested method that refines the decomposition obtained with Basic SSA: we first obtain an estimate of the trajectory matrix of the signal of interest using Basic SSA, and then we decompose the obtained matrix using SSA-ICA.

Since it is difficult to measure statistical dependence, there are different approaches to ICA. We consider two of them. The content of Sections 2.4.1 and 2.4.2 can be found in [29, Section 1.5.4] and [14] respectively.

### 2.4.1. Maximization of entropy

Let us use the idea from the projection pursuit method of multivariate analysis (see [123] for a review). For choosing directions, the projection pursuit uses a criterion based on the form of the distribution of the projection in a given direction. Assuming  $L \leq K$ , we apply the projection pursuit to the trajectory matrix with its rows considered as variables.

Let us start by considering projections in different directions for two vectors taken from subspaces corresponding to different time series components. For simplicity of depiction, we rotate the data and consider projections on the x-axis. Fig. 2.21c shows projections for different rotations of two sine wave variables. The first picture in a row (the case  $\alpha = 0$ ) corresponds to the proper rotation, and the last one (with  $\alpha = \pi/4$ ) shows the worst possible mixture. We can see that the

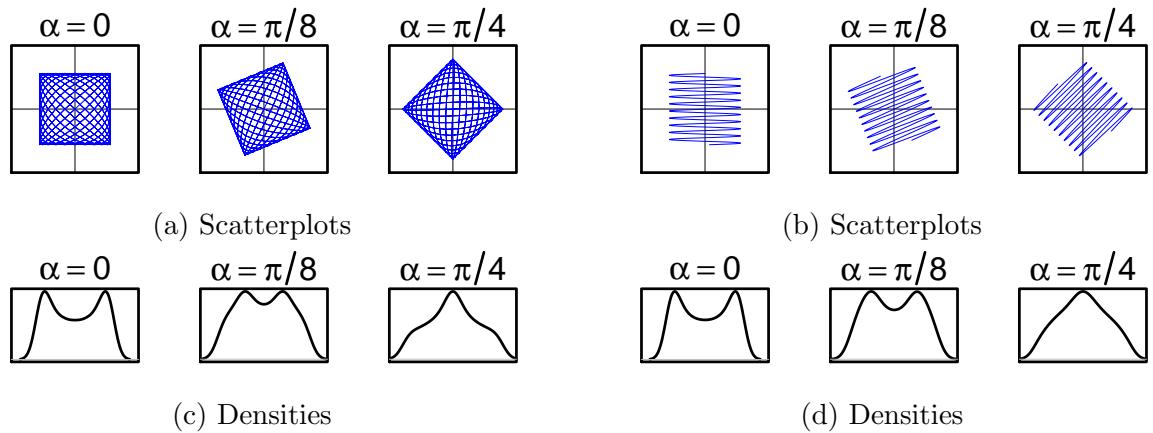


Figure 2.21: Projection: two sine waves (left) and sine wave with linear series (right).

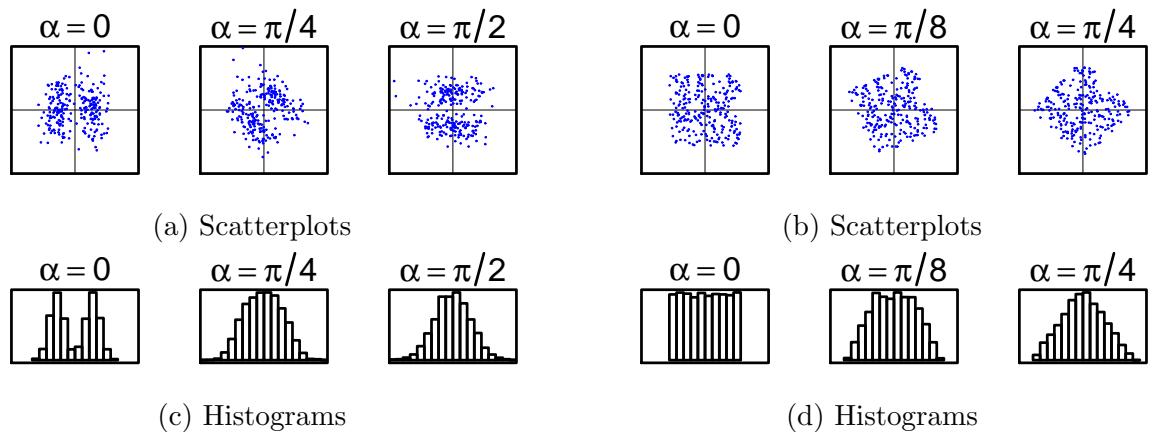


Figure 2.22: Projection pursuit: two clusters (left) and two independent uniformly distributed variables (right).

estimated densities are totally different. To check that this result is generic, let us consider similar pictures for a sine wave and a linear function (Fig. 2.21d). The result is very similar. We thus conclude that the idea of projection pursuit may help in solving the problem of separation.

Let us consider the projection pursuit method for cluster analysis where the proper rotation ( $\alpha = 0$ ) corresponds to the maximal distance from the normal distribution. Fig. 2.22c shows that the distributions of proper projections and improper projections are similar to the ones depicted in Fig 2.21c and 2.21d.

One approach to ICA is the use of the projection pursuit method (Fig. 2.22d confirms it) [121, 122]. Let us describe this approach. Without loss of generality, we can assume that  $\{\xi_i\}$  are pre-whitened.

The mutual information of the random vector  $(\eta_1, \dots, \eta_p)$  can be measured as

$$I(\eta_1, \dots, \eta_p) = \sum_{k=1}^p H(\eta_k) - H(\eta_1, \dots, \eta_p),$$

where  $H(\eta) = - \int f(x) \log_2(f(x)) dx$  is the differential entropy and  $f(x)$  is the density function of  $\eta$ . Therefore, searching for independent components is equivalent to searching for random variables  $\{\eta_i\}$ , which are linear combinations of  $\{\xi_i\}$  and have the minimal value of the mutual information.

It appears that the minimization of the mutual information is equivalent to the maximization of the total negentropy of  $\{\eta_i\}$ , which is the sum of marginal negentropies  $J(\eta_i) = H(v) - H(\eta_i)$ ,  $v \sim N(0, 1)$ . This means that the ICA works similarly to the search for the direction with the distribution of projections that are maximally distant from the normal distribution; that is, to the projection pursuit.

Rather than maximizing negentropies, which requires the estimation of the marginal densities for calculating entropies of  $\eta_i$ 's, we can consider maximization of simple functionals like

$$J(\eta_i) \sim [-EG(\eta_i) + C_v]^2, \quad (2.12)$$

where  $C_v = EG(v)$ ,  $G(u) = e^{-u^2/2}$ ; other choices of  $G$  can be considered as well, see [124]. An implementation of the ICA using optimizing the functional (2.12) can be found in the R-package FASTICA, see [125].

Since we observe realizations of  $p$  random variables  $\mathbf{Y} = [Y_1 : \dots : Y_p]$ ,  $Y_i \in \mathbb{R}^K$ , rather than maximizing (2.12) we should calculate and maximize the following functional of their linear combinations with coefficients  $W \in \mathbb{R}^p$ :

$$J(Z) = \left( -\frac{1}{K} \sum_{i=1}^K e^{\xi_i^2/2} - C_v \right)^2 \rightarrow \max_{Z = \mathbf{Y}W, ||Z||=1} \quad (2.13)$$

In applications to blind signal separation, the cooperation between SSA and ICA has been already considered, see [126]. In this application, Basic SSA is used for the removal of noise and then the ICA is applied for the extraction of independent components from the mixture.

The theory of ICA was developed for random variables and is not applicable in the deterministic case. Therefore, the application of the ICA to deterministic sources can be formally considered as the projection pursuit which searches for the linear combination of the observed variables (factor vectors in SSA) that maximizes some functional like (2.13). Since the concept of statistical independence is not defined for deterministic vectors we will use the names ‘ICA’ and ‘independent vectors’ purely formally and may use quotes while referring to them. It has been established by computer simulations and confirmed by theoretical results that in the examples considered in Fig 2.22 and some similar ones, the ‘ICA’ does indeed succeed in separating the time series components, even if the SVD does not provide strong separability.

The ‘ICA’ has the following important drawback: it does not make the ordering of the found components (vectors) as the SVD does. In particular, two vectors corresponding to a sine wave can have arbitrary numbers in the decomposition by the ICA and therefore searching for them is a more difficult task than applying the SVD. Also, the accuracy of weak separability which the ICA provides is worse than that for the SVD case. Moreover, the stability of numerical the ICA procedures is worse than for the SVD case. Therefore, in SSA, the ICA is worthwhile to consider only as a supplement to the SVD for finding proper rotations in the presence of weak separability but lack of strong separability. By no means the ICA can be recommended as a full replacement for the SVD.

Below we suggest a scheme for building a refined grouping by the SSA–ICA procedure. This scheme could be used as a substitution for the grouping step in Basic SSA. Let us have the expansion  $\mathbf{X} = \sum_{m=1}^d \sqrt{\lambda_j} U_j V_j^T$  at the SVD step.

### Refined grouping by SSA–ICA

1. Make a grouping  $\mathbf{X} = \mathbf{X}_{I_1} + \dots + \mathbf{X}_{I_m}$  as in Basic SSA; this corresponds to strongly separated time series components.
2. Choose a group  $I$  consisting of  $p$  indices, which is possibly composed of several interpretable components that are mixed.
3. Extract  $p$  ‘independent’ vectors  $Q_i$  applying the ‘ICA’ to  $\mathbf{X}_I$ . Then  $\mathbf{X}_I = \sum_{i=1}^p P_i Q_i^T$ , where  $P_i = \mathbf{X}_I Q_i$ .
4. Make  $k$  subgroups from the group  $I$  by splitting  $\mathbf{X}_I = \mathbf{X}_{I,1} + \dots + \mathbf{X}_{I,k}$ .

**Example 1.** Consider the example ‘Fortified wines’ depicted in Fig. 2.19. For the analysis, we take the first 120 points. The window length  $L$  does not provide strong separability for ET8–11 (sine waves with periods 2.4 and 3), see Fig. 2.23a, which depicts the  $\mathbf{w}$ -correlation matrix, where the block of four correlated components is clearly seen. 2D-scatterplots of factor vectors are depicted in Fig. 2.23c and demonstrate the absence of structure. Let us apply ‘ICA’ to the trajectory matrix reconstructed by the eigentriples 8–11. Fig. 2.23b and 2.23d show that the ‘ICA’ makes a

successful separation of the two sine waves. Let us remark that the resultant components of the ‘ICA’ needed an additional ordering so that the two sine waves with the same frequency obtain consecutive indices.

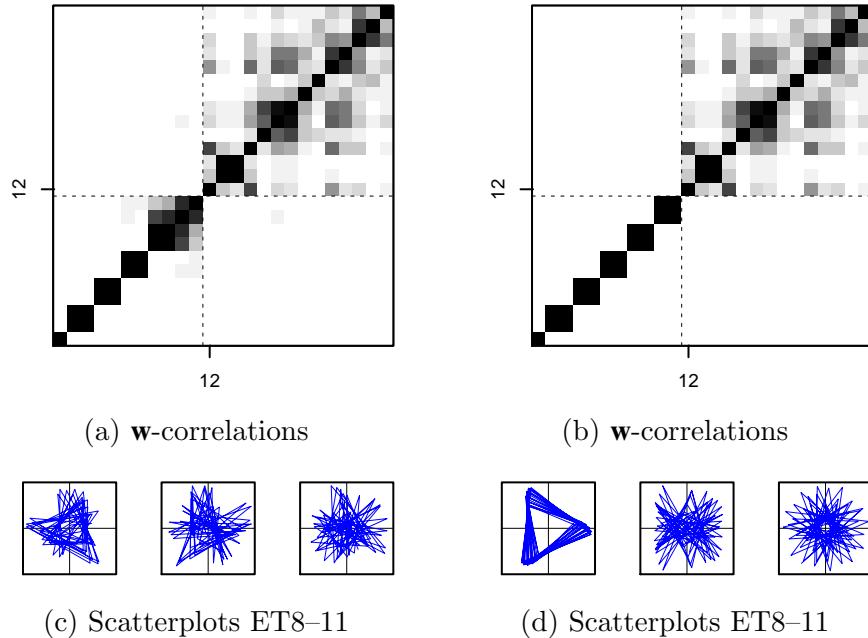


Figure 2.23: ‘Fortified wines’: SVD (left) and ICA for separability ET8–11 (right).

#### 2.4.2. SOBI-AMUSE

The method considered in this section uses the AMUSE method, which is another version of ICA that was proposed for random signal separation [127, 128, 122] and it is more suitable for time series analysis.

Suppose we observe a series  $\mathbf{X} = \mathbf{X}_N = (x_1, \dots, x_N)$  of length  $N$ , and  $\mathbf{X}_N = \mathbf{X}_N^{(1)} + \mathbf{X}_N^{(2)}$ .

Here is an algorithm that solves the problem of the lack of strong separability of weakly separable series. The weakly separable series remain weakly separable under slightly changed conditions on the series length and the window length.

In the SSA-AMUSE algorithm, in addition to the window length parameter  $L$ , there is a shift parameter  $\tau \geq 1$ . We will describe the algorithm with an arbitrary  $\tau$ . However, a value of  $\tau = 1$  is sufficient to obtain strong separability.

Let us give the auxiliary statements necessary to justify the proposed algorithm. Let us introduce a notation that is key to the algorithm and links it to the AMUSE method of independent component analysis.

Let  $\mathbf{B}$  be an arbitrary matrix of  $\mathcal{M}_{A,B}$  and  $\tau$  be some positive integer. Denote by  $\overline{\mathbf{B}}^{(\tau)}$  the matrix  $\mathbf{B}$  without the first  $\tau$  rows,  $\mathbf{B}_{(\tau)}$  the matrix  $\mathbf{B}$  without the last  $\tau$  rows,  ${}_{(\tau)}|\mathbf{B}$  matrix  $\mathbf{B}$  without the first  $\tau$  columns,  $\mathbf{B}|_{(\tau)}$  matrix  $\mathbf{B}$  without the last  $\tau$  columns when  $0 < \tau < A$ ,  $0 < \tau < B$ .

Denote

$$\mathbf{C}_\tau(\mathbf{B}) = \frac{1}{2} \left[ \left( \bar{\mathbf{B}}^{(\tau)} \right)^T \underline{\mathbf{B}}_{(\tau)} + \left( \underline{\mathbf{B}}_{(\tau)} \right)^T \bar{\mathbf{B}}^{(\tau)} \right]. \quad (2.14)$$

### Supporting statements

**Lemma 5.** Let  $\mathbf{Y} \in \mathcal{M}_{L,K}$ ,  $\tau < K - r$ . Then for any SVD of the matrix  $[(\tau) | \mathbf{Y} : \mathbf{Y}|_{(\tau)}] = \mathbf{U}\Lambda^{1/2}\mathbf{T}^T$ , where  $\Lambda \in \mathcal{M}_{r,r}$  is a diagonal matrix with singular numbers on the diagonal,  $\mathbf{U} \in \mathcal{M}_{L,r}$  is the matrix of left singular vectors, and  $\mathbf{T} \in \mathcal{M}_{2K-2\tau,r}$  is the matrix of right singular vectors, the matrix  $\mathbf{T}$  has the form  $\mathbf{T} = [\bar{\mathbf{Q}}^{(\tau)} : \underline{\mathbf{Q}}_{(\tau)}]$ , where  $\mathbf{Q} = \mathbf{Y}^T \mathbf{U} \Lambda^{-1/2}$ .

*Proof.* The SVD of the matrix  $[(\tau) | \mathbf{Y} : \mathbf{Y}|_{(\tau)}]$  can be written as

$$[(\tau) | \mathbf{Y} : \mathbf{Y}|_{(\tau)}] = \left[ \mathbf{U} \Lambda^{1/2} \left( \mathbf{T}^{(1)} \right)^T : \mathbf{U} \Lambda^{1/2} \left( \mathbf{T}^{(2)} \right)^T \right],$$

where  $\mathbf{T}^{(1)} = ((\tau) | \mathbf{Y})^T \mathbf{U} \Lambda^{-1/2}$  and  $\mathbf{T}^{(2)} = (\mathbf{Y}|_{(\tau)})^T \mathbf{U} \Lambda^{-1/2}$ . Then  $\mathbf{Q} = \mathbf{Y}^T \mathbf{U} \Lambda^{-1/2}$ . The equalities  $\bar{\mathbf{Q}}^{(\tau)} = \mathbf{T}^{(1)}$  and  $\underline{\mathbf{Q}}_{(\tau)} = \mathbf{T}^{(2)}$  follow from matrix multiplication rules.  $\square$

**Remark 17.** Since it is easy to show that the columns of matrix  $\mathbf{U}$  form an orthonormal basis of the column space of the matrix  $\mathbf{Y}$ , we get  $\mathbf{Y} = \mathbf{U} \mathbf{U}^T \mathbf{Y}$ .

**Corollary 4.**  $\mathbf{Y} = \mathbf{U} \Lambda^{1/2} \mathbf{Q}^T$ .

*Proof.* It follows from  $\mathbf{Q} = \mathbf{Y}^T \mathbf{U} \Lambda^{-1/2}$  and Remark 17.  $\square$

Note that the columns of the  $\mathbf{Q}$  matrix are not orthogonal and normalized.

**Lemma 6.** Let the matrix  $\mathbf{Q} \in \mathcal{M}_{K,r}$  and  $\mathbf{C}_\tau(\mathbf{Q}) = \mathbf{W} \mathbf{D} \mathbf{W}^T$  is the eigendecomposition. Then for  $\mathbf{S} = \mathbf{Q} \mathbf{W}$  the matrix  $\mathbf{C}_\tau(\mathbf{S}) = \mathbf{D}$  is diagonal.

*Proof.* The lemma is derived by substituting  $\mathbf{S} = \mathbf{Q} \mathbf{W}$  into  $\mathbf{C}_\tau(\mathbf{S})$  and multiplying the matrices.  $\square$

The following theorem approves the decomposition obtained by the SSA-AMUSE method.

**Theorem 3.** Let  $\mathbf{Y} \in \mathcal{M}_{L,K}$  be a matrix of rank  $r$ . Then  $\mathbf{Y} = \sum_{i=1}^r \hat{U}_i S_i^T$ , where  $\hat{U}_i$  are columns of the matrix  $\hat{\mathbf{U}} = \mathbf{U} \Lambda^{1/2} \mathbf{W}$ ,  $\mathbf{W}$  from the spectral decomposition  $\mathbf{C}_\tau(\mathbf{Q}) = \mathbf{W} \mathbf{D} \mathbf{W}^T$ ,  $\mathbf{Q} = \mathbf{Y}^T \mathbf{U} \Lambda^{-1/2}$ ,  $\mathbf{U}$  are left singular vectors  $[(\tau) | \mathbf{Y} \mathbf{Y}|_{(\tau)}]$ .

*Proof.* By Corollary 4 and Lemma 6,  $\mathbf{Y} = \mathbf{U} \Lambda^{1/2} \mathbf{W} \mathbf{S}^T = \hat{\mathbf{U}} \mathbf{S}^T = \sum_{i=1}^r \hat{U}_i S_i^T$  where  $\hat{U}_i$  are the columns of the matrix  $\hat{\mathbf{U}} = \mathbf{U} \Lambda^{1/2} \mathbf{W}$ .  $\square$

## Algorithm

As mentioned above, the SSA-AMUSE algorithm constructs a decomposition of some grouped matrix  $\mathbf{Y} = \mathbf{X}_I$ , which was obtained at the Decomposition step of the Basic SSA method. The algorithm is arranged as follows.

---

### ALGORITHM 2.12: SSA-AMUSE algorithm

---

*Input:* Matrix  $\mathbf{Y}$  of rank  $r$ , shift  $\tau > 0$ .

*Output:* Decomposition of  $\mathbf{Y} = \mathbf{X}_I$  on elementary matrices  $\mathbf{Y} = \mathbf{Y}_1 + \dots + \mathbf{Y}_r$ , where  $\mathbf{Y}_i = \hat{\mathbf{U}}_i S_i^T$ .

- 1: Calculate the SVD of the matrix  $[(\tau) | \mathbf{Y} : \mathbf{Y}|_{(\tau)}]$ , which by Lemma 5 has the form  $[(\tau) | \mathbf{Y} : \mathbf{Y}|_{(\tau)}] = \mathbf{U} \boldsymbol{\Lambda}^{1/2} \left[ (\bar{\mathbf{Q}}^{(\tau)})^T : \underline{\mathbf{Q}}_{(\tau)}^T \right]$ , where  $\mathbf{Q} = \mathbf{Y}^T \mathbf{U} \boldsymbol{\Lambda}^{-1/2}$ .
  - 2: Calculate  $\mathbf{W} = [W_1 : \dots : W_r]$ , where  $W_i$ ,  $i = 1, \dots, r$ , are eigenvectors of the matrix  $\mathbf{C}_\tau(\mathbf{Q})$ .
  - 3: Calculate  $\mathbf{S} = [S_1 : \dots : S_r] = \mathbf{Q} \mathbf{W}$ .
  - 4: Obtain  $\hat{\mathbf{U}} = [\hat{\mathbf{U}}_1 : \dots : \hat{\mathbf{U}}_r] = \mathbf{U} \boldsymbol{\Lambda}^{1/2} \mathbf{W}$ . By Theorem 3,  $\mathbf{Y} = \hat{\mathbf{U}} \mathbf{S}^T$ . So we get a decomposition into elementary matrices  $\mathbf{Y} = \sum_{i=1}^r \hat{\mathbf{U}}_i S_i^T$ .
- 

## Separability

For convenience, let us assume that the trajectory matrix  $\mathbf{X}$  of the series  $X_N$  has rank  $d$ . As the index groups  $I$  in the algorithms, we will take the numbers of nonzero singular numbers of the SVD of the trajectory matrix. Accordingly, the grouped matrix  $\mathbf{Y} = \mathbf{X}_I = \mathbf{X}$  and has rank  $r = d$ . In the general case, the group  $I$  can correspond to a component of the series, strongly separable from the remainder.

**Auxiliary statements** Let us consider time series of finite rank governed by LRRs and prove several propositions about the properties of their trajectory spaces.

Denote  $\mathfrak{X}^{(K,1)}$ ,  $\mathfrak{X}^{(K,2)}$  and  $\mathfrak{X}^{(K)}$  the row spaces of the trajectory matrices of time series  $X_N^{(1)}$ ,  $X_N^{(2)}$  and  $X_N$  of dimensions  $d_1$ ,  $d_2$  and  $d$  respectively. Let  $\mathfrak{X}^{(K,1)}|_{(\tau)}$ ,  $\mathfrak{X}^{(K,2)}|_{(\tau)}$  and  $\mathfrak{X}^{(K)}|_{(\tau)}$  be the matrix row spaces without the last  $\tau$  columns, and also  $(\tau) | \mathfrak{X}^{(K,1)}$ ,  $(\tau) | \mathfrak{X}^{(K,2)}$  and  $(\tau) | \mathfrak{X}^{(K)}$  be the matrix row spaces without the first  $\tau$  columns.

**Lemma 7.** *If a series is governed by an LRR of rank  $d$ ,  $d \leq \min(K, L - 2\tau)$ , then  $\mathfrak{X}^{(K)}|_{(\tau)} = (\tau) | \mathfrak{X}^{(K)}$  and  $\dim(\mathfrak{X}^{(K)}|_{(\tau)}) = \dim((\tau) | \mathfrak{X}^{(K)}) = \dim(\mathfrak{X}^{(K)}) = d$ .*

*Proof.* To prove the equality of subspaces, since in the matrix  $\mathbf{X}|_{(\tau)}$  the rows from  $\tau + 1$ -th to  $L$ -th coincide with the rows from the first to  $L - \tau$ -th of the matrix  $(\tau) | \mathbf{X}$ , it remains to show that the first  $\tau$  rows of the matrix  $\mathbf{X}|_{(\tau)}$  belong to the space  $(\tau) | \mathfrak{X}^{(K)}$  and the rows from  $(L - \tau + 1)$ -th to  $L$ -th belong to  $\mathfrak{X}^{(K)}|_{(\tau)}$ . Indeed, this is true since the series is governed by the LRR and the specified rows are linear combinations of matched rows.

Let us now prove the equality of the dimensions of incomplete and complete spaces. Evidently,  $\dim(\mathfrak{X}^{(K)}|_{(\tau)}) = \dim((\tau)|\mathfrak{X}^{(K)})$  and  $\dim(\mathfrak{X}^{(K)}|_{(\tau)}) \leq \dim(\mathfrak{X}^{(K)})$ . Consider the basis  $V_1, \dots, V_d$  in  $\mathfrak{X}^{(K)}$ . The corresponding vectors  $\underline{V}_1, \dots, \underline{V}_d \in \mathfrak{X}^{(K)}|_{(\tau)}$  are linearly independent, since otherwise they could be continued by the LRR to dependent vectors that coincide in uniqueness with  $V_1, \dots, V_d$ . The lemma is proved.  $\square$

Here is another lemma that will be used in the derivation of strong SSA-AMUSE separability conditions.

**Lemma 8.** *Let the matrix  $\mathbf{S} = [S_1 : \dots : S_d]$  have rank  $d$ , and for fixed  $\tau > 0$  the matrix  $\mathbf{C}_\tau(\mathbf{S})$  is diagonal:  $\mathbf{C}_\tau(\mathbf{S}) = \text{diag}(\mu_1, \dots, \mu_d)$ ,  $\mu_1 \geq \mu_2 \geq \dots \geq \mu_d$ , matrix  $\mathbf{W}$  is orthogonal,  $\mathbf{T} = \mathbf{SW}^T$ . If for some  $I_1$  and  $I_2$  the sets  $\{\mu_i, i \in I_1\}$  and  $\{\mu_j, j \in I_2\}$  do not intersect, then for any spectral decomposition of the matrix  $\mathbf{C}_\tau(\mathbf{T}) = \mathbf{HDH}^T$  with eigenvalues on the diagonal of  $\mathbf{D}$  ordered in descending order, the following statements are valid:*

1.  $\mathbf{D} = \mathbf{C}_\tau(\mathbf{S})$ ,
2.  $\text{span}(H_k : k \in I_1) = \text{span}(W_l : l \in I_1)$  and  $\text{span}(H_k : k \in I_2) = \text{span}(W_l : l \in I_2)$ , where  $H_i$ ,  $i = 1, \dots, d$ , are the columns of the matrix  $\mathbf{H}$ ,  $W_j$ ,  $j = 1, \dots, d$ , are the columns of the matrix  $\mathbf{W}$ .

*Proof.* By Lemma 6,  $\mathbf{C}_\tau(\mathbf{S})$  and  $\mathbf{C}_\tau(\mathbf{T})$  are similar, so they have the same eigenvalues, which proves the first statement of the lemma. The columns of the matrix  $\mathbf{W}$  are orthonormalized eigenvectors of the matrix  $\mathbf{C}_\tau(\mathbf{T})$ . Since the subsets of eigenvalues do not intersect, the eigenspaces corresponding to these subsets are uniquely defined. Vectors  $W_i$ ,  $i \in I_1$ , and  $W_j$ ,  $j \in I_2$ , are orthogonal bases of these subspaces, and vectors  $H_i$ ,  $i \in I_1$ , and  $H_j$ ,  $j \in I_2$ , form another orthogonal bases of the same subspaces.  $\square$

**Separability conditions** Let  $\mathbf{X}_N = \mathbf{X}_N^{(1)} + \mathbf{X}_N^{(2)}$ ,  $\mathbf{Y}^{(1)}$  and  $\mathbf{Y}^{(2)}$  be trajectory matrices of the time series  $\mathbf{X}_N^{(1)}$  and  $\mathbf{X}_N^{(2)}$ .

It is easy to see that if the row spaces  $(\tau)|\mathfrak{X}^{(K,1)}$  and  $(\tau)|\mathfrak{X}^{(K,2)}$  of matrices  $(\tau)|\mathbf{Y}^{(1)}$  and  $(\tau)|\mathbf{Y}^{(2)}$  are orthogonal, then the series  $\mathbf{X}_N^{(1)}$  and  $\mathbf{X}_N^{(2)}$  are weakly SSA-AMUSE separable. We obtain the necessary and sufficient conditions for the SSA-AMUSE separability.

**Theorem 4.** *The series  $\mathbf{X}_N^{(1)}$  and  $\mathbf{X}_N^{(2)}$  are weakly SSA-AMUSE separable if and only if*

$$\mathbf{Y}^{(1)}|_{(\tau)} \left[ \mathbf{Y}|_{(\tau)}^{(2)} \right]^T +_{(\tau)} |\mathbf{Y}^{(1)}|_{(\tau)} \left[ (\tau)|\mathbf{Y}^{(2)} \right]^T = 0, \quad (2.15)$$

$$\mathbf{Y}^{(1)}|_{(\tau)} \left[ (\tau)|\mathbf{Y}^{(2)} \right]^T +_{(\tau)} |\mathbf{Y}^{(1)}|_{(\tau)} \left[ \mathbf{Y}^{(2)}|_{(\tau)} \right]^T = 0. \quad (2.16)$$

*Proof.* Here is a scheme of the proof.

Necessity. It follows from separability that there exists a basis  $\{S_1, \dots, S_d\}$  of the space  $\mathfrak{X}^{(K)}$  that splits into subsets  $\{S_1^{(1)}, \dots, S_{d_1}^{(1)}\}$  and  $\{S_1^{(2)}, \dots, S_{d_2}^{(2)}\}$  such that the first is the basis  $\mathfrak{X}^{(K,1)}$ , and the second is the basis  $\mathfrak{X}^{(K,2)}$ . Moreover, if we denote  $\mathbf{S}^{(i)} = [S_1^{(i)} : \dots : S_{d_i}^{(i)}]$ ,  $i = 1, 2$ ,  $\mathbf{S} = [\mathbf{S}^{(1)} : \mathbf{S}^{(2)}]$  (we reordered the vectors for convenience), then the matrix  $\mathbf{C}_\tau(\mathbf{S})$  is diagonal. Denote  $\mathbf{\Xi}^{(i)} = \left[ \left( \overline{\mathbf{S}^{(i)}}^{(\tau)} \right)^T : \left( \underline{\mathbf{S}^{(i)}}^{(\tau)} \right)^T \right]^T$ . The columns of matrix  $\mathbf{\Xi}^{(1)}$  are orthogonal to columns of  $\mathbf{\Xi}^{(2)}$ , and the columns of matrix  $\mathbf{\Xi}^{(i)}$  form an orthogonal basis of the row space of the matrix  $[\cdot]_{(\tau)} | \mathbf{Y}^{(i)}]$ . Hence the equality (2.15) follows.

Using the separability condition, we can show that the matrix  $\mathbf{P}^{(i)} = \mathbf{U} \boldsymbol{\Lambda}^{1/2} \mathbf{W}^{(i)} (\mathbf{W}^{(i)})^T \boldsymbol{\Lambda}^{-1/2} \mathbf{U}^T$  defines an orthogonal projector on the column space of the matrix  $\mathbf{Y}^{(i)}$ . Using this, we derive the second condition from the diagonality of the matrix  $\mathbf{C}_\tau(\mathbf{S})$ . The matrix can be written in the form

$$\mathbf{C}_\tau(\mathbf{S}) = \begin{pmatrix} \mathbf{C}_\tau(\mathbf{S}^{(1)}) & \mathbf{C}^{(1,2)} \\ \mathbf{C}^{(1,2)} & \mathbf{C}_\tau(\mathbf{S}^{(2)}) \end{pmatrix},$$

where

$$\mathbf{C}^{(1,2)} = (\mathbf{W}^{(1)})^T \boldsymbol{\Lambda}^{-1/2} \mathbf{U}^T \left[ \mathbf{Y}^{(1)}|_{(\tau)} \left( \cdot_{(\tau)} | \mathbf{Y}^{(2)} \right)^T +_{(\tau)} | \mathbf{Y}^{(1)} T \left( \mathbf{Y}|_{(\tau)}^{(2)} \right) \right] \mathbf{U} \boldsymbol{\Lambda}^{-1/2} \mathbf{W}^{(2)}/2$$

and equals zero by the diagonality condition. We multiply by  $(\mathbf{W}^{(i)})^T \boldsymbol{\Lambda}^{-1/2} \mathbf{U}^T$  to the right and by  $\mathbf{U} \boldsymbol{\Lambda}^{1/2} \mathbf{W}^{(i)}$  to the left, then use the projector properties of  $\mathbf{P}^{(i)}$  and obtain the condition (2.16).

Sufficiency. By the condition (2.15) and by Lemma 5 there exist bases  $\mathbf{S}^{(1)} \in \mathfrak{X}^{(K,1)}$  and  $\mathbf{S}^{(2)} \in \mathfrak{X}^{(K,2)}$  such that the columns of the matrix  $\mathbf{\Xi}^{(1)}$  are orthogonal to each other and to the columns of  $\mathbf{\Xi}^{(2)}$ . From this, we can deduce that the columns of the matrix  $\mathbf{S} = [\mathbf{S}^{(1)} : \mathbf{S}^{(2)}]$  form the basis of  $\mathfrak{X}^{(K)}$ . It follows from the condition (2.16) that the matrix  $\mathbf{C}_\tau(\mathbf{S})$  is diagonal. Next, it remains to project the trajectory matrices onto the found basis and find the elementary matrices:  $\mathbf{Y}^{(i)} = \mathbf{Y}^{(i)} \mathbf{S}^{(i)} \left( (\mathbf{S}^{(i)})^T \mathbf{S}^{(i)} \right)^{-1} (\mathbf{S}^{(i)})^T$ , which means separability.  $\square$

**Corollary 5.** *If  $\mathfrak{X}^{(K,1)}|_{(\tau)}$  is orthogonal to  $\mathfrak{X}^{(K,2)}|_{(\tau)}$ , then the series is weakly SSA-AMUSE separable.*

**Remark 18.** *By Lemma 7,  $\mathfrak{X}^{(K,i)}|_{(\tau)} =_{(\tau)} | \mathfrak{X}^{(K,i)}$ . Therefore, instead of  $\mathfrak{X}^{(K,i)}|_{(\tau)}$  in Corollary 5, we can use  $_{(\tau)} | \mathfrak{X}^{(K,i)}$ .*

**Theorem 5.** *Let the time series  $X_N^{(1)}$  and  $X_N^{(2)}$  be weakly SSA-AMUSE separable. Denote  $\mathbf{Y}^{(1)}$  and  $\mathbf{Y}^{(2)}$  their trajectory matrices. Let also  $\mathbf{Q}^{(1)}$  be some basis for the first series and  $\mathbf{Q}^{(2)}$  be some*

basis for the second series, such that

$$\left[{}_{(\tau)}|\mathbf{Y}^{(i)} : \mathbf{Y}^{(i)}|_{(\tau)}\right] = \mathbf{U}^{(i)} \left(\boldsymbol{\Lambda}^{(i)}\right)^{1/2} \left[ \left( \overline{(\mathbf{Q}^{(i)})^{(\tau)}} \right)^T : \left( \underline{(\mathbf{Q}^{(i)})}_{(\tau)} \right)^T \right], i = 1, 2,$$

are the SVDs. Then strong SSA-AMUSE separability is equivalent to the fact that the eigenvalues of the matrices  $\mathbf{C}_\tau(\mathbf{Q}^{(1)})$  and  $\mathbf{C}_\tau(\mathbf{Q}^{(2)})$  are not overlapping.

*Proof.* The proof of the theorem is based on Lemma 8, according to which the components of the expansion, corresponding to separable components of the series, cannot be mixed.  $\square$

Note that the strong separability conditions do not depend on the choice of bases  $\mathbf{Q}^{(i)}$ ,  $i = 1, 2$ , satisfying the conditions of the theorem.

The following proposition shows how the weak separability conditions are related when applying Basic SSA and SSA-AMUSE. Its proof directly follows from the formulation of the separability conditions.

**Proposition 11.** *Let  $X^{(1)}$  and  $X^{(2)}$  be infinite series governed by LRRs. If segments of these infinite series of length  $N$  are weakly separable in Basic SSA for a window lengths  $L$ , then segments of the series lengthened by  $\tau$  (that is, of length  $N + \tau$ ) will be weakly SSA-AMUSE separable for the same  $L$  and  $\tau$ . Also, the segments of length  $N$  will be weakly SSA-AMUSE separable for the window length  $L - \tau$  and the shift  $\tau$ .*

**Remark 19.** *Note that for the SSA-AMUSE separability, only the orthogonality of the segments of length  $K - \tau$  is necessary, whereas for Basic SSA the separability conditions depend on both  $K$  and  $L$ .*

**Examples of weak and strong separability** Here are some examples that show the features and pros of SSA-AMUSE separability. Below we will consider only the case  $\tau = 1$ , because it is enough to show the advantages of the SSA-AMUSE method.

1. Here is an example that the condition  ${}_{(\tau)}|\mathfrak{X}^{(K,1)} \perp_{(\tau)} |\mathfrak{X}^{(K,2)}$  is only a sufficient condition for weak separability. Consider  $X^{(1)} = (\cos \pi n, n = 1, \dots, N)$  and  $X^{(2)} = (\text{const}, n = 1, \dots, N)$ . If  $K$  is odd, the spaces are not orthogonal, but the conditions (2.15) and (2.16) are satisfied. Thus, these series are weakly SSA-AMUSE separable.
2. Let us obtain the conditions for the strong separability of two harmonics. Consider the series  $X^{(1)}$  and  $X^{(2)}$  of length  $N$  with common terms in the form  $x_n^{(i)} = A_i \sin(2\pi n \omega_i + \gamma_i)$ ,  $n = 1, \dots, N$ ,  $0 < \omega_i \leq 0.5$ ,  $\omega_1 \neq \omega_2$ . Let us prove that if  $(K-1)\omega_1$  and  $(K-1)\omega_2$  are integers, then the time series are strongly SSA-AMUSE separable. Weak SSA-AMUSE separability follows from weak separability in Basic SSA (see statement 11). Strong separability can be

proved as follows. Let  $\omega_i < 0.5$  for certainty. In this case, the bases found at step 3 of the SSA-AMUSE algorithm will be orthogonal and have the form:

$$\begin{aligned} S_1^{(i)} &= (A \sin(2\pi\omega_i + \zeta_i), \dots, A \sin(2\pi\omega_i K + \zeta_i))^T, \\ S_2^{(i)} &= (A \cos(2\pi\omega_i + \zeta_i), \dots, A \cos(2\pi\omega_i K + \zeta_i))^T, \end{aligned}$$

where  $A = 1/\sqrt{K-1}$ .

Moreover, for such bases, each of the matrices  $\mathbf{C}_\tau(\mathbf{S}^{(i)})$ ,  $i = 1, 2$ , is diagonal, with numbers  $\cos(2\pi\omega_i)/2$  on the diagonals, which are also eigenvalues of these matrices. Therefore, the conditions of strong separability in Theorem 5 are satisfied.

Note that in the case of SSA-AMUSE, the components will be ordered by the values  $\cos(2\pi\omega_i)$ , whereas in Basic SSA they are ordered by the amplitudes  $A_i$ . The ordering by  $A_i$  leads to mixing in the case of matching amplitudes.

**SSA-AMUSE asymptotic separability** In real-life problems, the exact separability conditions are rarely fulfilled and therefore the approximate separability takes place, which is a consequence of asymptotic separability.

The notion of asymptotic SSA-AMUSE weak separability is defined similarly to the asymptotic separability in the case of Basic SSA by replacing the exact orthogonality condition with the asymptotic one when the length of the series  $N$  tends to infinity. In the case of SSA-AMUSE, it is not required that  $L$  tends to infinity, and it is sufficient that only  $K = N - L + 1 \rightarrow \infty$  at  $N \rightarrow \infty$  (see note 19).

Thus, those time series that were weakly asymptotically separable by Basic SSA remain weakly asymptotically separable with SSA-AMUSE, and  $L$  may not tend to infinity.

Let us consider the notion of asymptotic strong separability. The condition for strong separability in SSA-AMUSE is the disjunction of the sets of the eigenvalues of the matrices  $\mathbf{C}_\tau(\mathbf{S}^{(K,i)})$ , where  $\mathbf{S}^{(K,i)}$  are matrices whose columns contain the bases found in step 3 of the SSA-AMUSE algorithm. Let us call the condition of asymptotic strong separability of asymptotically weakly separable series the disjunction of the sets of the limit eigenvalues of the matrices  $\mathbf{C}_\tau(\mathbf{S}^{(K,i)})$  at  $K \rightarrow \infty$  if such limits exist.

The harmonics  $X^{(1)}$  and  $X^{(2)}$  with common terms in the form  $x_n^{(i)} = A_i \sin(2\pi n \omega_i + \gamma_i)$ ,  $n = 1, \dots, N$ , are asymptotically strongly separable even with coincident amplitudes as  $K \rightarrow \infty$ . This is proved by direct calculation of the eigenvalues of the matrices  $\mathbf{C}_\tau(\mathbf{S}^{(K,i)})$ .

## Comparison

Let us compare SSA-AMUSE and DerivSSA described in Section 2.3.1. The methods have a lot in common. In particular, they are both nested, i.e., they are already used after signal extraction from noise, and also both serve for improving strong separability.

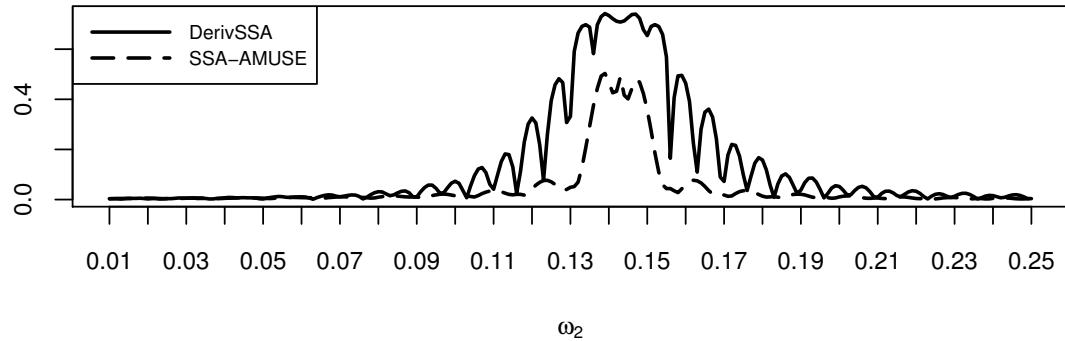


Figure 2.24: RMSE of the estimates of the series with common term  $\sin(2\pi n/7)$  if  $\sin(2\pi n/7) + \sin(2\pi n\omega_2)$  is observed; depending on  $\omega_2$ .

One of the advantages of SSA-AMUSE over DerivSSA is the ability to exactly separate components, which is not available in DerivSSA. A disadvantage of SSA-AMUSE is that the resulting decomposition is not a decomposition into orthogonal Frobenius matrices.

Nevertheless, the advantages listed above lead to improved accuracy in separating the components of the time series. Let us demonstrate this in a numerical experiment. Figure 2.24 shows the RMSE errors of estimating one harmonic component from the observed sum of two sine waves. The parameter  $\omega_2$  varies from 0.01 to 0.25 in steps of 0.001. Naturally, there is no separability when  $\omega_2 \approx \omega_1 = 1/7$ . However, it can be seen that the area with a large error around  $\omega_2 = 1/7 \approx 0.14$  is narrower for SSA-AMUSE. When noise is added to the signal, the comparison result is similar.

## 2.5. Automatic identification

In this section, we will consider the problem of automatic identification of decomposition components in SSA, see the recent paper [129]. Let us start with the definition of the components of the time series that we want to identify.

Consider a time series  $\mathbf{X} = (x_1, \dots, x_N)$ ,  $x_i \in \mathbb{R}$ .

We call a low-frequency component *trend*. For a one-dimensional real-valued time series, this is the component  $\mathbf{T}$ , for which in the Fourier expansion of the series  $\mathbf{T}$

$$t_n = C_0 + \sum_{k=1}^{\lfloor (N-1)/2 \rfloor} \sqrt{C_k^2 + S_k^2} \cos(2\pi nk/N + \phi_k) + C_{N/2}(-1)^n$$

the largest values have the coefficients  $\sqrt{C_k^2 + S_k^2}$  with a small value of  $k$ ; the last summand  $C_{N/2}(-1)^n$  is present only if  $N$  is even.

Various regular oscillations will be called *oscillatory components*. Formally, we will consider a

sum of exponentially-modulated (e-m) harmonic series (harmonics). For a one-dimensional real series, the  $n$ th element of an e-m harmonic with frequency  $\omega$  ( $\omega \leq 0.5$ ) is given by the expression:  $a e^{\alpha n} \cos(2\pi\omega n + \phi)$ ,  $0 \leq \phi < 2\pi$ ,  $a \neq 0$ .

A sum of the trend and the oscillatory components will be called *signal*.

We assume that the time series  $X$  contains the following additive components: trend  $T$ , oscillatory component  $P$  and random noise  $N$ . Thus, in general, the considered model of the object  $X$  looks like this:

$$X = T + P + N,$$

where the elements of  $N$  are realizations of a random variable.

### 2.5.1. Low-frequency method for trend identification

The considered automation method was introduced in [42]. An approach to the automatic selection of parameters of the automation method is proposed in [130]; we will not consider it here. The automation method allows identification of the components related to the trend. We will call this method the *low-frequency method for trend identification*.

For a series  $Y$  of length  $M$  and

Let us introduce the periodogram:

$$\Pi_Y^M(k/M) = \frac{M}{2} \begin{cases} 2C_0^2 & \text{for } k = 0, \\ C_k^2 + S_k^2 & \text{for } 0 \leq k \leq M/2, \\ 2C_{M/2}^2 & \text{for } k = M/2, \text{ if } M \text{ even,} \end{cases} \quad (2.17)$$

where the coefficients  $C_k$  and  $S_k$  are taken from the Fourier decomposition of  $Y = (y_1, \dots, y_M)$ :

$$y_n = C_0 + \sum_{k=1}^{\lfloor M/2 \rfloor} \left( C_k \cos(2\pi n k / M) + S_k \sin(2\pi n k / M) \right).$$

For a series  $Y$  of length  $M$  and for  $0 \leq \omega_1 \leq \omega_2 \leq 0.5$ , we define

$$T(Y; \omega_1, \omega_2) = \sum_{k: \omega_1 \leq k/M < \omega_2} I_Y^M(k/M), \quad (2.18)$$

where

$$I_Y^M(k/M) = \Pi_Y^M(k/M) / \|Y\|^2, \quad (2.19)$$

$\Pi_Y^M$  is defined in (2.17). Since we have  $\|Y\|^2 = \sum_{k=1}^{\lfloor M/2 \rfloor} \Pi_Y^M(k/M)$ , the measure  $T(Y; \omega_1, \omega_2)$  can be considered as a proportion of frequencies contained in the frequency bin  $[\omega_1, \omega_2]$ .

One of the aims in performing grouping is the extraction of a series component with frequency

range mostly from the chosen frequency bin. Therefore, it is natural to calculate the value of  $T$  for elementary reconstructed components. Moreover, SSA reconstruction can be considered as a linear filter. It appears that the frequency response of the filter generated by the  $i$ th eigentriple is almost the same as the periodogram of the corresponding singular vector, see [29, Proposition 3.13]. Therefore, it is reasonable to apply  $T$  also to singular vectors to reconstruct the series components with the given frequency ranges.

Since the trend of a series can be defined as its slowly varying series component, for extracting a trend, a frequency bin in the form  $[0, \omega)$  should be chosen. Then we consider

$$T(\mathbf{Y}; \omega) = \sum_{k:k/M < \omega} I_{\mathbf{Y}}^M(k/M). \quad (2.20)$$

The value of  $\omega$  reflects the frequency range, which we associated with a trend. For example, if the series has monthly seasonality,  $\omega$  should be notably smaller than  $1/12$ . Note that the grouping method does not answer the question of whether the extracted component is indeed a deterministic trend or simply a result of smoothing.

Values of  $T$  for each elementary decomposition component can be used for performing the grouping. To perform an automatic grouping, a threshold  $T_0$ ,  $0 \leq T_0 \leq 1$ , should be given. For example, if the value  $T(\mathbf{Y}_i; 0, \omega)$  is larger than  $T_0$  for some small  $\omega$ , where  $\mathbf{Y}_i$  is the  $i$ th elementary series or  $i$ th left/right singular vector, then the corresponding eigentriple can be automatically considered as a part of the trend.

The suggested method is presented in Algorithm 2.13.

#### ALGORITHM 2.13: 1D-SSA: Frequency identification of trend components, by the threshold

*Input:* Frequency range  $[\omega_1, \omega_2]$ , threshold  $T_0$ , group  $I$ , type of series: eigenvectors, factor vectors or reconstructed series.

*Output:* A group of components  $J \subset I$ .

- 1: For each series  $\mathbf{Y}_i$ ,  $i \in I$ , the measure  $T(\mathbf{Y}_i; \omega_1, \omega_2)$  given in (2.18) is calculated.
- 2: The resultant group  $J$  consists of indices  $i \in I$  such that  $T(\mathbf{Y}_i; \omega_1, \omega_2) \geq T_0$ .

#### 2.5.2. Frequency method for identifying the oscillating component

Here we define the oscillatory component as a sum of the e-m harmonics. Let us consider the algorithm for the automatic identification of the e-m harmonics. We will call this method *frequency method of identification of the oscillatory component*.

The method was suggested in [64] and further developed in [36] and [130]; it is based on the study of periodograms of singular vectors corresponding to the series. The method consists of two parts, at the first stage a preliminary check is carried out, and the singular triples identified at this stage are further checked at the second stage.

As we know from Proposition 1, an e-m harmonic (1.18) can have rank 1 if frequency  $\omega = 0.5$ , or rank 2 otherwise, i.e. the e-m harmonic can correspond to either one singular vector or two.

The considered method for automatically identifying the components corresponding to e-m harmonics checks

- each pair of singular vectors if they are similar to e-m harmonics with the same frequency,
- each singular vector if it is similar to an e-m harmonic with period 2 ( $\omega = 0.5$ ).

**Remark 20.** Since under some non-restrictive conditions a one-dimensional real-valued harmonic series with  $\omega < 0.5$  produces two equal (Proposition 2) or close ([1]) eigenvalues of the trajectory matrix and the SSA method algorithm sorts components of the SVD by eigenvalues, it is sufficient to consider only consecutive pairs of singular vectors to identify harmonics of rank 2.

According to Section 1.8.4, a harmonic series of the form (1.18) corresponds to two singular vectors of the form (1.19) with equal frequencies.

The following result is known; the normalized periodogram  $I_Y^M$  is given by (2.19).

**Proposition 12.** [130, Proposition 3.1] Let  $\alpha = 0$ ,  $\omega < 0.5$  and  $L\omega \in \mathbb{N}$ . Then for singular vectors  $U_1, U_2$  of time series  $S$  with elements given by (1.18)

$$\max_{0 \leq k \leq L} I_{U_1}^L(k/L) = \max_{0 \leq k \leq L} I_{U_2}^L(k/L) = I_{U_1}^L(\omega) = I_{U_2}^L(\omega) = 1.$$

The first step of the method is based on the fact that if for a harmonic ( $\alpha = 0$ ) with frequency  $\omega$  and window length  $L$  the relation  $L\omega \in \mathbb{N}$  is fulfilled, then the periodograms of its singular vectors are  $I_{U_j}^L(k/L) = \chi_\omega(k/L)$  (where  $\chi_\omega(\cdot)$  are the indicator of set  $\{\omega\}$ ). Given only approximate separability when  $L\omega \notin \mathbb{N}$  or  $\alpha \neq 0$ , at the first stage of the method, we choose from all consecutive pairs of singular vectors those for which arguments of periodogram maxima are greater than 0 and close, i.e. differ from each other by no more than  $s_0/L$  (where  $s_0 \in \mathbb{Z}_+$  is a fixed parameter of the method):

$$J_1^{(P)} = \{(i, i+1) : \theta_1, \theta_2 > 0, L|\theta_i - \theta_{i+1}| \leq s_0, 1 \leq i \leq d-1\}, \quad (2.21)$$

where  $\theta_j = \arg \max_{0 < k \leq L/2} \{I_{U_j}^L(k/L)\}$  is the argument of the maximum of the periodogram  $I_{U_j}^L$  of the singular vector  $U_j$ .

Similarly, each singular vector is checked against a harmonic with a frequency of 0.5:

$$J_2^{(P)} = \{i : L|\theta_i - 0.5| \leq s_0, 1 \leq i \leq d\}. \quad (2.22)$$

The results of the first step are the sets  $J_1^{(P)}$  and  $J_2^{(P)}$  of the singular vectors' numbers. The set  $J_1^{(P)}$  consists of pairs of numbers of singular vectors identified as corresponding to e-m harmonics with  $0 < \omega < 0.5$ . The set  $J_2^{(P)}$  contains numbers of singular vectors identified as corresponding to e-m harmonics with  $\omega = 0.5$ .

**Definition 9.** Let  $\mathcal{A} = \{W_j\}$ ,  $W_j$  be a finite set of real vector vectors, denote the power of the set as  $\#\mathcal{A}$ . Let us introduce a function  $\rho_{\mathcal{A}}$  defined on the set  $\mathcal{A}$ :

$$\rho_{\mathcal{A}}(k/L) = \frac{1}{\#\mathcal{A}} \sum_{W_j \in \mathcal{A}} I_{W_j}^L(k/L)$$

In view of Proposition 12, to check that the singular vectors with numbers  $j, j+1$  correspond to a harmonic, we must ensure that their periodogram maxima are reached at one point and that each of them has a value close to 1, since e-m harmonic with frequency  $\omega$  such that  $L\omega \notin \mathbb{N}$  also corresponds to a large value of the periodogram maximum (though smaller than 1), because its periodogram has one distinct peak. This is what is checked in the second step of the method.

Before [36], based on which we describe the method, the considered approach was proposed in [64], but the idea itself is not new and was proposed back in 1929 by Fisher in [131].

Thus, the second step of the method is to choose a threshold  $\rho_0$  for the measure

$$\rho_{i,j} := \max_{0 < k \leq L/2} \left( \rho_{\{U_i, U_j\}(k/L)} + \rho_{\{U_i, U_j\}}((k+1)/L) \right), \quad (2.23)$$

where  $U_i, U_j$  are a pair of singular vectors with indices from the set of indices  $J_1^{(P)}$  selected in the first step.

To identify the components belonging to a harmonic of frequency 0.5, introduce the measure in the form

$$\rho_i := \rho_{\{U_i\}}((\lfloor L/2 \rfloor)/L) + \rho_{\{U_i\}}((\lfloor L/2 \rfloor + 1)/L), \quad (2.24)$$

where  $i \in J_2^{(P)}$ .

The final result of the frequency method implemented by Algorithm 2.14 is the indices

$$J^{(P)} = \{(i, j) \in J_1^{(P)} : \rho_{i,j} \geq \rho_0\} \cup \{i \in J_2^{(P)} : \rho_i \geq \rho_0\}. \quad (2.25)$$

#### ALGORITHM 2.14: 1D-SSA. Frequency identification of oscillatory components

*Input:* The data and parameters are as follows.

1. **Data:** left singular vectors  $\{U_i\}_{i=1}^d$ .
2. **Parameters:** parameter  $s_0 \in \mathbb{Z}_+$ , threshold  $\rho_0 \in [0, 1]$ .

*Output:* A group of component indices  $J^{(P)}$  related to the oscillatory component.

- 1: Based on  $\{U_i\}_{i=1}^d$ , we obtain the index group  $J_1^{(P)}$  using (2.21) with  $s_0$  and the index group  $J_2^{(P)}$  using (2.22) with  $s_0$ .

2: Obtain the index group  $J^{(P)}$  using (2.25) with  $\rho_0$  applied to  $\{U_j\}_{j \in J_1^{(P)} \cup J_2^{(P)}}$ .

---

The author [130] advises taking  $s_0 = 1$ . The disadvantage of the method is the fact that it is originally designed to identify unmodulated harmonics. When applied to simulated harmonics, the threshold [36] must be reduced, which can lead to false detection of harmonics in the unmodulated case ( $\alpha = 0$ ).

### 2.5.3. Method for identifying the oscillatory component by the regularity of angles

#### Description and justification

Let  $P = (p_1, \dots, p_L)^T$  and  $Q = (q_1, \dots, q_L)^T$  be two real vectors of length  $L$ . Introduce the measure

$$\tau(P, Q) := \hat{D}(\Theta) = \frac{1}{L-1} \sum_{k=1}^{L-1} (\theta_k - \bar{\theta})^2, \quad (2.26)$$

where  $\Theta = (\theta_1, \dots, \theta_L)^T$ ,  $\bar{\theta} = \sum_{k=1}^{L-1} \theta_k / (L-1)$ ,  $\theta_k$  is the angle between  $(p_k, q_k)^T$  and  $(p_{k+1}, q_{k+1})^T$ , that is,

$$\theta_k = \arccos \left( \frac{p_k p_{k+1} + q_k q_{k+1}}{\sqrt{p_k^2 + q_k^2} \sqrt{p_{k+1}^2 + q_{k+1}^2}} \right).$$

Note that the values of  $\theta_k$  belong to  $[0, \pi]$ .

**Proposition 13.** *For singular vectors  $U_1$  and  $U_2$  of the time series  $S$  with elements given by (1.19), the following statements are valid.*

1. *If  $\alpha = 0$  and  $L\omega$  is integer, then  $\tau(U_1, U_2) = 0$ .*
2. *If  $\alpha = 0$  and  $L = [\beta N]$ , where  $0 < \beta < 1$ , then  $\lim_{L \rightarrow \infty} \tau(U_1, U_2) = 0$ .*
3. *If  $\alpha = \alpha_N = C/N$ , where  $C$  is some constant, and  $L = [\beta N]$ , where  $0 < \beta < 1$ , then  $\lim_{L \rightarrow \infty} \tau(U_1, U_2) = 0$ .*

Let us describe the approach to identifying an oscillating component based on the properties of the measure  $\tau$ . The equality  $\tau(P, Q) = 0$  means that the angles between the sequential points  $(p_k, q_k)^T$  are equal. If the norms of the vectors  $P$  and  $Q$  are monotonically changed in  $k$ , then the 2D diagrams of  $P$  and  $Q$  look like spirals.

Therefore, in conditions of Proposition 13 for  $d = 2$  and sufficiently large  $L$ , the series (1.18) can be identified with a two-dimensional diagram of singular vectors  $U_1$  and  $U_2$  and the measure  $\tau$  will be close to zero. Examples of resulting images with  $\langle\langle$ spirals $\rangle\rangle$  for series with  $N = 99$ ,  $L = 50$  are presented in Fig. 2.25.

Under the conditions of Proposition 13 for  $d = 2$ , at sufficiently large  $L$ , the series (1.18) can be identified with the help of a two-dimensional diagram of singular vectors  $U_1$  and  $U_2$ , which corresponds to the measure  $\tau$  close to zero. Fig. 2.25 contains several examples of resulting images with  $\langle\langle$ spirals $\rangle\rangle$  for series with  $N = 99$ ,  $L = 50$ .

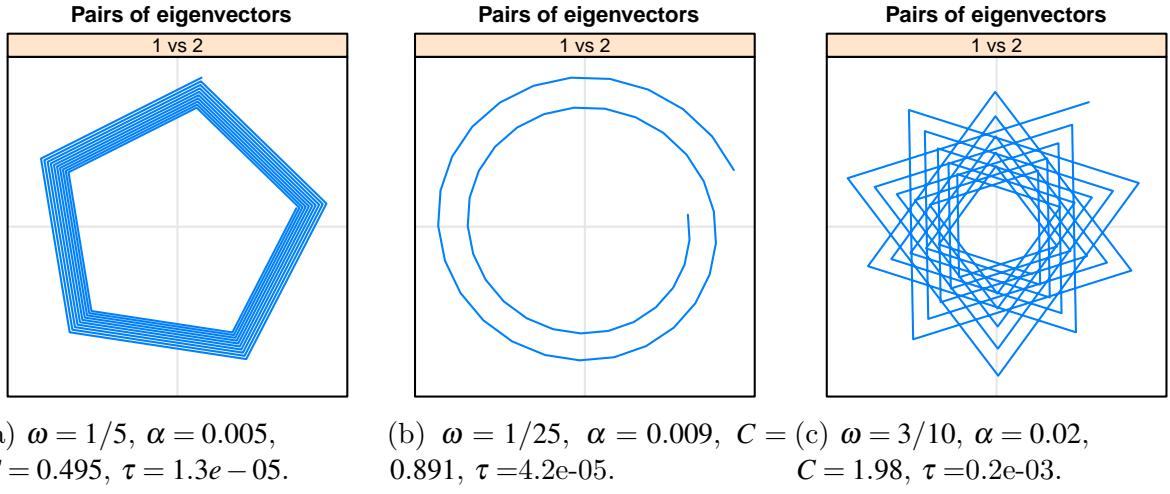


Figure 2.25: Two-dimensional diagrams of singular vectors of real e.m. harmonics,  $N = 99$ ,  $L = 50$ ,  $\beta = 1/2$ .

Note that in practice we do not deal with series of infinite length, so for a series of finite length  $N$  with a constant value  $\alpha$  the condition  $\alpha = C/N$  means not too large value  $e^{\alpha N}$ , i.e. the limited range of series values.

If the number of e-m harmonics we want to identify is known, the algorithm consists simply in selecting pairs of vectors  $U_i$  and  $U_{i+1}$  with minimal values of the measure  $\tau(U_i, U_{i+1})$ ,  $i = 1, \dots, d - 1$ .

In the case where the number of e-m harmonics is unknown, pairs of singular vector pairs can be selected by using a threshold, i.e., those vectors  $U_i$  and  $U_{i+1}$  whose value  $\tau(U_i, U_{i+1})$  is less than a given threshold can be assigned to a harmonic, but there is no theoretical justification for choosing the threshold. Therefore, in this section, we present an empirical justification for the choice of the threshold and conduct numerical studies using the example of a noisy e-m cosine.

Note that the method does not work for the case  $\omega = 0.5$ . Therefore, we do not consider it in this section.

We will call this method *the method for identifying the oscillatory component by the regularity of angles*.

## Algorithm

All the explanations given in the previous section explained why the value of the measure  $\tau$  will be 0 for the singular vectors of the e-m harmonic. The question arises whether a value  $\tau$  close to 0 can be obtained for singular vectors of not the e-m harmonic.

For example, if some vectors  $P$  and  $Q$  give, although different, small values of the angles  $\theta_k$  given in the definition (2.26) of the measure  $\tau$ , then the variance  $\hat{D}(\Theta)$  from the same definition

will also be small. To overcome this problem, we will use a normalized version of the measure  $\tau$  for the vectors  $P = (p_1, \dots, p_L)^T$  and  $Q = (q_1, \dots, q_L)^T$ :

$$\tilde{\tau}(P, Q) := \frac{\tau(P, Q)}{\min(1, \bar{\theta}^2)} = \frac{\hat{D}(\Theta)}{\min(1, \bar{\theta}^2)} = \frac{1}{(L-1)\min(1, \bar{\theta}^2)} \sum_{k=1}^{L-1} (\theta_k - \bar{\theta})^2, \quad (2.27)$$

where, as before,  $\Theta = (\theta_1, \dots, \theta_L)^T$ ,  $\bar{\theta} = \sum_{k=1}^{L-1} \theta_k / (L-1)$  and  $\theta_k$  is the angle between  $(p_k, q_k)^T$  and  $(p_{k+1}, q_{k+1})^T$ .

Dividing by  $\min(1, \bar{\theta}^2)$  can only increase the value of the measure, so it fights the problem of small angle values. Small values can be also produced by angles for singular vectors of the e-m harmonic; however, then the variance will be close to zero and normalization will not play a significant role.

For  $\tilde{\tau}$  as well as for  $\tau$ , the statement 13 holds, since if  $\tau(P, Q) = 0$ , then obviously also  $\tilde{\tau}(P, Q) = 0$ .

As discussed in the previous section, the method has two modifications: for the case where the number of e-m harmonics is known and for the case where the number of e-m harmonics is unknown. These two modifications differ in the stopping criteria at the end of the algorithm 2.15 presented below.

**ALGORITHM 2.15: 1D-SSA.** The angle-regularity identification method for the oscillatory component

*Input:* The following data and parameters are input.

1. **Data:** number of components  $r$ ; left singular vectors  $\{U_j\}_{j=1}^r$ .
2. **Parameters:** number of e-m harmonics  $m \leq r/2$ , or threshold  $t_0 \geq 0$ , depending on the stopping criterion.

*Output:* A group of  $J$  singular vector indices related to the oscillatory component.

- 1: Based on  $\{U_i\}_{i=1}^r$ , compute and order in ascending order the values  $\tilde{\tau}(U_j, U_{j+1})$ ,  $j = 1, \dots, r-1$  using (2.27). While calculating for each  $j = 2, \dots, r-1$  we check: if  $\tilde{\tau}(U_j, U_{j+1}) < \tilde{\tau}(U_{j-1}, U_j)$ , then we drop  $\tilde{\tau}(U_{j-1}, U_j)$  from consideration; otherwise we drop  $\tilde{\tau}(U_j, U_{j+1})$ . The resulting values are  $\tau_1, \dots, \tau_{[r/2]}$ .
- 2: Two versions of the stopping criterion are as follows. Try the elements of the set  $i = 1, \dots, [r/2]$ 
  1. either until  $i < m$ ,
  2. or until  $\tau_i < t_0$ .

Denote  $i_0$  the moment of stopping.

- 3: The set  $J$  consists of the indices  $j, j+1$  of the singular vectors  $U_j$  and  $U_{j+1}$  involved in computing the values  $\tau_1, \dots, \tau_{i_0-1}$ .

**Remark 21.** Not only the  $r$  leading singular vectors can be the input of the algorithm. Any set of consecutive vectors (a set where the index of each next vector is one more than the previous one) from the set  $\{U_j\}_{j=1}^d$  can be the input. In the algorithm, the variant with the  $r$  leading vectors was given for simplicity.

**Remark 22.** The question about the choice of normalization in the definition of (2.27) needs further elaboration.

### Choice of the threshold

There are two options for using methods of automatic identification.

1. Analyze one time series without information in advance.
2. Analyze many time series of similar structures (batch processing).

For the second option, the method of selecting the threshold is straightforward. For several series from the batch, we compute the values of the measure  $\tau$  for consecutive pairs of singular vectors; look at the images of singular vectors, choose a group of indices  $I$  of the components that belong to the oscillatory component; choose a threshold  $t_0$  at some point, between the maximum value  $\tau$  for  $I$  and the minimum value  $\tau$  for the remaining indices  $\{1, \dots, d\} \notin I$ . Then we use the obtained value of the threshold  $t_0$  for all remaining series. In the 2.5.4 section, the method under consideration will be compared to the frequency method just for this case.

For the first option, it is not possible to invent and theoretically justify a universal method for selecting the threshold. Therefore, the problem of selecting a threshold that leads to extracting only the oscillatory component is replaced by a preprocessing that finds all the oscillatory components, but can also extract something else.

Therefore, let us run simulations of series with a single e-m harmonic and noise; then look at the 95% quantile of the value  $\tau(U_1, U_2)$  for different noise levels and the dependence of quantiles on noise levels.

Hereafter, we consider the following S, N, and X series for model data studies:

$$s_k = e^{\alpha k} \cos(2\pi k \omega), \quad (2.28)$$

$$n_k = e^{\alpha k} \sigma \varepsilon_k, \quad (2.29)$$

$$x_k = s_k + n_k, \quad (2.30)$$

where  $k = 1, \dots, N$ ,  $\varepsilon_k$  are independent Gaussian random variables distributed as  $N(0, 1)$ . Consider  $N = 99$ ,  $L = 50$ ,  $\sigma = 0, 0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4$ . The number of simulations is 1000.

For each value of  $\sigma$ , simulate the series N 1000 times and count the 95% quantile of the sample value of the measure  $\tau(U_1, U_2)$  from the first two singular vectors based on the obtained sample series in the form (2.30). Figure 2.26 shows a graph of the dependence of the quantile on the value of  $\sigma$ .

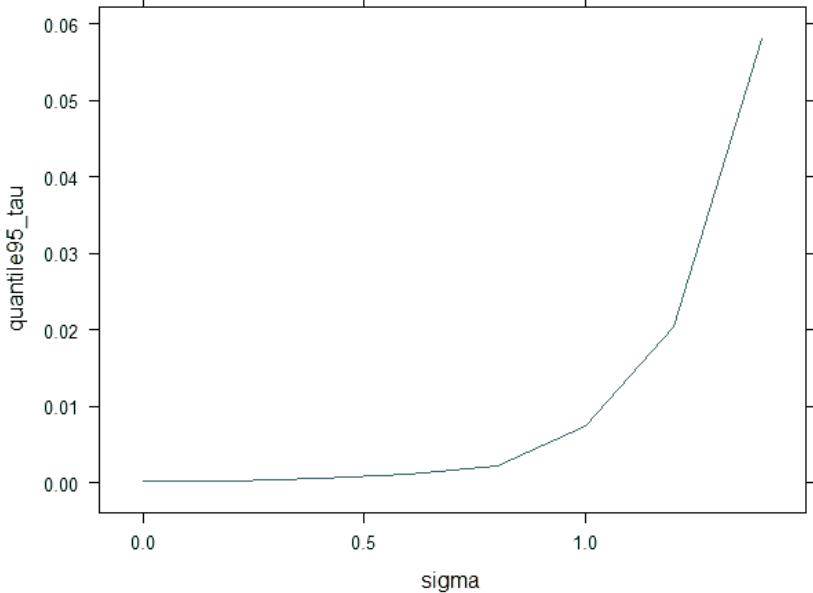


Figure 2.26: Dependence of the 95% quantile for  $\tau(U_1, U_2)$  on the  $\sigma$  value of noise.

The value  $\sigma = 1$  is very large for the length of the series under investigation, i.e., the noise is mixed with the oscillatory component. This can be seen in Fig. 2.26: after  $\sigma = 1$  the graph went sharply upwards. So it is worth considering the results for  $\sigma \leq 1$ . The results show that the approximate appropriate value of the threshold is  $t_0 = 0.01$  for  $\sigma = 1$ .

#### 2.5.4. Comparison of methods for harmonic identification

Let us compare the method by the regularity of angles with the frequency method from Section 2.5.2. As before, we consider the series  $S$ ,  $N$  and  $X$  with elements given by (2.28), (2.29) and (2.30).

The aim of automatic identification is the same as that of visual identification. Let us therefore compare the methods in the following way. We will simulate the e-m harmonic  $S$  and noise  $N$  with  $\sigma = 0.2, 0.4, 0.6, 0.8, 1$ ,  $\alpha = 0, 0.01$ ,  $\omega = 1/7$ . As before,  $N = 99$  and  $L = 50$  (therefore,  $L\omega$  is not integer). For fixed values of parameters  $\alpha$ ,  $\omega$ ,  $\sigma$  we will model the series  $X = S + N$ . We will call “visual identification” the signal reconstruction by two leading eigentriples.

Let  $X^{(1)}$  and  $X^{(2)}$  be two realizations of the series  $X$ . Let  $S^{(V,1)}$  and  $S^{(V,2)}$  be the series reconstructed from the series  $X^{(1)}$  and  $X^{(2)}$  by means of  $\{\}$ visual identification $\}$ .

Denote the threshold value for the method with  $\tau$  by  $t_0$ , and for the frequency method by  $\rho_0$ . Let  $S^{(A,1,\varepsilon)}$  and  $S^{(A,1,\rho_0)}$  be the series recovered from the series  $X^{(1)}$  using automatic identification algorithms with threshold values  $t_0$  and  $\rho_0$  respectively. We also denote the set of values  $T = \{0, 0.01, 0.02, \dots, 0.99, 1\}$ , i.e., numbers from 0 to 1 in steps of 0.01.

Then we find the values of thresholds  $t_0^{opt}$  and  $\rho_0^{opt}$  by solving the following minimization

problems:

$$t_0^{opt} = \arg \min_{t_0 \in T} \left( \frac{1}{N} \sum_{k=0}^{N-1} \left( s_k^{(V,1)} - s_k^{(A,1,t_0)} \right)^2 \right),$$

$$\rho_0^{opt} = \arg \min_{\rho_0 \in T} \left( \frac{1}{N} \sum_{k=0}^{N-1} \left( s_k^{(V,1)} - s_k^{(A,1,\rho_0)} \right)^2 \right).$$

Next, let  $S^{(A,2,t_0^{opt})}$  and  $S^{(A,2,\rho_0^{opt})}$  be the series reconstructed from the series  $X^{(2)}$  using the automatic identification algorithms with threshold values  $t_0^{opt}$  and  $\rho_0^{opt}$  respectively.

Let us calculate the identification errors for these series as

$$E_\tau = \frac{1}{N} \sum_{k=0}^{N-1} \left( s_k^{(V,2)} - s_k^{(A,2,t_0^{opt})} \right)^2, \quad E_\rho = \frac{1}{N} \sum_{k=0}^{N-1} \left( s_k^{(V,2)} - s_k^{(A,2,\rho_0^{opt})} \right)^2.$$

Thus, we calculated the optimal threshold values for one realization of the  $X$  series and used these optimal threshold values to reconstruct the series over the second realization, and calculated the identification error for the resulting series.

Let us repeat the above procedure 200 times and calculate the mean and median of the identification errors  $E_\tau$  and  $E_\rho$ . The results for  $\alpha = 0$ ,  $\omega = 1/7$  are shown in Table 2.1. The results for  $\alpha = 0.02$ ,  $\omega = 1/7$  are presented in Table 2.2.

Table 2.1: Comparison of harmonic identification;  $\alpha = 0$ ,  $\omega = 1/7$ .

	mean_τ	mean_ρ	median_τ	median_ρ
$\sigma = 0.2$	0.0042	0.040	0	0.0028
$\sigma = 0.4$	0.0156	0.064	0	0.0046
$\sigma = 0.6$	0.0444	0.100	0	0.0342
$\sigma = 0.8$	0.1012	0.151	0	0.0656
$\sigma = 1$	0.1224	0.183	0	0.1179

Table 2.2: Comparison of harmonic identification;  $\alpha = 0.02$ ,  $\omega = 1/7$ .

	mean_τ	mean_ρ	median_τ	median_ρ
$\sigma = 0.2$	0.0003	0.050	0	0.0010
$\sigma = 0.4$	0.0010	0.080	0	0.0020
$\sigma = 0.6$	0.0082	0.103	0	0.0170
$\sigma = 0.8$	0.0549	0.185	0	0.0655
$\sigma = 1$	0.1917	0.239	0	0.1554

The following conclusions can be drawn from tables 2.1 and 2.2. First, for all values of  $\alpha$  and  $\sigma$ , the angle regularity identification method with the  $\tau$  measure gives smaller errors than the frequency method with the  $\rho$  measure. Second, both methods give approximately the same results for unmodulated ( $\alpha = 0$ ) and modulated harmonics ( $\alpha = 0.02$ ). For the method of identification by the regularity of angles, this result is confirmed by theory, since the measure  $\tau$  does not depend

on  $\alpha$ . For the frequency method, this result is unexpected, since the method is based on the periodogram of singular vectors, and for the modulated harmonic there is a “leakage” of the periodogram. The result can be explained by the fact that, for the frequency method, the algorithm takes into account possible leakage, and as the key measure  $\rho$  given by the formula (2.23), the sum of two neighboring values of the periodogram is taken.

## Chapter 3

# Model-based problems: gap filling, signal estimation, signal detection

The SSA analysis of time series, which is considered in Chapter 2, can be classified as model-free. In this chapter, on the contrary, we consider the methodologies within the 1D-SSA approach, which require a model. These methodologies include the common problems of forecasting, interpolation, low-rank approximation and parameter estimation. The model used is based on properties of the approximating subspace constructed in the process of 1D-SSA analysis of Chapter 2 and so the methodologies of this chapter belong to the class of subspace-based methods of time series analysis and signal processing.

As in Chapter 2, for the sake of brevity, in this chapter we will refer to 1D-SSA simply as SSA. In contrast to the SSA analysis, the input for the algorithms of this chapter is not necessarily a collection  $X_N = (x_1, \dots, x_N)$  of  $N$  real numbers; it can be an estimated subspace. This subspace is, as a rule, obtained after Grouping step of any of the SSA algorithms and has the form  $\text{span}(U_i, i \in I)$ .

Section 3.1 is devoted to the problem of filling the missing data in time series.

Section 3.2 is related to the problem of estimation of signals in noisy time series if the signal is low-rank and the noise is autoregressive. There are two forms of the statement of the problem and therefore two approaches to its solution. We consider a particular problem of equivalence of the forms.

In Section 3.3 we consider the problem of signal detection, where the signal is sine-wave and noise is red (i.e. an autoregressive process of order 1 with positive coefficient).

The chapter also contains Section 3.4 devoted to the choice of parameters in methods described in this and previous chapters.

### **3.1. Subspace-based method of SSA gap filling**

#### **3.1.1. Overview of gap filling methods in SSA**

This section is devoted to the extension of the SSA forecasting algorithms for the analysis of time series with missing data.

There are three approaches to solving this problem. The first approach was suggested in [132]. This approach is suitable for stationary time series only and uses the following simple idea: in the process of the calculation of the inner products of vectors with missing components, we use only pairs of valid vector components and omit the others. The RSSA package does not implement this approach because of its limitations. We hence concentrate on the other two approaches, the subspace-based approach [4] and the iterative one [133].

Usually, the problem of missing data imputation is stated as the problem of filling-in the signal

data. However, the problem of imputation is more general. For example, one can be interested in the imputation of missing data in the trend or seasonality only. To do it, the structure, which we are interested in, should be detected by the method. From the viewpoint of SSA, it means that the interesting series component should be separated from the residual and also the rule for the component extraction should be fixed (e.g., the indices of the eigentriples for reconstruction should be set in Basic SSA). Therefore, it makes sense to combine the considered methods with the SSA modifications described in Chapter 2 that improve separability.

For detection of the structure before performing gap filling, Shaped SSA can be applied to the series if the location of the gaps allows the decomposition (see Section 5.2.1). If Shaped SSA gives unsatisfactory results (for example, if the number of complete lagged vectors is too small and therefore detection of the structure is impossible), then the subspace-based approach is not applicable. However, the following general technique can be applied in the framework of the iterative approach: artificial gaps can be added and parameters of the method of gap filling can be chosen to minimize the error of imputation.

## Subspace-based approach

The subspace-based method of gap filling suggested in [4] (see also [29, Section 3.7] and [30]) is an extension of SSA forecasting algorithms. We will describe the subspace-based method according to these references.

For forecasting, the last vector coordinate in a chosen subspace can be uniquely imputed as a linear combination of the first  $L - 1$  coordinates. The approach can be extended for imputing a set of unknown (missing) vector coordinates as linear combinations of known coordinates. Here we use a found signal structure (in the form of a subspace) to fill the gaps. In a particular case, when missing values are located at the end of the series, the problem of filling-in of these values coincides with the problem of forecasting.

The assumptions for the gap filling are the same as for forecasting; that is, SSA should be able to approximately separate the series component of interest.

Note that imputation of gaps in separate signal components can be performed as the following two-step procedure: first, we fill gaps in the whole signal and then decompose the reconstructed signal into desired components.

**Clusters of missing data** In the subspace-based approach, the gap filling method can be applied to different groups of missing data independently. To introduce such independent clusters, let us give several definitions following.

**Definition 10.** For a fixed  $L$ , a sequence of missing data of a time series is called a *cluster of missing data* if every two adjacent missing values from this sequence are separated by less than  $L$  non-missing values and there is no missing data among  $L$  neighbours (if they exist) of the left/right element of the cluster.

Thus, a group of not less than  $L$  successive non-missing values of the series separates clusters of missing data.

A cluster is called *left/right* if its left/right element is located at a distance of less than  $L$  from the left/right end of the series. A cluster is called *continuous* if it does not contain non-missing data.

**The layout of the algorithm** Let us describe the algorithm layout in the form that is implemented in the RSSA package.

Assume that we have the initial time series  $\mathbf{X}_N = (x_1, \dots, x_N)$  consisting of  $N$  elements, some part of which is unknown. Let us describe the scheme of the algorithm assuming that we are reconstructing the first component  $\mathbf{X}_N^{(1)}$  of the observed series  $\mathbf{X}_N = \mathbf{X}_N^{(1)} + \mathbf{X}_N^{(2)}$ .

The scheme of the method is as follows. Parameters are the window length  $L$  and a group  $I$  of components in the SSA decomposition. We assume that the location of missing data allows application of Shaped SSA for the chosen  $L$ . Two versions, ‘sequential’ and ‘simultaneous’ are suggested. These versions correspond to sequential recurrent forecasting and simultaneous vector forecasting respectively.

### Scheme of subspace-based gap filling

1. **Shaped SSA.** For the series, the shaped version of SSA is applied for the given window length  $L$  and group  $I$ . Any modification described in Chapter 2 and consistent with Shaped SSA can be used. As a result, we obtain a reconstructed series and a set of orthonormal vectors providing a basis for the approximated signal subspace.
2. **Detection of clusters of missing data.** All missing entries are split into clusters. For sequential version, each cluster is transformed into a continuous one; that is, non-missing values within the cluster are changed to NA.
3. **Forecasting.** For forecasting or filling-in several values, two approaches can be used, sequential and simultaneous. Both of these approaches can use either the recurrent or vector forecasting methods. In the current version of the RSSA package, the sequential approach uses the recurrent forecasting method, while the simultaneous approach uses a method similar to the vector forecasting one.

Sequential gap filling-in makes forecasting from the left and from the right for each cluster with subsequent weighted averaging of the forecasting results; the subspace used for forecasting is estimated by Shaped SSA. If a cluster is left or right, then only one forecast is used for gap-filling.

For simultaneous gap filling the so-called simultaneous forecasting is used. In [29, Section 3.1], simultaneous forecasting is described in its recurrent version. Since here we use the

vector version, the simultaneous gap filling consists of two operations called ‘ $\Pi$ -projection’ and ‘simultaneous filling-in’.

**Discussion** Note that for successful imputation, an approximate separability of the imputed component is necessary. For exactly separated component, the missing values can be reconstructed with no error. The location of missing data is very important for the possibility of imputation by the subspace method, since the number of non-missing values should be large enough for achieving separability by Shaped SSA. At least, the number of the complete lagged vectors should be larger than the rank of the imputed time series component.

**Algorithms** Below we only provide a short description of the algorithms of subspace-based filling-in. The algorithms can deal with several gaps. We only describe their versions for one internal gap.

The following algorithm corresponds to a combination of methods ‘sequential filling-in from the left’ and ‘sequential filling-in from the right’. In the gap-filling algorithms, we will use the term Shaped SSA, which will be introduced later in Chapter 5; however, when applied to a time series with gaps, Shaped SSA differs from regular SSA only in that the trajectory matrix is not constructed from all embedding vectors, but only from those with no gaps.

#### ALGORITHM 3.1: Sequential recurrent subspace-based gap filling

*Input:* Time series  $X$  of length  $N$  containing a gap, which starts from  $i$ th and finished in  $j$ th points, set of gap indices  $P = \{i, \dots, j\}$ ,  $p = |P|$ , window length  $L$ , version of SSA, group of eigentriples  $I$ .

*Output:* Reconstructed series component  $\tilde{X}$  with a filled gap.

- 1: Apply the shaped form of the chosen SSA version to  $\tilde{X}$  (Section 5.2.1) and obtain the subspace  $\mathcal{L} = \text{span}\{P_i, i \in I\}$  and the reconstructed series  $\tilde{X}$  with gaps.
- 2: Apply the forward recurrent forecasting algorithm in the subspace  $\mathcal{L}$  starting from  $(\tilde{x}_{i-L+1}, \dots, \tilde{x}_{i-1})$  and construct the  $p$ -step recurrent forecast  $G^{\text{left}}$ .
- 3: Apply the backward recurrent forecasting algorithm in the subspace  $\mathcal{L}$  starting from  $(\tilde{x}_{j+L-1}, \dots, \tilde{x}_{j+1})$  and construct the  $p$ -step recurrent forecast  $G^{\text{right}}$ .
- 4: Combine  $G^{\text{left}}$  and  $G^{\text{right}}$  to obtain  $G$ . For example,  $g_i = (1 - \alpha_i)g_i^{\text{left}} + \alpha_i g_i^{\text{right}}$ ,  $i = 1, \dots, p$ , where  $\alpha_i = i/(p+1)$ .
- 5: Set  $\tilde{X}|_P = G$ .

Note that if the gap is right (or left), then only forward (or backward) recurrent forecasting is applied. The algorithm 3.1 does not require a particular justification, since it is based on the SSA forecasting algorithm.

The following algorithm corresponds to the combination of the method ‘ $\Pi$ -projector’ and ‘simultaneous filling-in’.

**ALGORITHM 3.2: Simultaneous vector subspace-based gap filling**

*Input:* Time series  $X$  of length  $N$  containing a gap, which starts from  $i$ th and finishes in  $j$ th points, set of gap indices  $P = \{i, \dots, j\}$ ,  $p = |P|$ , window length  $L$ , version of SSA, group of eigentriples  $I$ .

*Output:* Reconstructed series component  $\tilde{X}$  with a filled gap.

- 1: Apply the shaped form of the chosen SSA version to  $\tilde{X}$  (Section 5.2.1), and obtain the subspace  $\mathcal{L} = \text{span}\{P_i, i \in I\}$  and the reconstructed matrix  $\hat{X}$  consisting of vectors with non-missing values at all positions.
- 2: Continue the values of the complete reconstructed vectors according to Hankel structure of the matrix to obtain partly filled reconstructed vectors.
- 3: Project the valid parts of vectors by means of the  $\Pi$ -projector.
- 4: Simultaneously fill-in the missing parts of the vectors. If it is impossible, then put `NA` (‘not available’).
- 5: Hankelize the matrix  $\hat{X}$  to obtain the series  $\tilde{X}$ . Hankelization is performed by averaging by non-missing values. If there are no non-missing values, then the result is `NA`.

The theoretical justification of Algorithm 3.2 is contained in Sections 3.1.2–3.1.4.

### Iterative approach

A natural and simple idea for filling-in missing values is the iterative approach, when the missing entries are initially filled-in using some reasonable values and then these values are iteratively improved by updating the SSA approximations for underlying structure of the object. This idea was suggested in [134] for the imputation of missing values in noisy rank-deficient matrices and was later extended to time series in [133].

For a rank-deficient matrix, the structure is defined by its rank and therefore the improvement is performed by the SVD, where the first  $r$  SVD components describe this structure. For time series of finite rank  $r$ , the improvement can be obtained with the help of the SVD of the trajectory matrix with subsequent hankelization. Note that this is exactly the Basic SSA algorithm with reconstruction. Also, Toeplitz SSA or SSA with projection can be used at iterations, if the series is stationary or we partly know the series model.

At each iteration, we insert the improved values at the places of missing entries and restore the initial data at the places of non-missing entries.

The approach described above can be formally applied for almost any location of missing values. Numerical experiments shows that the iterative approach can fail if missing data are located

at the ends of the time series.

The iterative approach has no rigorous proof of convergence. Another drawback of the iterative approach is its impossibility to fill-in the gaps exactly even for noiseless signals. Moreover, the iterative method has large computational cost.

Let us write down the algorithm of the iterative gap-filling algorithm. For a collection  $\mathbf{Y}$  and a set of indices  $P$  we denote by  $\mathbf{Y}|_P$  the part of the collection with the indices from  $P$ . Set  $\mathcal{N} = \{1, \dots, N\}$ .

### ALGORITHM 3.3: Iterative gap filling

*Input:* Time series  $\mathbf{X}$  of length  $N$  containing gaps, set of indices of missing values  $P$ , window length  $L$ , version of SSA, series  $\mathbf{G}$  of length  $N$  as the source of initial values for gaps, rank for reconstruction  $r$ , stop criterion STOP.

*Output:* Reconstructed series component  $\tilde{\mathbf{X}}$  with no gaps.

- 1:  $k \leftarrow 0$ ,  $\tilde{\mathbf{G}}^{(k)}|_P = \mathbf{G}|_P$ ,  $I = \{1, \dots, r\}$ .
- 2: Set  $\tilde{\mathbf{X}}^{(k+1)}$  such that  $\tilde{\mathbf{X}}^{(k+1)}|_{\mathcal{N} \setminus P} = \mathbf{X}|_{\mathcal{N} \setminus P}$  and  $\tilde{\mathbf{X}}^{(k+1)}|_P = \tilde{\mathbf{G}}^{(k)}|_P$ .
- 3: Apply the selected version of SSA with the chosen  $L$  and  $I$  to  $\tilde{\mathbf{X}}^{(k+1)}$  and obtain the reconstructed series  $\mathbf{G}^{(k+1)}$ .
- 4:  $k \leftarrow k + 1$
- 5: If not STOP, go to Step 2; else  $\tilde{\mathbf{X}} = \mathbf{G}^{(k)}$ .

Input of the algorithm can contain several groups of indices  $I_k$ ,  $k = 1, \dots, m$ . Then the iterations are performed for  $r = \max\{i : i \in I_k, k = 1, \dots, m\}$ . In this case, the reconstruction at the last step before STOP is performed for each group  $I_k$  separately.

#### 3.1.2. Preliminary results

Let us start the theoretical justification of Algorithm 3.2 with several auxiliary propositions.

#### Recovery of vector's components in a subspace

Let us give necessary notation. Define  $\mathcal{I} = \{1, \dots, n\}$  and denote by  $\mathcal{S} = \{i_1, \dots, i_s\} \subset \mathcal{I}$  an ordered set,  $|\mathcal{S}| = s$ . Let  $\mathbf{I}_s$  denote the unit  $s \times s$  matrix.

By definition, *restriction of a vector*  $\mathbf{X} = (x_1, \dots, x_n)^T \in \mathbb{R}^n$  onto a set of indices  $\mathcal{S}$  is the vector  $\mathbf{X}|_{\mathcal{S}} = (x_{i_1}, \dots, x_{i_{|\mathcal{S}|}})^T \in \mathbb{R}^{|\mathcal{S}|}$ .

*Restriction of a matrix* onto a set of indices is the matrix consisting of restrictions of its column vectors onto this set.

*Restriction of a  $q$ -dimensional subspace*  $\mathfrak{G}_q$  onto a set of indices  $\mathcal{S}$  is the subspace spanned by restrictions of all vectors of  $\mathfrak{G}_q$  onto this set; the restricted subspace will be denoted by  $\mathfrak{G}_q|_{\mathcal{S}}$ . It is

easy to prove that for any basis  $\{H_i\}_{i=1}^q$  of the subspace  $\mathfrak{G}_q$  the equality  $\mathfrak{G}_q|_{\mathcal{S}} = \text{span}(H_1|_{\mathcal{S}}, \dots, H_q|_{\mathcal{S}})$  holds.

Consider an  $m$ -dimensional subspace  $\mathfrak{D}_m \in \mathbb{R}^n$ . Denote by  $\{R_k\}_{k=1}^m$  an orthonormal basis of the  $\mathfrak{D}_m$  and define the matrix  $\mathbf{R} = [R_1 : \dots : R_m]$ . Fix an ordered set of indices  $\mathcal{P}$ .

**Proposition 14.** *Let the matrix  $\mathbf{I}_{|\mathcal{P}|} - \mathbf{R}|_{\mathcal{P}}(\mathbf{R}|_{\mathcal{P}})^T$  be non-singular. Then for any vector  $X \in \mathfrak{D}_m$  the following formula expressing  $X|_{\mathcal{P}}$  in terms of  $X|_{\mathcal{J} \setminus \mathcal{P}}$  holds:*

$$X|_{\mathcal{P}} = (\mathbf{I}_{|\mathcal{P}|} - \mathbf{R}|_{\mathcal{P}}(\mathbf{R}|_{\mathcal{P}})^T)^{-1} \mathbf{R}|_{\mathcal{P}}(\mathbf{R}|_{\mathcal{J} \setminus \mathcal{P}})^T X|_{\mathcal{J} \setminus \mathcal{P}}. \quad (3.1)$$

*Proof.* For simplicity of notation, let  $\mathcal{P} = \{1, \dots, |\mathcal{P}|\}$ . Denote  $X_1 = X|_{\mathcal{P}}$ ,  $X_2 = X|_{\mathcal{J} \setminus \mathcal{P}}$ ,  $\mathbf{R}_1 = \mathbf{R}|_{\mathcal{P}}$ ,  $\mathbf{R}_2 = \mathbf{R}|_{\mathcal{J} \setminus \mathcal{P}}$ . Since  $\mathbf{R}\mathbf{R}^T X = X$  for  $X \in \mathfrak{D}_m$  and

$$\mathbf{R}\mathbf{R}^T = \begin{pmatrix} \mathbf{R}_1\mathbf{R}_1^T & \mathbf{R}_1\mathbf{R}_2^T \\ \mathbf{R}_2\mathbf{R}_1^T & \mathbf{R}_2\mathbf{R}_2^T \end{pmatrix},$$

we have  $X_1 = \mathbf{R}_1\mathbf{R}_1^T X_1 + \mathbf{R}_1\mathbf{R}_2^T X_2$ . Turning back to the original notation, we come to the equality

$$(\mathbf{I}_{|\mathcal{P}|} - \mathbf{R}|_{\mathcal{P}}(\mathbf{R}|_{\mathcal{P}})^T)X|_{\mathcal{P}} = \mathbf{R}|_{\mathcal{P}}(\mathbf{R}|_{\mathcal{J} \setminus \mathcal{P}})^T X|_{\mathcal{J} \setminus \mathcal{P}}.$$

This completes the proof.  $\square$

**Lemma 9.** *The following conditions are equivalent:*

- 1)  $\dim \mathfrak{D}_m|_{\mathcal{J} \setminus \mathcal{P}} = \dim \mathfrak{D}_m$ ;
- 2) the vectors  $\{R_k|_{\mathcal{J} \setminus \mathcal{P}}\}_{k=1}^m$  are linearly independent and form a basis of the subspace  $\mathfrak{D}_m|_{\mathcal{J} \setminus \mathcal{P}}$ ;
- 3)  $Y|_{\mathcal{J} \setminus \mathcal{P}} \neq 0_{|\mathcal{J} \setminus \mathcal{P}|}$  for any nonzero vector  $Y \in \mathfrak{D}_m$ ;
- 4)  $\text{span}(\mathbf{e}_i, i \in \mathcal{P}) \cap \mathfrak{D}_m = \{0_n\}$ ;
- 5) the matrix  $(\mathbf{I}_{|\mathcal{P}|} - \mathbf{R}|_{\mathcal{P}}(\mathbf{R}|_{\mathcal{P}})^T)^{-1}$  exists.

*Proof.* The equivalence of the conditions 1) – 4) is evident.

Consider the equivalence of 4) and 5). Proposition 14 implies the equivalence of 5) to the following assertion: for any vector  $V \in \mathfrak{D}_m|_{\mathcal{J} \setminus \mathcal{P}}$  there exists an unique vector  $G \in \mathfrak{D}_m$  such that  $V = G|_{\mathcal{J} \setminus \mathcal{P}}$ . Let us prove the equivalence of 4) to the same assertion.

Since the vectors  $\{R_k|_{\mathcal{J} \setminus \mathcal{P}}\}_{k=1}^m$  span the subspace  $\mathfrak{D}_m|_{\mathcal{J} \setminus \mathcal{P}}$ , the vector  $V$  can be expressed as  $V = \sum_{k=1}^m a_k R_k|_{\mathcal{J} \setminus \mathcal{P}}$ . Then the required vector is  $G = \sum_{k=1}^m a_k R_k$ . Suppose that there are two different vectors  $G_1$  and  $G_2$  such that  $V = G_1|_{\mathcal{J} \setminus \mathcal{P}} = G_2|_{\mathcal{J} \setminus \mathcal{P}}$ . Consider their difference  $G_1 - G_2 = \sum_{i \in \mathcal{P}} \alpha_i \mathbf{e}_i \in \mathfrak{D}_m$ . This difference is not equal to the zero vector if and only if  $\text{span}(\mathbf{e}_i, i \in \mathcal{P}) \cap \mathfrak{D}_m = \{0_n\}$ . The lemma is proved.  $\square$

**Remark 23.** It follows from the item 4) of Lemma 9 that  $n - m \geq |\mathcal{P}|$ . This constraint on the number of vector's missing components is a necessary condition for applying the formula (3.1).

Let us consider two special cases, when the first ( $\mathcal{P} = \{1\}$ ) or the last ( $\mathcal{P} = \{n\}$ ) coordinate is expressed through the rest.

**Corollary 6.** Denote  $v^2 = \pi_1^2 + \dots + \pi_m^2$ , where  $\pi_i$  is the  $n$ -th component of the vector  $R_i$ , and  $\{\underline{R}_i\}_{i=1}^m$  are the vectors  $\{R_i\}_{i=1}^m$  without the last components (their dimension is equal to  $n-1$ ). Suppose that  $\mathbf{e}_n \notin \mathfrak{D}_m$  and  $X = (x_1, \dots, x_n)^T \in \mathfrak{D}_m$ . Then  $v^2 < 1$  and  $x_n = \sum_{k=1}^{n-1} a_k x_{n-k}$ , where

$$(a_{n-1}, \dots, a_1)^T = \frac{1}{1-v^2} \sum_{i=1}^m \pi_i \underline{R}_i. \quad (3.2)$$

**Corollary 7.** Denote  $\mu^2 = \rho_1^2 + \dots + \rho_m^2$ , where  $\rho_i$  is the first coordinate of the vector  $R_i$ , and  $\{\overline{R}_i\}_{i=1}^m$  are the vectors  $\{R_i\}_{i=1}^m$  without the first components (their dimension is equal to  $n-1$ ). Suppose that  $\mathbf{e}_1 \notin \mathfrak{D}_m$  and  $X = (x_1, \dots, x_n)^T \in \mathfrak{D}_m$ . Then  $\mu^2 < 1$  and  $x_1 = \sum_{k=2}^n a_k x_k$ , where

$$(a_2, \dots, a_n)^T = \frac{1}{1-\mu^2} \sum_{i=1}^m \rho_i \overline{R}_i. \quad (3.3)$$

Formula (3.2) is exactly the formula that set the coefficients for the recurrent SSA-forecasting.

## Projection operator

Consider the subspaces  $\mathcal{L}^{(1)}$  and  $\mathcal{L}^{(2)}$  of  $\mathbb{R}^n$ , of dimensions  $m$  and  $\tilde{m} \leq n-m$  correspondingly. Let us first find the matrix of the operator  $\Pi_{\mathcal{J} \setminus \mathcal{P}}^{(1)}$  corresponding to the orthogonal projection  $\mathbb{R}^{|\mathcal{J} \setminus \mathcal{P}|} \rightarrow \mathcal{L}^{(1)}|_{\mathcal{J} \setminus \mathcal{P}}$ . Denote by  $\{R_k\}_{k=1}^m$  an orthonormal basis of the space  $\mathcal{L}^{(1)}$ ,  $\mathbf{R} = [R_1 : \dots : R_m]$ . Set  $\mathbf{V} = \mathbf{R}|_{\mathcal{J} \setminus \mathcal{P}}$  and  $\mathbf{W} = \mathbf{R}|_{\mathcal{P}}$  for convenience of notation.

**Proposition 15.** Assume that the matrix  $\mathbf{I}_{|\mathcal{P}|} - \mathbf{W}\mathbf{W}^T$  is nonsingular. Then the matrix  $\Pi_{\mathcal{J} \setminus \mathcal{P}}^{(1)}$  of the orthogonal projection operator  $\Pi_{\mathcal{J} \setminus \mathcal{P}}^{(1)}$  has the form

$$\Pi_{\mathcal{J} \setminus \mathcal{P}}^{(1)} = \mathbf{V}\mathbf{V}^T + \mathbf{V}\mathbf{W}^T(\mathbf{I}_{|\mathcal{P}|} - \mathbf{W}\mathbf{W}^T)^{-1}\mathbf{W}\mathbf{V}^T. \quad (3.4)$$

*Proof.* Introduce the matrix  $\mathbf{A} = \mathbf{V}^T \mathbf{V}$  ( $\mathbf{A}$  and  $\mathbf{W}^T \mathbf{W}$  are  $m \times m$  matrices). According to Lemma 9, the matrix  $\mathbf{A}$  is nonsingular as a Gram matrix of a linearly independent set of vectors; therefore  $\mathbf{A}$  is reversible. It is known that in this case the operator of orthogonal projection onto the space spanned by the columns of the matrix  $\mathbf{V}$  has the form  $\Pi_{\mathcal{J} \setminus \mathcal{P}}^{(1)} = \mathbf{V}\mathbf{A}^{-1}\mathbf{V}^T$ . The challenge is to find the explicit formula of the matrix  $\mathbf{A}^{-1}$ .

Since the vectors  $\{R_k\}_{k=1}^m$  constitute an orthonormal basis of the space  $\mathcal{L}^{(1)}$ , we have  $\mathbf{R}^T \mathbf{R} = \mathbf{I}_m$ . On the other hand,

$$\mathbf{R}^T \mathbf{R} = (\mathbf{R}|_{\mathcal{J} \setminus \mathcal{P}})^T \mathbf{R}|_{\mathcal{J} \setminus \mathcal{P}} + (\mathbf{R}|_{\mathcal{P}})^T \mathbf{R}|_{\mathcal{P}} = \mathbf{A} + \mathbf{W}^T \mathbf{W}.$$

Thus,  $\mathbf{A} = \mathbf{I}_m - \mathbf{W}^T \mathbf{W}$ . Straightforward demonstration of

$$(\mathbf{I}_m - \mathbf{W}^T \mathbf{W})(\mathbf{I}_m + \mathbf{W}^T (\mathbf{I}_{|\mathcal{P}|} - \mathbf{W} \mathbf{W}^T)^{-1} \mathbf{W}) = \mathbf{I}_m$$

completes the proof.  $\square$

Denote by  $\boldsymbol{\Pi}^{(1)}$  the operator of orthogonal projection of  $\mathbb{R}^n$  onto  $\mathcal{L}^{(1)}$ . The evident conditions of permutability of the projection and the restriction procedures are given in the following proposition.

**Proposition 16.** *If  $\mathcal{L}^{(1)} \perp \mathcal{L}^{(2)}$  and  $\mathcal{L}^{(1)}|_{\mathcal{J} \setminus \mathcal{P}} \perp \mathcal{L}^{(2)}|_{\mathcal{J} \setminus \mathcal{P}}$ , then*

$$\boldsymbol{\Pi}_{\mathcal{J} \setminus \mathcal{P}}^{(1)}(X|_{\mathcal{J} \setminus \mathcal{P}}) = (\boldsymbol{\Pi}^{(1)} X)|_{\mathcal{J} \setminus \mathcal{P}}$$

for any  $X \in \mathcal{L}^{(1)} \oplus \mathcal{L}^{(2)}$ .

### 3.1.3. Lagged vectors and trajectory spaces of time series of finite rank with missing data

#### Recovery of missing components of lagged vectors

Consider a time series  $\mathbf{X}_N$  of finite rank  $d$  and fix a window length  $L$ ,  $d < \min(L, K)$ ,  $K = N - L + 1$ . Suppose that some data are missing in the observed series  $\mathbf{X}_N$  (i.e. some data are considered to be unknown), but its  $L$ -trajectory space  $\mathcal{L}_d \subset \mathbb{R}^L$  is known.

Let us take an incomplete  $L$ -lagged vector, which contains both missing and non-missing data of  $\mathbf{X}_N$ . Denote this vector by  $X$  and the ordered set of its missing components' indices by  $\mathcal{P}$ . From now on,  $\mathcal{J}$  stands for the set  $\{1, \dots, L\}$  and the set of indices  $\mathcal{J} \setminus \mathcal{P}$  indicates the set of non-missing components of the vector  $X$ . The restrictions  $X|_{\mathcal{P}}$  and  $X|_{\mathcal{J} \setminus \mathcal{P}}$  are thereby the vectors, which consist of missing and non-missing components of the vector  $X$  correspondingly.

To solve the problem of reconstructing the missing components  $X|_{\mathcal{P}}$  of the vector  $X$  by means of the non-missing components  $X|_{\mathcal{J} \setminus \mathcal{P}}$ , we apply the theory of Subsection 3.1.2 (with  $m = d$ ,  $\mathfrak{D}_m = \mathcal{L}_d$ ) to the vector  $X$ . Suppose that the trajectory space  $\mathcal{L}_d$  meets the conditions of Proposition 14. Then the formula (3.1) solves the problem of finding  $X|_{\mathcal{P}}$  in terms of  $X|_{\mathcal{J} \setminus \mathcal{P}}$  and therefore it restores the missing values of the series  $\mathbf{X}_N$  that belong to the vector  $X$ .

Let us turn to the case when the initial series with missing data is a sum of two separable series of finite rank  $\mathbf{X}_N^{(1)}$  and  $\mathbf{X}_N^{(2)}$ , i.e.  $\mathbf{X}_N = \mathbf{X}_N^{(1)} + \mathbf{X}_N^{(2)}$ . Assume that the trajectory spaces  $\mathcal{L}^{(1)}$  and  $\mathcal{L}^{(2)}$  of the series  $\mathbf{X}_N^{(1)}$  and  $\mathbf{X}_N^{(2)}$  correspondingly are given. Since the series are separable, their trajectory spaces are orthogonal:  $\mathcal{L}^{(1)} \perp \mathcal{L}^{(2)}$ . Again, let  $X$  be a lagged vector of the series  $\mathbf{X}_N$  with indices of missing data from  $\mathcal{P}$  and  $X = X^{(1)} + X^{(2)}$ , where  $X^{(i)}$ ,  $i = 1, 2$ , are the corresponding lagged vectors of  $\mathbf{X}_N^{(i)}$ . Let us solve the problem of finding the lagged vector  $X^{(1)}$  of the first series in terms of  $X|_{\mathcal{J} \setminus \mathcal{P}}$ .

The problem splits into two: finding  $X^{(1)}|_{\mathcal{J} \setminus \mathcal{P}}$  and finding  $X^{(1)}|_{\mathcal{P}}$ . If the first problem is solved, we can apply the theory of Subsection 3.1.2 with  $m = \text{rank } \mathbf{X}_N^{(1)}$  and  $\mathfrak{D}_m = \mathcal{L}^{(1)}$  to find  $X^{(1)}|_{\mathcal{P}}$  in

terms of  $X^{(1)}|_{\mathcal{J} \setminus \mathcal{P}}$ . The propositions of Section 3.1.2 provide us solution to the problem of finding  $X^{(1)}|_{\mathcal{J} \setminus \mathcal{P}}$  if the condition of orthogonality of the restricted trajectory spaces  $\mathcal{L}^{(1)}|_{\mathcal{J} \setminus \mathcal{P}}$  and  $\mathcal{L}^{(2)}|_{\mathcal{J} \setminus \mathcal{P}}$  holds. Since the subspaces  $\mathcal{L}^{(1)}$  and  $\mathcal{L}^{(2)}$  are orthogonal, the lagged vector  $X^{(1)}$  of the series  $\mathbf{X}_N^{(1)}$  is equal to  $\mathbf{\Pi}^{(1)}\mathbf{X}$ ; therefore the corresponding terms of the series  $\mathbf{X}_N^{(1)}$  can be obtained by orthogonal projection of the vector  $X|_{\mathcal{J} \setminus \mathcal{P}}$  onto  $\mathcal{L}^{(1)}|_{\mathcal{J} \setminus \mathcal{P}}$ .

Thus,  $X^{(1)}|_{\mathcal{J} \setminus \mathcal{P}} = \mathbf{\Pi}_{\mathcal{J} \setminus \mathcal{P}}^{(1)} X|_{\mathcal{J} \setminus \mathcal{P}}$ , where  $\mathbf{\Pi}_{\mathcal{J} \setminus \mathcal{P}}^{(1)}$  is defined by formula (3.4) with the matrix  $\mathbf{R}$  whose columns are the vectors of the orthonormal basis of  $\mathcal{L}^{(1)}$ , and  $X^{(1)}|_{\mathcal{P}}$  is expressed in terms of  $X|_{\mathcal{J} \setminus \mathcal{P}}$  by the formula (3.1) with the same matrix  $\mathbf{R}$ . Thereby, the values of the series  $\mathbf{X}_N^{(1)}$  belonging to its lagged vector  $X^{(1)}$  have been found, including those which are located on the places of missing data of the series  $\mathbf{X}_N$ .

### 3.1.4. Finding trajectory spaces of the initial time series and of its additive components

Consider a time series  $\mathbf{X}_N$  with  $\text{rank}_L(\mathbf{X}_N) = d$  and its  $L$ -lagged vectors  $\{X_i\}_{i=1}^K$ . At first, let us obtain conditions of possibility to find the basis of the trajectory space  $\mathcal{L}_d = \text{span}(X_i, i = 1, \dots, K)$  using only non-missing values of the observed series. Denote by  $\mathcal{C} \subset \{1, \dots, K\}$  the set of numbers of the complete lagged vectors with no missing entries. Assume that  $\mathcal{C} \neq \emptyset$  and consider the matrix  $\tilde{\mathbf{X}}$  consisting of lagged vectors  $X_i, i \in \mathcal{C}$ , as its columns. Let  $\tilde{\mathcal{L}}_d = \text{span}(X_i, i \in \mathcal{C})$ . It is easy to prove the following proposition.

**Proposition 17.** *The set  $X_i, i \in \mathcal{C}$ , contains at least  $d$  linearly independent vectors if and only if  $\tilde{\mathcal{L}}_d = \mathcal{L}_d$ . As this takes place, the eigenvectors  $U_1, \dots, U_d$  of the matrix  $\tilde{\mathbf{S}} = \tilde{\mathbf{X}}\tilde{\mathbf{X}}^T$  corresponding to  $d$  nonzero eigenvalues of  $\tilde{\mathbf{S}}$  form an orthonormal basis of the subspace  $\mathcal{L}_d$ .*

Let us formulate a more constructive sufficient condition.

**Proposition 18.** *If the time series  $\mathbf{X}_N$  has  $L$ -rank  $d$ ,  $\mathbf{e}_1 \notin \mathcal{L}_d$ ,  $\mathbf{e}_L \notin \mathcal{L}_d$  and the series contains at least  $L + d - 1$  successive non-missing values, then  $\tilde{\mathcal{L}}_d = \mathcal{L}_d$ .*

*Proof.* In order to prove the proposition, let us show that there are  $d$  linearly independent vectors among complete lagged vectors. We conclude from the conditions of this proposition that there exists such a number  $k$  that lagged vectors  $\{X_{k+i}\}_{i=1}^d$  do not contain missing data.

Since  $\mathbf{e}_1 \notin \mathcal{L}_d$  and  $\mathbf{e}_L \notin \mathcal{L}_d$ , the time series  $\mathbf{X}_N$  can be continued to the infinite in both directions time series  $\mathbf{X}$  of  $L$ -rank  $d$  (the proof is based on Corollary 6 and Corollary 7 with  $n = L$ ,  $m = d$ ,  $\mathfrak{D}_m = \mathcal{L}_d$ ). It follows that  $\mathbf{X}$  is a time series of finite rank  $d$  and, in particular,  $\text{rank}_{d+1}\mathbf{X}_N = d$ ,  $\mathbf{e}_1, \mathbf{e}_{d+1} \notin \mathcal{L}^{(d+1)}(\mathbf{X}_N)$ . Therefore, due to Corollary 6 with  $n = d + 1$ ,  $m = d$ ,  $\mathfrak{D}_m = \mathcal{L}^{(d+1)}(\mathbf{X}_N)$ ,

$$X_{i+d} = \sum_{j=1}^d a_j X_{i+d-j}, \quad 1 \leq i \leq K-d.$$

Hence all lagged vectors with indices exceeding  $k + d$  can be expressed as a linear combination of vectors  $X_{k+1}, \dots, X_{k+d}$ .

Analogously, taking into account Corollary 7, we find that lagged vectors with indices from 1 to  $k$  can be written as linear combinations of vectors  $X_{k+1}, \dots, X_{k+d}$ . Therefore, all the lagged vectors are expressed through the set of vectors  $\{X_{k+i}\}_{i=1}^d$ . Since dimension of the trajectory space is equal to  $d$ , these vectors are independent.  $\square$

We now turn to the case when the observed series is a sum of two separable series:  $\mathbf{X}_N = \mathbf{X}_N^{(1)} + \mathbf{X}_N^{(2)}$ . Assume that the conditions of Proposition 17 are met and a basis of the trajectory space  $\mathcal{L}_d$  of the series  $\mathbf{X}_N$  has been found. Let us formulate the problem of finding the trajectory spaces  $\mathcal{L}^{(1)}$  and  $\mathcal{L}^{(2)}$  of the series  $\mathbf{X}_N^{(1)}$  and  $\mathbf{X}_N^{(2)}$  using  $\tilde{\mathbf{X}} = \tilde{\mathbf{X}}^{(1)} + \tilde{\mathbf{X}}^{(2)}$ .

If the row spaces of the trajectory matrices  $\tilde{\mathbf{X}}^{(1)}$  and  $\tilde{\mathbf{X}}^{(2)}$  are orthogonal and so are the column spaces, then the eigenvectors  $U_1, \dots, U_d$  of  $\tilde{\mathbf{S}}$  forming the basis of  $\mathcal{L}_d$  can be generally split into two groups constituting bases of the spaces  $\mathcal{L}^{(1)}$  and  $\mathcal{L}^{(2)}$ . The trajectory spaces of columns are orthogonal due to the separability of the series  $\mathbf{X}_N^{(1)}$  and  $\mathbf{X}_N^{(2)}$ . It is easy to show that the following conditions are necessary and sufficient for orthogonality of the row trajectory spaces:

$$\sum_{k \in \mathcal{C}} x_{i+k-1}^{(1)} x_{j+k-1}^{(2)} = 0, \quad i, j = 1, \dots, L. \quad (3.5)$$

Thus, (3.5) and the conditions of Proposition 17 are sufficient to find bases of the trajectory spaces  $\mathcal{L}^{(1)}$  and  $\mathcal{L}^{(2)}$ .

Sum of a constant and a harmonic series of period  $T$  can be given as an example. It is known that if  $L$  and  $K$  are divisible by  $T$ , then the series are separable. If  $T$  successive values are missing, and  $N$  and  $L$  are large enough, then it is still possible to find bases of the spaces  $\mathcal{L}^{(1)}$  and  $\mathcal{L}^{(2)}$ .

### Conditions for orthogonality of the restricted subspaces

Properties of the operator of orthogonal projection  $\Pi_{\mathcal{I} \setminus \mathcal{P}}^{(1)}$  were obtained in Proposition 16 of Section 3.1.2 (and were used in Section 3.1.3). In the mentioned proposition, the vector  $X$  and the set  $\mathcal{P}$  of indices of vector's missing components were considered under the condition that the spaces  $\mathcal{L}^{(1)}|_{\mathcal{I} \setminus \mathcal{P}}$  and  $\mathcal{L}^{(2)}|_{\mathcal{I} \setminus \mathcal{P}}$  are orthogonal.

Let  $\mathcal{L}^{(1)}$  and  $\mathcal{L}^{(2)}$  be trajectory spaces of the series  $\mathbf{X}_N^{(1)}$  and  $\mathbf{X}_N^{(2)}$ . The following equalities give us the necessary and sufficient condition of orthogonality  $\mathcal{L}^{(1)}|_{\mathcal{I} \setminus \mathcal{P}} \perp \mathcal{L}^{(2)}|_{\mathcal{I} \setminus \mathcal{P}}$ :

$$\sum_{k \in \mathcal{I} \setminus \mathcal{P}} x_{i+k-1}^{(1)} x_{j+k-1}^{(2)} = 0, \quad i, j = 1, \dots, K. \quad (3.6)$$

**Remark 24.** If the series  $\mathbf{X}_N^{(1)}$  and  $\mathbf{X}_N^{(2)}$  are separable, then the condition (3.6) is equivalent to

$$\sum_{k \in \mathcal{P}} x_{i+k-1}^{(1)} x_{j+k-1}^{(2)} = 0, \quad i, j = 1, \dots, K. \quad (3.7)$$

Sum of a harmonic series of period  $T$  and a constant series, where  $L, K$  and the number of

the successive missing data are divisible by  $T$ , is an example when the condition (3.7) is satisfied. Also,  $L$  should be greater than the number of missing values of the time series.

### 3.1.5. Comments to implementation of the subspace gap-filling method

Here we describe details of implementation of Algorithms 3.1 and 3.2.

#### Methods for reconstructing values at the positions of the non-missing components

Let us describe possible solutions to the problem of calculation of  $\widehat{X}_i|_{\mathcal{J} \setminus \mathcal{P}}$ ,  $l \leq i \leq p$ .

The method when the operator of orthogonal projection onto  $\mathcal{M}_r|_{\mathcal{J} \setminus \mathcal{P}}$  is applied to  $X_i|_{\mathcal{J} \setminus \mathcal{P}}$  for each  $\mathcal{P}$ -incomplete lagged vectors of the cluster of missing data is called “ $\Pi$  Projector”. Proposition 15 with  $m = r$ ,  $\mathcal{L}^{(1)} = \mathcal{M}_r$ , and the matrix  $\mathbf{R}$ , whose columns are the eigenvectors  $U_i, i \in I_r$ , provides conditions of applying this method and the formula for calculation of  $\Pi_{\mathcal{J} \setminus \mathcal{P}}^{(1)}$  in  $\widehat{X}_i|_{\mathcal{J} \setminus \mathcal{P}} = \Pi_{\mathcal{J} \setminus \mathcal{P}}^{(1)} X_i|_{\mathcal{J} \setminus \mathcal{P}}$ .

One more method can be applied to continuous (extreme or inner) clusters. This method is based on the fact that values of the diagonal entries of the trajectory matrices (recall that  $\widehat{\mathbf{X}}$  is an approximation of the trajectory matrix of the series  $\mathbf{X}_N^{(1)}$ ) with indices  $(i, j)$ ,  $i + j = \text{Const}$ , are equal. Let us consider an inner cluster at first. The positions of the non-missing components  $\widehat{X}_q|_{\mathcal{J} \setminus \mathcal{P}}$  in the corresponding set of vectors  $\widehat{X}_q$ ,  $q = l, \dots, p$ , form two “triangles” (right and left). Denoting by  $\widehat{x}_{i,j}$  the  $i$ -th element of the vector  $\widehat{X}_j$ , we will describe the method by the example of filling in the left “triangle”, which will be expressed as the set of the  $s$ -th “diagonals”:  $\{(i, j) : i + j = s, l + 1 \leq s \leq l + L - 1, l \leq j \leq l + L - 2\}$ . Note that “triangle” and “diagonal” are interpreted as parts of the matrix  $[\widehat{X}_l : \dots : \widehat{X}_p]$ .

Since we deal with the inner cluster, there are  $l_0$ ,  $l_0 \geq 1$ , left adjacent vectors  $\widehat{X}_{l-m}$ ,  $m = 1, \dots, l_0$ , which are complete and all their components have already been calculated at the previous step. If we consider the matrix  $[\widehat{X}_{l-l_0} : \dots : \widehat{X}_p]$ , then the method, which can be called “Components of adjacent vectors”, consists in the replacement of all the components of the  $s$ -th diagonal  $\widehat{x}_{i,j}$  with  $i + j = s$  and  $l - l_0 \leq j \leq l + L - 2$  by the average value of  $\widehat{x}_{i,j}$  with  $i + j = s$ ,  $l - l_0 \leq j \leq l - 1$ . Note that the method also changes some components of vectors  $\widehat{X}_{l-l_0}, \dots : \widehat{X}_{l-1}$ , since it uses the same approach as in the vector SSA forecasting.

A fully similar procedure is carried out with the right “triangle”. For extreme clusters, only one of “triangles” needs to be filled.

#### Methods for reconstructing values at the positions of missing components

Here we propose several solutions to the problem of calculation of  $\widehat{X}_i|_{\mathcal{P}}$ ,  $l \leq i \leq p$ , if vectors  $\widehat{X}_i|_{\mathcal{J} \setminus \mathcal{P}}$  have already been obtained.

For the first method, we will require each vector of the lagged-vectors set to satisfy the conditions of Proposition 14. If all the conditions are met, then we can reconstruct missing

components of each vector by the formula (3.1), where  $m = r$ ,  $\mathfrak{D}_m = \mathcal{M}_r$  and  $\mathbf{R}$  is a matrix whose columns are the eigenvectors  $U_i, i \in I_r$ . Such a method of reconstructing missing data can be called a “*simultaneous filling in*”. Note that conditions for applying the simultaneous filling in are restrictive enough. In particular, it is impossible to apply the method when there are no non-missing components just in one vector from the lagged-vectors set.

Other methods are based on the fact that values with indices  $(i, j)$  on the diagonals  $i + j = \text{Const}$  of the trajectory matrices are equal. Therefore, we can reconstruct missing components of one of the lagged vectors and use the obtained values for filling in missing entries of the adjacent vectors.

Consider the lagged-vectors set  $X_l, \dots, X_p$  of the considered cluster of missing values. We assume that the vectors  $\widehat{X}_q|_{\mathcal{J} \setminus \mathcal{P}}$ ,  $q = l, \dots, p$ , have already been calculated before. Let us describe several variants of the methods of sequential filling in.

1) Let the considered cluster be not a left cluster. In this case the left vector  $X_l$  contains only one missing entry, which is located on the place of the last coordinate, so its set of indices of missing entries is  $\mathcal{P} = \{L\}$ . Therefore Corollary 6 can be applied. If  $\mathbf{e}_L \notin \mathcal{M}_r$ , then we can obtain the last value  $y$  of the vector  $\widehat{X}_l$  as a linear combination of the previous values with coefficients given by the formula (3.2). Further this value  $y$  is used to fill in the  $(L - j)$ -th components of the vectors  $\widehat{X}_{l+j}$ ,  $j = 1, \dots, L - 1$ . After this procedure the adjacent vector  $\widehat{X}_{l+1}$  will have only one missing entry on the place of the last coordinate (if the cluster is continuous). Therefore the whole procedure may be repeated once more. And so on. Such a method will be called a *sequential filling in from the left*. Note that if the cluster of missing data is not continuous, the filling procedure need not be applied to some of the vectors  $X_q$ ,  $l < q < p$ .

2) The *sequential filling in from the right* is fully analogous to the sequential filling in from the left and comes down to filling in the first coordinate by means of the formula (3.3) given in Corollary 7. Therefore, the applicable conditions of this method are: the cluster of missing data is not a right cluster and  $\mathbf{e}_1 \notin \mathcal{M}_r$ .

3) Different combination of the sequential filling in from the left and from the right (so called two-sided methods) can be considered.

Note that the conditions of sequential filling in are less restrictive in comparison with simultaneous imputation; however, errors can accumulate.

**Remark 25.** Consider a continuous cluster of missing data of length  $m$ , which is a right extreme cluster (and there are no other clusters of missing data in the series). If the methods “Components of adjacent vectors” and “Sequential filling in from the left” are applied to this cluster, the result will coincide with the recurrent forecast for  $m$  terms ahead; the forecasted time series component is extracted by Basic SSA from the times series consisting of the first  $N - m$  points.

## Example

To demonstrate the work of the methods of filling in missing data, let us consider the famous time series of length 144 representing monthly numbers of passengers (in thousands) on the international airlines, since January, 1949 (the data was published for the first time in [135]).

Let us remove 12 known values, starting with the 68-th point (i.e. we consider that values are unknown for a year since August, 1954). For such kind of artificially missing data we can estimate the accuracy of their recovering for different versions of the algorithm. Also, to simulate forecast, we add 12 missing data after the last, 144-th point of the series. The time series obtained is illustrated in Fig. 3.1.

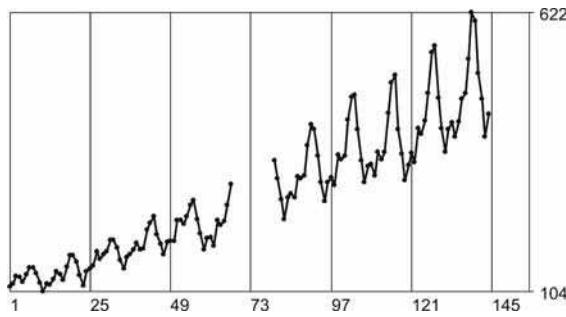


Figure 3.1: The initial time series with missing data

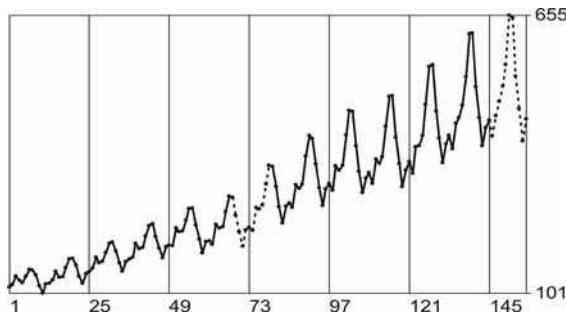


Figure 3.2: The reconstructed time series with filled in data

The first question is how to choose the window length  $L$ . In the case of no missing data, the general recommendation is to choose the window length close to the half of the series length and divisible by the period of expected periodicity (12 months here). The window length equal to 72 meets these conditions. However, under such choice of  $L$  all the lagged vectors will contain missing data. Let us choose a smaller length. The choice of  $L = 36$  provides us 62 complete lagged vectors with no missing data.

The analysis of the eigenvectors  $U_1, \dots, U_L$  (for real-life series it is common that  $d = L$ ) shows that the eigenvectors with indices 1, 6, and 9 correspond to a trend and the eigenvectors with

indices 2–5, 7–8, and 10–13 correspond to a seasonal component. All the rest eigenvectors may be classified as produced by noise. Therefore, we will choose  $r = 13$  and  $I_r = \{1, 2, \dots, 13\}$  in order to reconstruct the deterministic component of the series (a signal).

All the eight variants described above are applicable to the first continuous inner cluster of missing data: two variants of reconstruction of values at the places of non-missing data (“ $\Pi$  Projector” and “Components of adjacent vectors”), and four variants of filling in missing data (simultaneous filling in and three types of sequential filling in). A comparison of the reconstruction results with the values that were artificially removed from the initial time series shows an advantage of the variant “ $\Pi$  Projector” with simultaneous filling in the missing data. Reconstruction error therewith is approximately equal to 6 for the missing data and is equal to 4.75 (this is not much less) for other terms.

We apply the same method to fill in the second cluster of missing data. The result is illustrated in Fig. 3.2. The reconstructed series is marked by the dotted line in the area of missing data.

### 3.2. Possibility of construction of weights in the HSLRA problem

In this section, in accordance with deconvolution notation, we will numerate the vector coordinates from 0 and denote vectors by bold lowercase letters.

Our motivation is the problem of extraction of a signal  $\mathbf{s} = (s_0, s_1, \dots, s_{N-1})^T$  from an observed noisy signal  $\mathbf{x} = (x_0, x_1, \dots, x_{N-1})^T = \mathbf{s} + \boldsymbol{\xi}$  of length  $N$ , where  $\boldsymbol{\xi} = (\xi_0, \xi_1, \dots, \xi_{N-1})^T$  is a vector of (unobserved) random noise with zero mean and covariance matrix  $\boldsymbol{\Sigma} = E\boldsymbol{\xi}\boldsymbol{\xi}^T$ .

As before, we consider the signal model (1). This class of signals can be defined through low-rank Hankel matrices as follows. Set a window length  $L$ ,  $1 < L \leq N/2$ ;  $K = N - L + 1$ . With a series  $\mathbf{z} = (z_0, z_1, \dots, z_{N-1})^T$ , we associate the trajectory matrix

$$\mathcal{T}_L(\mathbf{z}) = \begin{pmatrix} z_0 & z_1 & \dots & z_{K-1} \\ z_1 & \dots & \dots & z_K \\ \dots & \dots & \dots & \dots \\ z_{L-1} & z_L & \dots & z_{N-1} \end{pmatrix} \in \mathbb{R}^{L \times K}.$$

The problem of finite-rank signal extraction can be reduced to the problem of approximation of the  $L$ -trajectory matrix  $\mathcal{T}_L(\mathbf{x})$  of the observed time series  $\mathbf{x}$  by a Hankel matrix of rank  $r$ , i.e., the problem HSLRA [136, 2, 137].

The HSLRA problem can be stated in two forms: (a) vector form and (b) matrix form. The vector (time series) form of this problem is: for given  $\mathbf{x} \in \mathbb{R}^N$  and positive integer  $r < \lfloor N/2 \rfloor$ ,

$$\|\mathbf{x} - \mathbf{y}\|_{\mathbf{W}}^2 \rightarrow \min_{\mathbf{y}: \text{rank } \mathbf{y} \leq r}, \quad (3.8)$$

where  $\mathbf{y} = (y_0, y_1, \dots, y_{N-1})^T$ ,  $\|\mathbf{z}\|_{\mathbf{W}}^2 = \mathbf{z}^T \mathbf{W} \mathbf{z}$  for  $\mathbf{z} \in \mathbb{R}^N$  and  $\mathbf{W}$  is some positive definite matrix of

size  $N \times N$ .

The solution of (3.8) can be considered as a weighted least-square estimate (WLSE) of the signal  $\mathbf{s}$ . If noise  $\boldsymbol{\xi}$  is Gaussian with covariance matrix  $\boldsymbol{\Sigma}$ , then the WLSE with  $\mathbf{W}_0 = \boldsymbol{\Sigma}^{-1}$  is the maximum likelihood estimate (MLE). If the properties of the noise process are known then the vector form (3.8) is the most natural way of defining the HSLRA problem. In the paper [22], the fast and stable algorithm MGN (Modified Gauss-Newton) for solving (3.8), which is based on the approach from [24], is proposed and it is shown that MGN overcomes the existing algorithm presented in [138].

Although the vector form solution allows fast implementations, these implementations are very complex and need a starting point that is close to the solution [139, 138].

The matrix form of the HSLRA problem allows one to use simple subspace-based alternating projection methods (e.g., the Cadzow iterations (1.22)) and hence is computationally much preferable than the vector form (3.8); see [30, Sect. 3.4] for details. Note that SSA can formally be considered as one Cadzow iteration.

Define the inner product in  $\mathbb{R}^{L \times K}$  as  $\langle \mathbf{X}, \mathbf{Y} \rangle = \langle \mathbf{X}, \mathbf{Y} \rangle_{\mathbf{A}, \mathbf{B}} = \text{tr}(\mathbf{AXBY}^T)$ , where  $\mathbf{A} = (a_{i,j})_{i,j=0}^{L-1} \in \mathbb{R}^{L \times L}$ ,  $\mathbf{B} = (b_{i,j})_{i,j=0}^{K-1} \in \mathbb{R}^{K \times K}$ ;  $\|\mathbf{X}\|_{\mathbf{A}, \mathbf{B}}$  is the corresponding matrix norm in  $\mathbb{R}^{L \times K}$ . The HSLRA problem in the matrix form is the following optimization problem:

$$\|\mathbf{X} - \mathbf{Y}\|_{\mathbf{A}, \mathbf{B}}^2 \rightarrow \min_{\mathbf{Y} \in \mathcal{M}_r \cap \mathcal{H}}, \quad (3.9)$$

where  $\mathcal{H} \subset \mathbb{R}^{L \times K}$  is the space of Hankel matrices of size  $L \times K$ ,  $\mathcal{M}_r \subset \mathbb{R}^{L \times K}$  is the set of matrices of rank not larger than  $r$ . For reformulating the original HSLRA problem (3.8) in the matrix form (3.9), we have to choose  $\mathbf{X} = \mathcal{T}_L(\mathbf{x})$  and  $\mathbf{Y} = \mathcal{T}_L(\mathbf{y})$ ; the remaining issue then is to match the vector norm in (3.8) with the matrix norm in (3.9).

In [23], it is proposed to use mixed alternating projections to solve the problem in the form (3.9), where first weight matrices corresponding to slower convergence to a more exact solution of (3.8) is taken, and then weight matrices corresponding to faster convergence to a less exact solution is considered. It turns out that this mixture helps to obtain a more exact and faster solution. In [140] a numerical algorithm for the search of approximate matrix weights, which gives suboptimal weights at a given degree of degeneracy of the weight matrices given by the parameter  $\alpha$ ,  $0 < \alpha \leq 1$ , is considered. This algorithm is used in the construction of mixed alternating projections in [23], since the parameter  $\alpha$  allows improving the accuracy of the solution for the problem (3.8), while slowing the convergence; and vice versa.

The rest of this section is devoted to solving the HSLRA problem in the matrix form (3.9), which is convenient for finding the solution by the Cadzow iterations (1.22). Namely, it will be shown that choosing weights in (3.9) that provides exactly the optimal solution is impossible for the case of autoregressive noise [19]. In particular, therefore, the application of methods of numerical search for suboptimal weights is reasonable.

Let us start with auxiliary results.

### 3.2.1. Matrix and vectors convolution

Let  $A$  and  $B$  be positive integers and  $\mathbf{a} = (a_0, \dots, a_A)^T \in \mathbb{R}^{A+1}$  and  $\mathbf{b} = (b_0, \dots, b_B)^T \in \mathbb{R}^{B+1}$  be two vectors. The convolution of vectors  $\mathbf{a}$  and  $\mathbf{b}$  is the vector  $\mathbf{c} = \mathbf{a} * \mathbf{b} = (c_0, \dots, c_C)^T \in \mathbb{R}^{C+1}$  with  $C = A + B$  and  $c_i = \sum_k a_k b_{i-k}$ , where  $i = 0, 1, \dots, C$  and the sum taken over the set of indices  $k$  such that the elements  $a_k$  and  $b_{i-k}$  are defined (that is,  $\max\{0, i-B\} \leq k \leq \min\{A, i\}$ ).

The definition of vector convolution naturally extends to matrices as follows. Let  $\mathbf{A} = (a_{i,j})_{i,j=0}^A$  and  $\mathbf{B} = (b_{i,j})_{i,j=0}^B$  be matrices of sizes  $(A+1) \times (A+1)$  and  $(B+1) \times (B+1)$ , respectively. A matrix  $\mathbf{C} = (c_{i,j})_{i,j=0}^{A+B}$  is a convolution of the matrices  $\mathbf{A}$  and  $\mathbf{B}$  if  $c_{i,j} = \sum_{k,l} a_{k,l} b_{i-k,j-l}$ , where the sum is taken over the sets of indices  $k$  and  $l$  such that the elements  $a_{k,l}$  and  $b_{i-k,j-l}$  are defined; that is,  $\max\{0, i-B\} \leq k \leq \min\{A, i\}$  and  $\max\{0, j-B\} \leq l \leq \min\{B, j\}$ .

The generating function (gf) of a vector  $\mathbf{u} = (u_0, \dots, u_M)^T$  is defined as  $G_{\mathbf{u}}(t) = u_0 + u_1 t + \dots + u_M t^M$ . Similarly, the generating function of a matrix  $\mathbf{U} = (u_{i,j})_{i,j=0}^M$  is  $G_{\mathbf{U}}(t, s) = \sum_{i,j=0}^M u_{i,j} t^i s^j$ , where  $t, s \in \mathbb{C}$ .

From the definition of convolution,  $\mathbf{c} = \mathbf{a} * \mathbf{b}$  if and only if  $G_{\mathbf{c}}(t) = G_{\mathbf{a}}(t)G_{\mathbf{b}}(t)$  for all  $t \in \mathbb{C}$ . Similarly,  $\mathbf{C} = \mathbf{A} * \mathbf{B}$  if and only if  $G_{\mathbf{C}}(t, s) = G_{\mathbf{A}}(t, s)G_{\mathbf{B}}(t, s)$  for all  $t, s \in \mathbb{C}$ . This yields that all statements about vector and matrix convolutions and deconvolutions can be equivalently formulated in the language of generating functions.

A vector  $\mathbf{c}$  can be blindly deconvoluted if there exist vectors  $\mathbf{a}$  and  $\mathbf{b}$  such that  $\mathbf{c} = \mathbf{a} * \mathbf{b}$ . This problem (interesting only under some restrictions on  $\mathbf{a}$  and  $\mathbf{b}$ ) is equivalent to studying the roots of the generating function  $G_{\mathbf{c}}(t)$ .

A matrix  $\mathbf{C}$  can be blindly deconvoluted if there exist matrices  $\mathbf{A}$  and  $\mathbf{B}$  such that  $\mathbf{C} = \mathbf{A} * \mathbf{B}$ .

There exists an extensive literature related to matrix convolution and (blind) deconvolution but it is mostly related to applications in image processing; see e.g. [141, 142]. We are interested in the blind deconvolution of matrices  $\mathbf{C}$  which are proportional to inverses of the autocovariance matrices of autoregressive processes (ARs).

### 3.2.2. Relation between vector and matrix forms of HSLRA

Particular cases of the correspondence between the vector-norm and matrix-norm formulations (3.8) and (3.9) of the HSLRA problem were considered in [143] and [15]. The general case is established in the following theorem.

**Theorem 6.** *For any  $\mathbf{z} \in \mathbb{R}^N$ ,  $\|\mathcal{T}_L(\mathbf{z})\|_{\mathbf{A}, \mathbf{B}} = \|\mathbf{z}\|_{\mathbf{W}}$  if and only if*

$$\mathbf{W} = \mathbf{A} * \mathbf{B}. \quad (3.10)$$

*Proof.* Consider the squared norm  $\|\mathbf{X}\|_{\mathbf{A}, \mathbf{B}}^2 = \text{tr}(\mathbf{A} \mathbf{X} \mathbf{B} \mathbf{X}^T)$  with  $\mathbf{X} = \mathcal{T}_L(\mathbf{z})$  so that  $x_{l,k} = z_{l+k}$  for  $l = 0, \dots, L-1$  and  $k = 0, \dots, K-1$ . We have

$$\|\mathbf{X}\|_{\mathbf{A}, \mathbf{B}}^2 = \sum_{l, l', k, k'} a_{l, l'} x_{l', k'} b_{k', k} x_{l, k} = \sum_{l, l', k, k'} a_{l, l'} z_{l'+k'} b_{k', k} z_{l+k} = \sum_{l, l', k, k'} z_{n'} a_{l, l'} b_{n'-l', n-l} z_n,$$

where  $n = k + l$ ,  $n' = k' + l'$  and all sums above are taken for  $l, l' = 0, \dots, L - 1$  and  $k, k' = 0, \dots, K - 1$ . By changing the summation indices in the last sum  $k \rightarrow n$  and  $k' \rightarrow n'$  we obtain the required.  $\square$

In a typical application, when the structure of the noise in the model ‘signal plus noise’ is assumed, the HSLRA problem is formulated in a vector form with a given matrix  $\mathbf{W}$ . As mentioned above, algorithms of solving the HSLRA problem are much easier if we have the matrix rather than vector form of the HSLRA problem. Therefore, in view of Theorem 6, for a given  $\mathbf{W}$  we would want to find positive definite matrices  $\mathbf{A}$  and  $\mathbf{B}$  such that (3.10) holds; that is, we would want to perform a blind deconvolution of the matrix  $\mathbf{W}$ .

Matrices  $\mathbf{A}$  and  $\mathbf{B}$  in (3.9) and therefore in (3.10) should be symmetric non-negative definite, see e.g. [144] and [30, p.62]. In Theorems 2–4 below we shall require symmetry of matrices  $\mathbf{A}$  and  $\mathbf{B}$  in (3.10).

It follows from the results of [13] that in the case when the noise  $\xi$  is white, and therefore  $\mathbf{W} = \mathbf{I}_N$ , the matrix  $\mathbf{W}$  cannot be blindly deconvoluted under the condition that  $\mathbf{A}$  and  $\mathbf{B}$  are positive definite matrices; however, for a wide range of parameters  $N$  and  $L$  there are many pairs of non-negative definite diagonal matrices  $\mathbf{A}$  and  $\mathbf{B}$  such that (3.10) holds. This section extends results of [13] to the case of banded matrices corresponding to the case where the noise  $\xi$  forms an autoregressive process.

The white-noise model is the simplest and hence the most popular model of noise used for formulation of the ‘signal plus noise’ problems. The autoregressive model of noise is the second most common noise model used in such problems. In particular, in climatology, the most common model of noise is the so-called ‘red noise’; that is, an auto-regressive process of order one with a positive coefficient. Red noise suits SSA and related methods very well, since the spectral density of red noise is monotonic. Therefore, in addition to understanding of the equivalence between the vector and matrix forms of the HSLRA, a theoretical investigation of the HSLRA, SSA and other subspace-based methods with the autoregressive noise seems to be important too.

### 3.2.3. Generating functions and convolution of banded matrices

#### Generating function of a matrix via generating functions of its diagonals

In some cases (e.g, for banded matrices), it is natural to construct generating functions of matrices as a sum of generating functions of diagonals. Consider a matrix  $\mathbf{U} = \{u_{i,j}\}_{i,j=0}^M$  and let  $G_{\mathbf{U}}(t,s)$  be its generating function.

Define  $\text{diag}_i(\mathbf{U})$ , the  $i$ -th diagonal of  $\mathbf{U}$ , as the vector of length  $M - |i| + 1$  with elements with indices  $(j,k)$  satisfying  $k - j = i$ ,  $i = -M, \dots, M$ . This  $i$ -th diagonal  $\text{diag}_i(\mathbf{U})$  has the univariate gf  $u_i(\tau) = G_{\text{diag}_i(\mathbf{U})}(\tau) = \sum_{j=0}^{M-|i|} u_{j,j+i} \tau^j$ .

**Lemma 10.**

$$G_{\mathbf{U}}(t, s) = \sum_{i=-M}^M t^{(|i|-i)/2} s^{(|i|+i)/2} \mathbf{u}_i(ts).$$

If  $\mathbf{U}$  is symmetric, this formula simplifies to  $G_{\mathbf{U}}(t, s) = \mathbf{u}_0(ts) + \sum_{i=1}^M (t^i + s^i) \mathbf{u}_i(ts)$ .

*Proof.*

$$\begin{aligned} G_{\mathbf{U}}(t, s) &= \sum_{j,k=0}^M u_{j,k} t^j s^k = \sum_{i=-M}^M \sum_{k=j-i}^M u_{j,k} t^j s^k = \\ &= \sum_{i=1}^M t^i \sum_{j=0}^{M-i} u_{j+i,j} t^j s^j + \sum_{i=1}^M s^i \sum_{j=0}^{M-i} u_{j,j+i} t^j s^j + \sum_{i=0}^M u_{i,i} t^i s^i = \\ &= \sum_{i=0}^M t^i \mathbf{u}_{-i}(ts) + \sum_{i=1}^M s^i \mathbf{u}_i(ts) = \sum_{i=-M}^0 t^{-i} \mathbf{u}_i(ts) + \sum_{i=1}^M s^i \mathbf{u}_i(ts) = \\ &= \sum_{i=-M}^M t^{(|i|-i)/2} s^{(|i|+i)/2} \mathbf{u}_i(ts). \end{aligned}$$

□

### Convolution of matrices expressed through convolution of diagonals; banded matrices

**Lemma 11.** Let  $\mathbf{A} = (a_{i,j})_{i,j=0}^A$ ,  $\mathbf{B} = (b_{i,j})_{i,j=0}^B$  and  $\mathbf{C} = \mathbf{A} * \mathbf{B}$ . For a given integer  $i$ , let  $\mathbf{a}_i(t) = G_{\text{diag}_i(\mathbf{A})}(t) = \sum_{j=0}^{A-|i|} a_{j,j+|i|} t^j$  ( $|i| \leq A$ ),  $\mathbf{b}_i(t) = G_{\text{diag}_i(\mathbf{B})}(t)$  ( $|i| \leq B$ ) and  $\mathbf{c}_i(t) = G_{\text{diag}_i(\mathbf{C})}(t)$  ( $|i| \leq C = A+B$ ) be the generating functions of the diagonals  $\text{diag}_i(\mathbf{A})$ ,  $\text{diag}_i(\mathbf{B})$  and  $\text{diag}_i(\mathbf{C})$ , respectively. Then

$$\mathbf{c}_i(t) = \sum_{j+k=i} t^{(|j|+|k|-|j+k|)/2} \mathbf{a}_j(t) \mathbf{b}_k(t) \quad (i = -C, \dots, C). \quad (3.11)$$

*Proof.* In view of Lemma 10, we should prove

$$\sum_{i=-C}^C t^{(|i|-i)/2} s^{(|i|+i)/2} \mathbf{c}_k(ts) = \sum_{i=-C}^C t^{(|i|-i)/2} s^{(|i|+i)/2} \sum_{j+k=i} (ts)^{(|j|+|k|-|j+k|)/2} \mathbf{a}_j(ts) \mathbf{b}_k(ts).$$

We have:

$$\begin{aligned}
& \sum_{i=-C}^C t^{(|i|-i)/2} s^{(|i|+i)/2} c_k(ts) \\
&= \left( \sum_{j=-A}^A t^{(|j|-j)/2} s^{(|j|+j)/2} a_j(ts) \right) \left( \sum_{k=-B}^B t^{(|k|-k)/2} s^{(|k|+k)/2} b_k(ts) \right) \\
&= \sum_{j=-A}^A \sum_{k=-B}^B t^{(|j|-j)/2} s^{(|j|+j)/2} a_j(ts) t^{(|k|-k)/2} s^{(|k|+k)/2} b_k(ts) \\
&= \sum_{i=-(A+B)}^{A+B} \sum_{j+k=i} t^{(|j|-j)/2} s^{(|j|+j)/2} t^{(|k|-k)/2} s^{(|k|+k)/2} a_j(ts) b_k(ts) \\
&= \sum_{i=-C}^C \sum_{j+k=i} t^{(|j|+|k|-(j+k))/2} s^{(|j|+|k|+(j+k))/2} a_j(ts) b_k(ts).
\end{aligned}$$

Since  $j+k = i$  within the sum, the proof is complete.  $\square$

**Remark 26.** For any real  $j$  and  $k$ , we have:

$$(|j| + |k| - |j+k|)/2 = \begin{cases} 0, & \text{if } jk \geq 0 \\ |k|, & \text{if } jk < 0, |j| \geq |k|; \\ |j|, & \text{if } jk < 0, |j| < |k|. \end{cases}$$

The following corollary is a reformulation of Lemma 11 using the explicit form for the diagonals of  $\mathbf{C}$ .

**Corollary 8.** Let matrices  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C} = \mathbf{A} * \mathbf{B}$  be as in Lemma 11. Then the  $i$ -th diagonal of  $\mathbf{C}$  is

$$\text{diag}_i(\mathbf{C}) = \sum_{j+k=i} \begin{pmatrix} 0_{(|j|+|k|-|j+k|)/2} \\ \text{diag}_j(\mathbf{A}) * \text{diag}_k(\mathbf{B}) \\ 0_{(|j|+|k|-|j+k|)/2} \end{pmatrix}, \quad (3.12)$$

where  $i = -C, \dots, C$ ,  $C = A + B$ , and  $0_m \in \mathbb{R}^m$  is a vector of zeros of size  $m$ .

In the case of banded matrices, Corollary 8 takes the following form.

**Corollary 9.** Let  $\mathbf{A} = (a_{i,j})_{i,j=0}^A$  be a  $(2p_1 + 1)$ -diagonal matrix with  $p_1 \leq A$ ,  $\mathbf{B} = (b_{i,j})_{i,j=0}^B$  be  $(2p_2 + 1)$ -diagonal with  $p_2 \leq B$  and  $\mathbf{C} = \mathbf{A} * \mathbf{B}$ . Then  $\mathbf{C}$  is  $(2p + 1)$ -diagonal with  $p = p_1 + p_2$  and the  $i$ -th diagonal  $\text{diag}_i(\mathbf{C})$  of  $\mathbf{C}$  ( $i = -p, \dots, p$ ) is given by (3.12); its gf is given by (3.11).

When  $p_2 = 0$  so that  $\mathbf{B}$  is diagonal, Corollary 9 gives the following particular case.

**Corollary 10.** Let  $\mathbf{A}$  be a  $(2p + 1)$ -diagonal matrix and  $\mathbf{B}$  be diagonal with  $\mathbf{b}$  on the main diagonal. Then  $\mathbf{C} = \mathbf{A} * \mathbf{B}$  is  $(2p + 1)$ -diagonal with  $\text{diag}_i(\mathbf{C}) = \text{diag}_i(\mathbf{A}) * \mathbf{b}$ ,  $i = -p, \dots, p$ ; in terms of gf, we have  $c_i(t) = a_i(t)b_0(t)$ ,  $i = -p, \dots, p$ .

### 3.2.4. Autocovariance matrices and their inverses for AR(1) and AR(2) models

#### Inverse autocovariance matrices

It is well known that the inverse to the autocovariance matrix of AR( $p$ ) is positive-definite symmetric  $(2p+1)$ -diagonal matrix [145, p.534]. Below we consider explicit forms of  $\Sigma^{-1}$  for the AR(1) and AR(2) models.

**AR(1)** Let  $\xi = (\xi_0, \xi_1, \dots, \xi_W)^T$  follows the AR(1) process

$$\xi_j = \phi_1 \xi_{j-1} + \varepsilon_j, \quad (3.13)$$

where  $j = 1, \dots, W$ ,  $\phi_1 \neq 0$ , and  $\varepsilon_1, \varepsilon_2, \dots$  are i.i.d. random variables with distribution  $\mathcal{N}(0, 1)$ ;  $\xi_j$  have distribution  $\mathcal{N}(0, \sigma^2)$  for all  $j = 0, \dots, W$ , where  $\sigma^2 = 1/(1 - \phi_1^2)$ . The condition of stationarity of the process (3.13) is  $|\phi_1| < 1$ . The covariance matrix  $\Sigma$  of the vector  $\xi$  is  $\Sigma = \sigma^2(\phi_1^{|i-j|})_{i,j=0}^W$ . The inverse of  $\Sigma$  is the tridiagonal matrix  $\Sigma^{-1} = \mathbf{W} \in \mathbb{R}^{(W+1) \times (W+1)}$ , where

$$\mathbf{W} = \begin{pmatrix} 1 & k_1 & 0 & 0 & \dots \\ k_1 & k_0 & k_1 & 0 & \dots \\ 0 & k_1 & k_0 & k_1 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ & & 0 & k_1 & k_0 & k_1 \\ & & 0 & 0 & k_1 & 1 \end{pmatrix}, \quad (3.14)$$

$k_0 = 1 + \phi_1^2$ ,  $k_1 = -\phi_1$ . Note that the matrix (3.14) is defined for any  $\phi_1$ , not necessarily for  $|\phi_1| < 1$ .

**AR(2)** Let  $\xi = (\xi_0, \xi_1, \dots, \xi_W)^T$  follows the AR(2) process

$$\xi_j = \phi_1 \xi_{j-1} + \phi_2 \xi_{j-2} + \varepsilon_j \quad (j = 2, \dots, W) \quad (3.15)$$

where  $\varepsilon_j$  are i.i.d. with distribution  $\mathcal{N}(0, 1)$  and  $\xi_j$  have distribution  $\mathcal{N}(0, \sigma^2)$  for all  $j = 0, \dots, W$ ; here  $\sigma^2 = (1 - \phi_2)/[(1 + \phi_1 - \phi_2)(1 - \phi_1 - \phi_2)(1 + \phi_2)]$ . The conditions of stationarity are  $\phi_2 + |\phi_1| < 1$  and  $|\phi_2| < 1$ . The region of stationarity of the AR(2) model is depicted in Figure 3.3. The inverse

of the covariance matrix  $\Sigma$  is a five-diagonal matrix  $\Sigma^{-1} = \mathbf{W} = (w_{i,j})_{i,j=0}^W$ , where

$$\mathbf{W} = \begin{pmatrix} 1 & k_{12} & k_2 & 0 & 0 & 0 & \dots \\ k_{21} & k_{22} & k_1 & k_2 & 0 & 0 & \dots \\ k_2 & k_1 & k_0 & k_1 & k_2 & 0 & \dots \\ 0 & k_2 & k_1 & k_0 & k_1 & k_2 & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 0 & k_2 & k_1 & k_0 & k_1 & k_2 \\ & & 0 & 0 & k_2 & k_1 & k_{22} & k_{12} \\ & & 0 & 0 & 0 & k_2 & k_{21} & 1 \end{pmatrix}, \quad (3.16)$$

$k_0 = 1 + \phi_1^2 + \phi_2^2$ ,  $k_1 = -\phi_1 + \phi_1\phi_2$ ,  $k_2 = -\phi_2$ ,  $k_{12} = k_{21} = -\phi_1$ ,  $k_{22} = 1 + \phi_1^2$ . The matrix (3.16) is defined for any  $\phi_1$  and  $\phi_2$ .

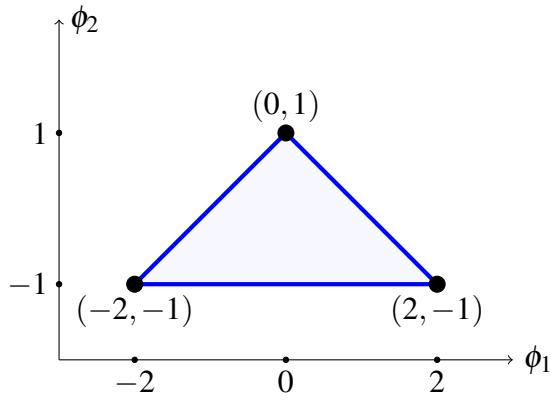


Figure 3.3: Region of stationarity of the AR(2) model and critical points

### Inverses of autocovariance matrices and their generating functions

Let  $\mathbf{W} = c\Sigma^{-1} = (w_{i,j})_{i,j=0}^W$ , where  $\Sigma$  is the autocovariance matrix of AR( $p$ ) with some  $p > 0$ , and  $c$  is chosen so that  $w_{0,0} = 1$  like in (3.14) and (3.16). Recall that the matrix  $\mathbf{W}$  is banded.

Consider the relation between diagonals of  $\mathbf{W}$  and their generating functions. Denote  $w_k(t) \stackrel{\text{def}}{=} G_{\text{diag}_k(\mathbf{W})}(t)$ ,  $t \in \mathbb{C}$ . For  $k \geq 0$ , an explicit formula for  $w_k$  is  $w_k(t) = \sum_{j=0}^{W-k} w_{j,j+k} t^j$ .

**Generating function of a vector of ones** Below we shall frequently use the following gf. Let  $\mathbf{c}_M = (1, \dots, 1)^T \in \mathbb{R}^{M+1}$ . Then

$$C_M(t) \stackrel{\text{def}}{=} G_{\mathbf{c}_M}(t) = 1 + \dots + t^M = \frac{1 - t^{M+1}}{1 - t}.$$

The following convenient formula connects polynomials  $C_{M+k}(t)$  and  $C_M(t)$  with  $k > 0$ :

$$C_{M+k}(t) = t^k C_M(t) + C_{k-1}(t). \quad (3.17)$$

**AR(1)** Consider the AR(1) model (3.13) with regression coefficient  $\phi_1$ . From (3.14), we have

$$w_1(t) = -\phi_1 C_{W-1}(t) \text{ and } w_0(t) = C_W(t) + \phi_1^2 t C_{W-2}(t).$$

Applying (3.17) with  $(k, M) = (1, W-1)$  and  $(k, M) = (1, W-2)$ , we obtain the following lemma.

**Lemma 12.** *For the AR(1) model (3.13),*

$$w_1(t) = -\phi_1 C_{W-1}(t), \quad w_0(t) = t C_{W-1}(t) + 1 + \phi_1^2 (C_{W-1}(t) - 1). \quad (3.18)$$

**AR(2)** Consider the AR(2) model (3.15) with regression coefficients  $\phi_1$  and  $\phi_2$ . From (3.16), we obtain

$$\begin{aligned} w_2(t) &= -\phi_2 C_{W-2}(t), \\ w_1(t) &= -\phi_1 C_{W-1}(t) + \phi_1 \phi_2 t C_{W-3}(t), \\ w_0(t) &= C_W(t) + \phi_1^2 t C_{W-2}(t) + \phi_2^2 t^2 C_{W-4}(t). \end{aligned}$$

Applying (3.17) with  $k = 1, M = W-3, W-2$  and with  $k = 2, M = W-4, W-2$ , we obtain the following lemma.

**Lemma 13.** *For the AR(2) model (3.15),*

$$\begin{aligned} w_2(t) &= -\phi_2 C_{W-2}(t), \\ w_1(t) &= -\phi_1 (t C_{W-2}(t) + 1) + \phi_1 \phi_2 (C_{W-2}(t) - 1), \\ w_0(t) &= t^2 C_{W-2}(t) + t + 1 + \phi_1^2 t C_{W-2}(t) + \phi_2^2 (C_{W-2}(t) - (t + 1)). \end{aligned} \quad (3.19)$$

### 3.2.5. Studying existence of solutions to the problem of blind deconvolution for the matrices proportional to inverses of covariance matrices in autoregressive models

For given square matrices  $\mathbf{A}$  and  $\mathbf{B}$ , we denote  $a_j(t) = G_{\text{diag}_j(\mathbf{A})}(t)$  and  $b_j(t) = G_{\text{diag}_j(\mathbf{B})}(t)$ .

**AR(1)**

**Theorem 7.** *Let  $W \geq 2$  and the matrix  $\mathbf{W}$  be defined by (3.14). There exist symmetric matrices  $\mathbf{A} = (a_{i,j})_{i,j=0}^A$  and  $\mathbf{B} = (b_{i,j})_{i,j=0}^B$  with  $A, B \geq 1$  such that  $\mathbf{W} = \mathbf{A} * \mathbf{B}$ , if and only if  $|\phi_1| = 1$ .*

*Proof.* Assume that  $\mathbf{W} = \mathbf{A} * \mathbf{B}$ . Since  $\mathbf{W}$  is 3-diagonal, in view of Corollary 10 and the assumption of symmetry of  $\mathbf{A}$  and  $\mathbf{B}$ , we can only consider the case when  $\mathbf{A}$  is 3-diagonal and  $\mathbf{B}$  is diagonal.

From (3.11) we obtain

$$w_1(t) = a_1(t)b_0(t), \quad w_0(t) = a_0(t)b_0(t), \quad \forall t \in \mathbb{C}. \quad (3.20)$$

Since  $\phi_1 \neq 0$ , from the left equalities in (3.18) and (3.20) we deduce that all the roots of  $C_{W-1}(t)$  are the roots of  $a_1(t)b_0(t)$ . Since  $B \geq 1$ , at least one of the roots of  $C_{W-1}(t)$  is a root of  $b_0(t)$ . Let  $t_1 \in \mathbb{C}$  be such root.

Suppose that  $|\phi_1| \neq 1$ . From the second equality in (3.20) we have  $w_0(t_1) = 0$  but the second equality in (3.18) yields  $w_0(t_1) = 1 - \phi_1^2 \neq 0$ . This contradiction proves the necessity of  $|\phi_1| = 1$ .

Assume now  $|\phi_1| = 1$  so that  $\phi_1 = \pm 1$ . In this case,  $w_0(t) = (t+1)C_{W-1}(t)$  and therefore, taking also into account the first equality in (3.18), the equalities (3.20) become

$$a_1(t)b_0(t) = -\phi_1 C_{W-1}(t), \quad a_0(t)b_0(t) = (t+1)C_{W-1}(t). \quad (3.21)$$

Represent  $C_{W-1}(t)$ , which is a polynomial of degree  $W-1 \geq 1$ , as a product of two non-zero polynomials  $p(t)$  and  $q(t)$ :

$$C_{W-1}(t) = p(t)q(t). \quad (3.22)$$

Here the polynomial  $p(t)$  can be a constant and  $q(t)$  has degree at least 1. Then we can choose

$$b_0(t) = q(t), \quad a_0(t) = (t+1)p(t) \text{ and } a_1(t) = -\phi_1 p(t). \quad (3.23)$$

□

## AR(2)

Consider the AR(2) model (3.15) with regression coefficients  $\phi_1$  and  $\phi_2$ . We aim at identifying matrices  $\mathbf{A}$  and  $\mathbf{B}$  so that (3.10) holds; that is,  $\mathbf{W} = \mathbf{A} * \mathbf{B}$ . Since  $\mathbf{W}$  is 5-diagonal and in view of Corollary 9 this may only happen in the following two cases: (a) each of  $\mathbf{A}$  and  $\mathbf{B}$  is 3-diagonal and (b)  $\mathbf{A}$  is 5-diagonal and  $\mathbf{B}$  is diagonal.

### A and B are symmetric and both matrices are 3-diagonal

**Theorem 8.** *Let  $W \geq 3$  and the matrix  $\mathbf{W}$  be defined in (3.16) with  $\phi_2 \neq 0$ . Assume that the matrices  $\mathbf{A} = (a_{i,j})_{i,j=0}^A$  and  $\mathbf{B} = (b_{i,j})_{i,j=0}^B$  are 3-diagonal and  $A \geq 2$ ,  $B \geq 1$ . There exist such matrices  $\mathbf{A}$  and  $\mathbf{B}$  with  $\mathbf{W} = \mathbf{A} * \mathbf{B}$ , if and only if either (i)  $\phi_2 = -1$  and  $|\phi_1| \geq 2$  or (ii)  $\phi_2 = 1$  and  $\phi_1 = 0$ .*

The proof of Theorem 8 is based on several lemmas.

**Lemma 14.** Under the conditions of Theorem 8, the existence of matrices  $\mathbf{A}$  and  $\mathbf{B}$  such that  $\mathbf{W} = \mathbf{A} * \mathbf{B}$ , implies that the polynomial  $w_1^2(t) - 4w_2(t)(w_0(t) - 2tw_2(t))$  is the square of a polynomial in  $t$ .

*Proof.* From (3.11) we obtain

$$\begin{aligned} w_2(t) &= a_1(t)b_1(t), \\ w_1(t) &= a_0(t)b_1(t) + a_1(t)b_0(t), \\ w_0(t) &= a_0(t)b_0(t) + 2ta_1(t)b_1(t). \end{aligned}$$

For brevity, we will omit the argument  $t$ . We have:

$$a_1 b_1 = w_2, \quad a_0 b_1 + a_1 b_0 = w_1, \quad a_0 b_0 = w_0 - 2tw_2.$$

Denote  $y = a_0 b_1$ ,  $z = a_1 b_0$ . Then we obtain

$$y + z = w_1, \quad yz = w_2(w_0 - 2tw_2).$$

This means that  $y$  and  $z$  are the roots of the quadratic equation  $x^2 + w_1 x + w_2(w_0 - 2tw_2) = 0$ . These roots are

$$x_{1,2} = \frac{w_1 \pm \sqrt{w_1^2 - 4w_2(w_0 - 2tw_2)}}{2}. \quad (3.24)$$

Since the roots should be polynomials,  $w_1^2 - 4w_2(w_0 - 2tw_2)$  should be a square of a polynomial in  $t$  (recall that  $w_i$  are polynomials in  $t$ ).  $\square$

**Lemma 15.** Assume that a polynomial  $P_{2M}(t)$  has the form

$$P_{2M}(t) = (P_1(t)P_{M-1}(t))^2 + Q_1(t)P_{M-1}(t) + P_0$$

for some  $M \geq 1$  (the lower indices mean polynomial degrees). If the square root of  $P_{2M}(t)$  exists then it has the form  $\pm P_1(t)P_{M-1}(t) + \text{const}$ .

*Proof.* Let  $P_1(t) = at + b$ ,  $Q_1(t) = dt + f$ . Then for  $\gamma = d/(2a)$  we have

$$P_{2M}(t) = (P_1(t)P_{M-1}(t) + \gamma)^2 + gP_{M-1}(t) + p_0,$$

where  $g$  and  $p_0$  are some constants. Let  $P_{2M}(t) = U_M^2(t)$ . Denote  $V_M(t) = P_1(t)P_{M-1}(t) + \gamma$ ,  $D_{M-1}(t) = gP_{M-1}(t) + p_0$ . Then

$$D_{M-1}(t) = U_M^2(t) - V_M^2(t) = (U_M(t) - V_M(t))(U_M(t) + V_M(t)).$$

The left-hand side part in this equation is a polynomial of order  $M - 1$  or less but the right-hand side part is either identical zero or a polynomial of order  $M$  or larger. Therefore this equality can be valid only if  $U_M(t) = \pm V_M(t)$ .  $\square$

**Lemma 16.** *For the matrix  $\mathbf{W}$  defined in (3.16) with  $\phi_2 \neq 0$ , the polynomial  $w_1^2(t) - 4w_2(t)(w_0(t) - 2tw_2(t))$  is a polynomial square if and only if at least one of the following relations hold:*

- (A)  $\phi_2 = -1$ , (B)  $\phi_2 = 1 - \phi_1$ , (C)  $\phi_2 = 1 + \phi_1$ .

*Proof.* For brevity, we will omit the polynomial argument  $t$ . Denote  $C = C_{W-2}(t)$ . Direct substitution using (3.19) gives

$$\begin{aligned} w_1^2 - 4(w_0 - 2tw_2)w_2 &= (-\phi_1 t + \phi_1 \phi_2)C - \phi_1(1 + \phi_2))^2 + \\ &+ 4\phi_2(t^2 + (\phi_1^2 + 2\phi_2)t + \phi_2^2)C^2 + 4C\phi_2(t + 1 - \phi_2^2(t + 1)). \end{aligned}$$

Consider this polynomial as a polynomial in  $C$ . By  $v_i$  we denote the coefficient for  $C^i$ ,  $i = 0, 1, 2$ . We have:

$$\begin{aligned} v_0 &= \phi_1^2(1 + \phi_2)^2, \\ v_1 &= 2(1 + \phi_2)(-\phi_1^2(\phi_2 - t) + 2\phi_2(t + 1)(1 - \phi_2)), \\ v_2 &= (\phi_1^2 + 4\phi_2)(t + \phi_2)^2. \end{aligned}$$

In view of Lemma 15, the determinant

$$v_1 - 4v_0v_2 = 4(1 + \phi_2)^2\phi_2(1 + \phi_1 - \phi_2)(1 - \phi_1 - \phi_2)((t + 1)^2\phi_2 + t\phi_1^2)$$

should be equal to zero. The solutions of the equation  $v_1 - 4v_0v_2 = 0$ , with respect to  $\phi_2$ , are:  $\phi_2 = 0, -1, 1 - \phi_1, 1 + \phi_1, -t\phi_1^2/(1 + t)^2$ . The root  $\phi_2 = 0$  is inappropriate, since  $\phi_2 \neq 0$  by the definition of AR(2). The root  $\phi_2 = -t\phi_1^2/(1 + t)^2$  is inappropriate as it depends on  $t$ . The proof is complete.  $\square$

*Proof of Theorem 8.* In view of Lemmas 14 and 16, matrices  $\mathbf{A}$  and  $\mathbf{B}$  such that  $\mathbf{W} = \mathbf{A} * \mathbf{B}$  holds may exist only in the three particular cases indicated in Lemma 16; that is, (A)  $\phi_2 = -1$ , (B)  $\phi_2 = 1 - \phi_1$ , (C)  $\phi_2 = 1 + \phi_1$ . Let us consider these three cases separately.

(A) Assume  $\phi_2 = -1$ . Then

$$w_2(t) = C_{W-2}(t), \quad w_1(t) = -\phi_1 C_{W-2}(t)(t + 1), \quad w_0(t) = C_{W-2}(t)(t^2 + \phi_1^2 t + 1).$$

From (3.19) and (3.24) we then have:

$$\begin{aligned} a_1(t)b_1(t) &= C_{W-2}(t), \\ a_0(t)b_1(t) + a_1(t)b_0(t) &= -\phi_1 C_{W-2}(t)(t + 1), \\ a_0(t)b_0(t) + 2ta_1(t)b_1(t) &= C_{W-2}(t)(t^2 + \phi_1^2 t + 1). \end{aligned} \tag{3.25}$$

Since polynomials  $a_0(t)$  and  $b_0(t)$  have one degree higher than the polynomials  $a_1(t)$  and  $b_1(t)$  respectively,  $a_1(t)b_1(t) = C_{W-2}(t)$  and  $a_0(t)b_0(t)$  contains  $C_{W-2}(t)$  as a multiplier, we obtain that  $a_0(t) = (\lambda_0 + \lambda_1 t)a_1(t)$  and  $b_0(t) = (\mu_0 + \mu_1 t)b_1(t)$  for some  $\lambda_0, \lambda_1, \mu_0$  and  $\mu_1$ . Substituting this into equations (3.25) and cancelling  $C_{W-2}(t)$ , which is a common multiplier in all equations, we obtain the following two equations for  $\lambda_0, \lambda_1, \mu_0$  and  $\mu_1$ :

$$\begin{cases} (\mu_0 + \mu_1 t)(\lambda_0 + \lambda_1 t) &= t^2 - 2t + \phi_1^2 t + 1, \\ (\mu_0 + \mu_1 t) + (\lambda_0 + \lambda_1 t) &= -(t+1)\phi_1. \end{cases}$$

Equating the coefficients of the two polynomials in  $t$  we find that there are no solutions for  $\lambda_0, \lambda_1, \mu_0$  and  $\mu_1$  when  $|\phi_1| < 2$ . On the other hand, there are the following solutions when  $|\phi_1| \geq 2$ : let

$$z_1 = \frac{-\phi_1 + \sqrt{\phi_1^2 - 4}}{2} \quad z_2 = \frac{-\phi_1 - \sqrt{\phi_1^2 - 4}}{2} \quad (3.26)$$

be two solutions of the equation  $z^2 + \phi_1 z + 1 = 0$ , then we can choose either  $\mu_0 = \lambda_1 = z_1$ ,  $\lambda_0 = \mu_1 = z_2$  or  $\mu_0 = \lambda_1 = z_2$ ,  $\lambda_0 = \mu_1 = z_1$ . This gives the required expressions for  $a_0(t)$ ,  $a_1(t)$ ,  $b_0(t)$  and  $b_1(t)$ .

(B) Assume  $\phi_2 = 1 - \phi_1$ . Then the equations (3.19) become

$$\begin{aligned} a_1(t)b_1(t) &= -(1 - \phi_1)C_{W-2}(t), \\ a_0(t)b_1(t) + a_1(t)b_0(t) &= -\phi_1(tC_{W-2}(t) + 1) + \phi_1(1 - \phi_1)(C_{W-2}(t) - 1), \\ a_0(t)b_0(t) + 2ta_1(t)b_1(t) &= t^2C_{W-2}(t) + t + 1 + \phi_1^2 t C_{W-2}(t) + (1 - \phi_1)^2(C_{W-2}(t) - (t+1)). \end{aligned} \quad (3.27)$$

Using the expression (3.24) for the products  $a_0(t)b_1(t)$  and  $a_1(t)b_0(t)$ , we obtain  $a_1(t)b_0(t) = (1 - \phi_1)(t+1)C_{W-2}(t)$  as one of the two roots given by (3.24). Similar to the case (A), from the first equation in (3.27) we obtain  $b_0(t) = -(1+t)b_1(t)$ . Take any root  $t_1$  of  $a_1(t)$  and substitute it into the second equation in (3.27). Since the roots of  $a_1(t)$  consist of the roots of  $C_{W-2}(t)$ , we obtain  $\phi_1(2 - \phi_1) = 0$ . The solution  $\phi_1 = 0$  implies  $\phi_2 = 1$  and  $\phi_1 = 2$  gives  $\phi_2 = -1$ . The case  $\phi_1 = 2, \phi_2 = -1$  gives a solution described above in the case (A). In the case  $\phi_1 = 0, \phi_2 = 1$  we have solutions to (3.24) obtained from arbitrary splitting

$$C_{W-2}(t) = p(t)q(t) \quad (3.28)$$

of  $C_{W-2}(t)$  into a product of two non-zero polynomials  $p(t)$  and  $q(t)$  and setting

$$a_0(t) = (1+t)p(t), \quad a_1(t) = p(t), \quad b_0(t) = (1+t)q(t) \quad \text{and} \quad b_1(t) = -q(t). \quad (3.29)$$

(C) Assume  $\phi_2 = 1 + \phi_1$ . The equations (3.19) become

$$\begin{aligned} a_1(t)b_1(t) &= -(1 + \phi_1)C_{W-2}(t), \\ a_0(t)b_1(t) + a_1(t)b_0(t) &= -\phi_1(tC_{W-2}(t) + 1) + \phi_1(1 + \phi_1)(C_{W-2}(t) - 1), \\ a_0(t)b_0(t) + 2ta_1(t)b_1(t) &= t^2C_{W-2}(t) + t + 1 + \phi_1^2tC_{W-2}(t) + \\ &\quad (1 + \phi_1)^2(C_{W-2}(t) - (t + 1)). \end{aligned}$$

One of the two solutions for  $a_1(t)b_0(t)$  is  $a_1(t)b_0(t) = C_{W-2}(t + (\phi_1 + 1)^2)$ . Similarly to the above we obtain  $\phi_1(2 + \phi_1) = 0$ . The solution  $\phi_1 = 0$  gives  $\phi_2 = 1$  and hence the same set of solutions to equations (3.19) as in Case (B). The solution  $\phi_1 = -2$  gives  $\phi_2 = -1$  and is covered in Case (A).  $\square$

### A is symmetric and 5-diagonal, B is diagonal

**Theorem 9.** Let  $W \geq 4$  and the matrix  $\mathbf{W}$  be defined in (3.16). There exist matrices  $\mathbf{A} = (a_{i,j})_{i,j=0}^A$  and  $\mathbf{B} = (b_{i,j})_{i,j=0}^B$  with  $A \geq 2$  and  $B \geq 2$ , where  $\mathbf{A}$  is 5-diagonal and  $\mathbf{B}$  is diagonal, such that  $\mathbf{W} = \mathbf{A} * \mathbf{B}$  if and only if either (a)  $\phi_2 = -1$  or (b)  $\phi_1 = 0$  and  $\phi_2 = 1$ .

*Proof.* Assume that  $\mathbf{W} = \mathbf{A} * \mathbf{B}$ . From (3.11) we obtain

$$w_2(t) = a_2(t)b_0(t), \quad w_1(t) = a_1(t)b_0(t), \quad w_0(t) = a_0(t)b_0(t), \quad \forall t \in \mathbb{C}. \quad (3.30)$$

Since  $\phi_2 \neq 0$ , from the left equalities in (3.19) and (3.30) we deduce that all the roots of  $C_{W-2}(t)$  are the roots of  $a_2(t)b_0(t)$ . Since by assumption  $B \geq 2$ , at least two of the roots of  $C_{W-2}(t)$  are the roots of  $b_0(t)$ . Let  $t_1$  and  $t_2$  be two of these roots.

Suppose that the conditions on  $\phi_1$  and  $\phi_2$  are not fulfilled. From the second equality in (3.19),  $w_1(t_i) = 0$  but the second equality in (3.30) yields  $w_1(t_i) = -\phi_1(1 + \phi_2)$ ,  $i = 1, 2$ . From the third equality in (3.19)  $w_0(t_i) = 0$  but the third equality in (3.30) yields  $w_0(t_i) = (1 - \phi_2^2)(t_i + 1)$ ,  $i = 1, 2$ . This is possible only in the cases (a)  $\phi_2 = -1$  and (b)  $\phi_1 = 0$ ,  $\phi_2 = 1$ . This contradiction proves the necessity of the conditions on  $\phi_1$  and  $\phi_2$ .

Assume (a):  $\phi_2 = -1$ . Then

$$w_2(t) = C_{W-2}(t), \quad w_1(t) = -\phi_1(C_{W-2}(t)(t + 1)), \quad w_0(t) = C_{W-2}(t)(t^2 + \phi_1^2t + 1).$$

Therefore,

$$\begin{aligned} a_0(t)b_0(t) &= C_{W-2}(t)(t^2 + \phi_1^2t + 1), \\ a_1(t)b_0(t) &= -\phi_1(C_{W-2}(t)(t + 1)), \\ a_2(t)b_0(t) &= C_{W-2}(t). \end{aligned}$$

Let  $C_{W-2}(t) = p(t)q(t)$ , where  $p(t)$  is a non-zero polynomial of any degree including 0 and  $q(t)$  is a

polynomial of degree at least 2. Then we can choose

$$\mathbf{b}_0(t) = \mathbf{q}(t), \mathbf{a}_2(t) = \mathbf{p}(t), \mathbf{a}_1(t) = -\phi_1(t+1)\mathbf{p}(t) \text{ and } \mathbf{a}_0(t) = (t^2 + \phi_1^2 t + 1)\mathbf{p}(t). \quad (3.31)$$

Now assume (b):  $\phi_1 = 0, \phi_2 = 1$ . Then

$$\mathbf{w}_2(t) = -\mathbf{C}_{W-2}(t), \mathbf{w}_1(t) = 0, \mathbf{w}_0(t) = (t^2 + 1)\mathbf{C}_{W-2}(t).$$

Therefore,

$$\mathbf{a}_0(t)\mathbf{b}_0(t) = (t^2 + 1)\mathbf{C}_{W-2}(t), \mathbf{a}_1(t)\mathbf{b}_0(t) = 0, \mathbf{a}_2(t)\mathbf{b}_0(t) = -\mathbf{C}_{W-2}(t).$$

Again, let  $\mathbf{C}_{W-2}(t) = \mathbf{p}(t)\mathbf{q}(t)$  with the same assumptions on  $\mathbf{p}(t)$  and  $\mathbf{q}(t)$ . Then we can choose  $\mathbf{b}_0(t) = \mathbf{q}(t), \mathbf{a}_2(t) = -\mathbf{p}(t), \mathbf{a}_1(t) = 0$  and  $\mathbf{a}_0(t) = (t^2 + 1)\mathbf{p}(t)$ .  $\square$

**Remark 27.** For technical reason, Theorem 9 does not cover the case  $B = 1$ . This case can be treated separately as follows. First, similarly to the discussion in Section 3.2.5, the polynomial  $\mathbf{C}_{W-2}(t)$  is divisible by a polynomial  $\mathbf{q}(t) = \mathbf{b}_0(t)$  of degree 1 with real coefficients if and only if  $W$  is odd; this polynomial is  $\mathbf{b}_0(t) = \lambda(1+t)$  with  $\lambda \neq 0$ . Consider, for odd  $W$ , the equalities (3.19) and (3.30). The first and third equalities in (3.19) show that  $\mathbf{a}_0(t) = \mathbf{w}_0(t)/\mathbf{b}_0(t)$  and  $\mathbf{a}_2(t) = \mathbf{w}_2(t)/\mathbf{b}_0(t)$  are polynomials for arbitrary values of  $\phi_1$  and  $\phi_2$ . However, the second equality in (3.19) implies that  $\mathbf{a}_1(t) = \mathbf{w}_1(t)/\mathbf{b}_0(t)$  is a polynomial if and only if  $\mathbf{w}_1(-1) = 0$ , that is, either  $\phi_2 = -1$  or  $\phi_1 = 0$  (in the latter case,  $\mathbf{a}_1(t) = 0$ ).

**Remark 28.** The main steps in the proofs of Theorems 7 and 9 is calculation of remainders of division of  $\mathbf{w}_k(t)$  by  $\mathbf{C}_{W-p}(t)$ , where  $p = 1$  and  $k = 0$  for AR(1) and  $p = 2$  and  $k = 0, 1$  for AR(2). Indeed, denote such a remainder as  $\mathbf{r}_k(t)$  and  $B$  roots of  $\mathbf{b}_0(t)$ , which are also the roots of  $\mathbf{C}_{W-p}(t)$ , as  $t_1, \dots, t_p$  (these roots exist since  $B \geq p$ ). Then  $\mathbf{w}_k(t_i) = \mathbf{r}_k(t_i)$  for any  $i = 1, \dots, p$ . If  $p$  is larger than the degrees of the remainders  $\mathbf{r}_k(t)$  for each  $k$ , then  $\{t_i\}_{i=1}^p$  are the roots of  $\mathbf{w}_k(t)$  if and only if the remainders  $\mathbf{r}_k(t_i)$  are zero. This is the case of Theorem 7 and Theorem 9.

**Remark 29.** Theorems 7, 8 and 9 show that for stationary AR(1) and AR(2) models the deconvolution  $\mathbf{W} = \mathbf{A} * \mathbf{B}$  with  $A, B \geq 2$  cannot be performed as the necessary and sufficient conditions in these theorems contradict to the stationarity conditions.

### 3.2.6. Non-existence of the deconvolution $\mathbf{W} = \mathbf{A} * \mathbf{B}$ for a stationary AR( $p$ ) model with general $p$ and diagonal $\mathbf{B}$

Consider the AR( $p$ ) model in the form  $\sum_{i=0}^p \alpha_i \xi_{n-i} = \varepsilon_n$ , where  $D\varepsilon_n = 1$  and  $\alpha_p \neq 0$ . This notation is connected to the basic notation in the form  $\xi_n = \sum_{i=1}^p \phi_i \xi_{n-i} + \varepsilon_n$  by simple relations  $\alpha_0 = 1$  and  $\alpha_i = -\phi_i$  for  $i > 0$ .

Let  $\Sigma$  be the autocovariance matrix of this AR( $p$ ) process. In accordance with [146] and [145, Eq. 10], the matrix  $\mathbf{W} = \Sigma^{-1}$  has the form of a symmetric  $(2p+1)$ -diagonal matrix with

$$w_{i,j} = \begin{cases} \sum_{m=0}^{\min(i,j)} \alpha_m \alpha_{m+|i-j|} & \text{for } \max(i,j) < p; \\ \sum_{m=0}^{p-|i-j|} \alpha_m \alpha_{m+|i-j|} & \text{for } \max(i,j) \geq p, \min(i,j) \leq W-p, |i-j| = 0, \dots, p; \\ \sum_{m=0}^{\min(W-i,W-j)} \alpha_m \alpha_{m+|i-j|} & \text{for } \min(i,j) > W-p; \\ 0, & \text{otherwise,} \end{cases}$$

where  $i, j = 0, \dots, W$ .

One of the stationarity conditions for the AR( $p$ ) model is  $|\phi_p| < 1$ , see e.g. Jury's test of stability [147, Section 3.9]. In this section, we will show that if  $|\phi_p| < 1$  then the deconvolution (3.10) is impossible under the assumption that the matrix  $\mathbf{B}$  is diagonal and has a size larger than  $p \times p$ .

In the same manner as before, we can express the diagonals of  $\mathbf{W}$  through the generating functions  $C_M(t)$  of vectors  $(1, \dots, 1)^T \in \mathbb{R}^{M+1}$  with  $M = W-2p, W-2p+1, \dots, W$ :

$$w_k(t) = \sum_{i=0}^{p-k} \alpha_i \alpha_{i+k} t^i C_{W-2i-k}, \quad k = 0, \dots, p. \quad (3.32)$$

For example,  $w_p(t) = -\phi_p C_{W-p}(t)$ .

The expressions for  $w_k(t)$  in terms of  $C_{W-p}(t)$  play an important role for obtaining the results for AR(1) and AR(2). In particular, Theorems 7 and 9, where either  $\mathbf{A}$  or  $\mathbf{B}$  in the deconvolution  $\mathbf{W} = \mathbf{A} * \mathbf{B}$  is diagonal, are based on the consideration of remainders of division of  $w_k(t)$  by  $C_{W-p}(t)$  for  $k = 0, 1$  and AR(1) and  $k = 0, 1, 2$  and AR(2), see Remark 28.

It follows from (3.32) that  $w_p(t)$  is proportional to  $C_{W-p}(t)$  for any  $p$ . Therefore, it can be proved, similarly to the proofs of Theorems 7 and 9, that a necessary condition for the possibility of deconvolution  $\mathbf{W} = \mathbf{A} * \mathbf{B}$ , where  $\mathbf{B}$  is a diagonal matrix and  $B > p$ , is zero remainders of division of  $w_k(t)$  by  $C_{W-p}(t)$  for  $k = 0, \dots, p-1$ . The proof uses the equalities  $w_k(t) = a_k(t)b_0(t)$ ,  $k \leq p$ , given in Corollary 10.

The following lemma calculates the remainders.

**Lemma 17.** *The remainder  $r_k(t)$  of the division of  $w_k(t)$  by  $C_{W-p}(t)$  for  $k = 0, \dots, p$  is equal to*

$$r_k(t) = \sum_{i=0}^{p-k} \alpha_i \alpha_{i+k} (C_{p-k-i-1}(t) - C_{i-1}(t)). \quad (3.33)$$

(Here we assume that  $C_{-1} = 0$ .)

*Proof.* Directly follows from (3.32) and the equalities  $t^i C_{W-2i-k} = C_{W-i-k} - C_{i-1}$  and  $C_{W-i-k} = t^{p-(i+k)} C_{W-p} + C_{p-k-i-1}$ , which are particular cases of (3.17).  $\square$

**Corollary 11.**

$$r_p(t) = 0, \quad (3.34)$$

$$r_{p-1}(t) = \alpha_0\alpha_{p-1} - \alpha_1\alpha_p, \quad (3.35)$$

$$r_0(t) = \sum_{m=0}^{p-1} \left( \sum_{i=0}^{\min(m, p-m-1)} (\alpha_i^2 - \alpha_{p-i}^2) \right) t^m. \quad (3.36)$$

*Proof.* The equality (3.34) is readily obtained from (3.33), since  $C_{-1} = 0$ . The equality (3.35) is proved by direct substitution of  $k = p - 1$  and  $i = 0, 1$  to (3.33).

For the proof of (3.36), let us substitute  $k = 0$  to the sum (3.33):

$$r_0(t) = \sum_{i=0}^p \alpha_i^2 (C_{p-i-1}(t) - C_{i-1}(t)) \quad (3.37)$$

and then write this sum as a polynomial in  $t$ . By (3.17) we obtain for  $i = 0, \dots, p$

$$C_{p-i-1}(t) - C_{i-1}(t) = \begin{cases} t^i C_{p-2i-1}, & i < p/2, \\ 0, & i = p/2, \text{ } p \text{ is even,} \\ -t^{p-i} C_{2i-p-1}, & i > p/2. \end{cases}$$

Therefore,

$$r_0(t) = \sum_{i=0}^{\lfloor (p-1)/2 \rfloor} \alpha_i^2 t^i C_{p-2i-1} - \sum_{j=\lceil (p+1)/2 \rceil}^p \alpha_j^2 t^{p-j} C_{2j-p-1}.$$

Taking  $j = p - i$ , we obtain

$$r_0(t) = \sum_{i=0}^{\lfloor (p-1)/2 \rfloor} (\alpha_i^2 - \alpha_{p-i}^2) t^i C_{p-2i-1} = \sum_{i=0}^{\lfloor (p-1)/2 \rfloor} (\alpha_i^2 - \alpha_{p-i}^2) \sum_{m=i}^{p-i-1} t^m.$$

The equality (3.36) is derived from this expression by regrouping the terms.  $\square$

**Remark 30.** It can be easily checked that the remainders, which have been calculated in the course of proofs of Theorems 7 and 9, can be deduced from Corollary 11. For AR(1), we have  $r_0(t) = 1 - \phi_1^2 = \alpha_0^2 - \alpha_1^2$ . For AR(2), we have  $r_1(t) = -\phi_1(1 + \phi_2) = \alpha_0\alpha_1 - \alpha_1\alpha_2$  and  $r_0(t) = (1 - \phi_2^2)(t + 1) = (\alpha_0^2 - \alpha_2^2)(t + 1)$ .

It follows from Corollary 11 that one of the necessary conditions of existence of the decomposition  $\mathbf{W} = \mathbf{A} * \mathbf{B}$  for the case of a diagonal  $\mathbf{B}$  with  $B \geq p$  is  $\phi_p = \pm 1$  (the coefficient in front of  $t^{p-1}$  in (3.36) is equal to  $\alpha_0^2 - \alpha_p^2 = 1 - \phi_p^2$  and should be equal to 0). This contradicts to  $|\phi_p| < 1$ , a necessary condition for the stationarity of AR( $p$ ). Therefore, for the matrix  $\mathbf{W}$  corresponding to a stationary AR( $p$ ) model, the deconvolution (3.10) cannot be performed.

### 3.3. Detection of signals by Monte Carlo singular spectrum analysis: Multiple testing

Here we consider a specific problem of detection of a signal (e.g., a periodic component) in a noisy time series following [25].

Generally, SSA is a model-free method. However, a considerable part of the SSA theory is devoted to the extraction of time series components, which are governed (maybe, approximately) by a linear recurrence relation [30]; in particular, a sum of sine waves with slowly-varying amplitudes belong to this class of signals. SSA extracts periodic components well; however, it is well-known that the extracted components can be spurious, since they can be produced by noise. In a sense, this is a payment for the nonparametric nature of the method. If we want to apply the statistical approach for judging, a model should be assumed. Usually, the question about the existence of a signal in the time series is formulated as testing the hypothesis that the series is a stochastic process. The criterion should be powerful against the alternative hypothesis, which states the existence of a non-random (e.g. periodic) component. There are a lot of statistical criteria for testing these hypotheses for different classes of stochastic processes (see e.g. [148] for red noise or [149] for an Ornstein-Uhlenbeck state space, which is a continuous analogue of the first-order autoregressive processes AR(1)). We consider the construction of such a criterion in the framework of SSA, since SSA is able to reconstruct the detected time series component.

It follows from the properties of SSA (see e.g. the description of the relation between the spectral density and the eigenvectors in [1, Sect. 6.4.3], where the results from [150] in terms of SSA are discussed) that a natural assumption for the noise model is that noise is red (the AR(1) with a positive coefficient). This is because the spectral density of red noise is monotonic; therefore, the eigenvectors are similar to sinusoids and can be connected to different frequencies. Recall that  $\xi = (\xi_1, \dots, \xi_N)$  is red noise with parameters  $\varphi$  and  $\delta$  if  $\xi_n = \varphi \xi_{n-1} + \delta \varepsilon_n$ , where  $0 < \varphi < 1$ ,  $\varepsilon_n$  is white Gaussian noise with mean 0 and variance 1 and  $\xi_1$  has a normal distribution with mean zero and variance  $\delta^2/(1 - \varphi^2)$ . Moreover, in climatology, the common model is a weak signal (if any) in red noise. Both considerations inspired the creation of the Monte Carlo SSA method (MC-SSA). MC-SSA was proposed in [151] and later was considered in many papers ([152], [153], [154], [155], [156], [157], among others).

However, the terminology in these papers differs from the standard statistical terminology and therefore it is very important to bridge the gap between the applied and statistical approaches. Moreover, this can help to avoid wrong conclusions in real-life applications. Investigation of criterion properties (and comparison with other tests) will be performed by the estimation of the type I and type II errors.

Let us briefly described how the basic MC-SSA test is constructed. At Decomposition step of SSA, each decomposition component (related to the eigenvectors of the sample autocovariance matrix, in the basic version) can be put into correspondence with a frequency, since the eigenvectors of the autocovariance matrix of red noise, which is a Kac–Murdock–Szegő Toeplitz matrix, are sine

waves with almost equidistant frequencies [158]. The eigenvalues are equal to the total squared norms of projection of the trajectory matrix columns on the eigenvectors and therefore reflect contributions of the decomposition components (and therefore of the corresponding frequencies) into the time series. The original idea of MC-SSA is to estimate the parameters of red noise and apply the bootstrap simulations to construct prediction intervals for eigenvalues in the case when there is no signal. If an eigenvalue of the time series is beyond the constructed prediction intervals, the corresponding eigenvector frequency is considered significant. Moreover, it is possible to reconstruct the detected signal. In the modifications of MC-SSA, the choice of vectors for projection can vary.

It is clear from the method description that there is the problem of multiple testing when the probability of the false detection of a periodic component for one of the considered frequencies (family-wise error rate) is unknown and is much larger than the given significance level (single-test error rate). This problem is discussed in different papers devoted to MC-SSA. The theoretical approach to multiple testing, which we propose in this section, allows constructing the multivariate criterion with the given family-wise error rate.

The novelty is applying the statistical approach to the signal-detection problem in the framework of Monte Carlo SSA to control the type I error and estimate the type II error. For simultaneous testing of the presence of multiple frequencies, a multiple version of MC-SSA is proposed to control the family-wise error rate. Basing on the numerical study, we discuss pluses and minuses of several versions of MC-SSA.

### 3.3.1. Statistical approach to hypothesis testing

In papers starting from [151], the method MC-SSA is described mostly as a method for applied problems and therefore the way of description is not conventional for statisticians. Therefore, let us start with a statistical approach to the problem.

Let the null hypothesis be that the time series is a pure stationary stochastic process. In the considered context, it can be white or red noise. Sometimes, one says that the presence of a signal in noise is tested, whereas the null hypothesis is formulated as the hypothesis about the absence of a signal in noise. Consider a criterion, which determines if the null hypothesis is rejected or is not rejected. If the null hypothesis is rejected at the given significance level  $\alpha$ , then one can claim the presence of a signal (more precisely, a deviation from the null hypothesis). The probability to reject the null hypothesis if it is true is called type I error ( $\alpha_I$ ). If a criterion is correct, then the type I error is equal to the given significance level (or at least not larger than  $\alpha$ ). Different criteria differ by the power against an alternative hypothesis. The power is the probability to reject the null hypothesis if the alternative hypothesis is true. The alternative hypothesis that the time series contains a periodic component is important in practice; therefore, we will consider criteria, which are powerful against such hypotheses.

If we want to make a false discovery rarely, then we choose a small significance level  $\alpha$  and

this guarantees that we will reject the true null hypothesis with probability not larger than  $\alpha$ . However, we can not choose a very small significance level, since the test power decreases as  $\alpha$  decreases.

Note that it is not permitted to consider a criterion if its type I error exceeds the given significance level (a *liberal* criterion). Therefore, before a comparison of criteria by power, one should be sure that the type I error lies in the given range. If the type I error is less than the significance level (a *conservative* criterion), this is admissible; however, this means that this criterion can be improved, that is, the power can be increased by a correction to obtain the type I error closer to the significance level.

A useful characteristic of a criterion is the possibility to interpret the difference from the null hypothesis if this hypothesis is rejected.

## Bootstrapping

Most of the criteria have the following form. A constructed test statistic measures the difference between data and the null hypothesis in some way. There is a threshold such that if the test statistic is larger than the threshold, the null hypothesis is rejected. Certainly, this threshold depends on the significance level  $\alpha$ . It is not uncommon that this threshold can not be obtained theoretically. Then simulations are used. Surrogate data are simulated according to the null hypothesis and the test statistic is calculated many times ( $G$ ) to determine the threshold; this approach is widely used, see e.g. [159]. Consequently, the threshold, for which the null hypothesis is rejected approximately  $\alpha G$  times, is found. The estimated threshold is used for testing the hypothesis for real-life data. The surrogate data should be obtained exactly in the same way as the test statistic was generated for the original data. The described approach can be called Monte-Carlo one. This approach helps to construct the criterion with the type I error tending to  $\alpha$  as  $G$  tends to infinity.

However, the Monte Carlo approach can be applied if the null hypothesis fully determines the data. For example, the null hypothesis states that the time series is red noise with variance  $\delta^2$  and coefficient  $\varphi$ , where  $\delta^2$  and  $\varphi$  are known numbers.

Unfortunately, this is not the case in practice. Therefore, the so-called bootstrapping is used (“pull yourself up by your bootstraps”). If  $\delta^2$  and  $\varphi$  are unknown, then they are estimated with the help of the real-life data under study and then the surrogate data are produced by simulations with the estimated parameter. Since the estimated parameters differ from the true (unknown) parameters, then the type I error can generally be far from  $\alpha$  and the test can become liberal or conservative. Thus, Monte-Carlo SSA (this is the name of a family of concrete algorithms) is a kind of testing with bootstrapping.

## Estimation of type I error and power

The above considerations about the relation between the type I error, the significance level  $\alpha$  and the level of power can not be applied in practice, since the type I error and the power are unknown. If something is unknown theoretically, then simulation helps again.

Let a criterion be constructed to make the decision (reject or not reject) for a given significance level  $\alpha$ . It can be constructed theoretically or with bootstrapping/simulations within, it does not matter. Then the sample data with given parameters according to the null hypothesis (this is the Monte-Carlo approach) are simulated many times ( $M$ ). Then the proportion of cases with the rejection of the null hypothesis is an estimate of the type I error.

To estimate power, we should set an alternative hypothesis. There is freedom in the choice. The common rule is to include into the alternative hypothesis the case, which is important, that is, the case that should be distinguished from the null hypothesis. For example, the alternative can state that the time series is a sine wave with a given frequency, amplitude and phase corrupted by noise with the same parameters as were considered for the null hypothesis.

For power estimation, the procedure similar to that for the type-I error is fulfilled. We simulate surrogate data with the given parameters according to the alternative hypothesis many times ( $M$ ). Then the proportion of cases with the rejection of the null hypothesis is the estimate of the power.

## Prediction intervals for testing hypotheses

Let a test statistic  $t$  provide an interpreted characteristic of data (e.g. a contribution of a frequency to the observed time series). Thus, the question about the presence of a signal can be reformulated as “can this value of  $t$  be caused by the noise component only?”.

The answer to the question can be obtained in the standard way. Since the contribution  $t$  is random, there is a prediction interval for it. The prediction interval can be constructed by simulation. If we generated a sample of possible contributions, then the 95% prediction interval is the interval between the 2.5% and 97.5% quantiles of this sample. In statistical terminology, this interval is not called confidence, since confidence intervals are constructed for unknown (non-random) parameters and their length would tend to zero as the number of simulations (sample size) tends to infinity.

$\gamma$ -Prediction intervals serve for testing hypotheses with a significance level  $\alpha$  for  $\alpha = 1 - \gamma$ . If the observed value of  $t$  does not belong to the prediction interval, the null hypothesis is rejected. It is convenient to depict, say, 95%-prediction intervals for  $t$  to visualize the hypothesis testing with the significance level 5%.

## One-tailed and two-tailed criteria

We mentioned that a criterion consists of a test statistic  $t$  and a threshold  $t_0$ . The use of this threshold can be different. Moreover, the threshold can consist of two numbers,  $t_1$  and  $t_2$ . For

example, the null hypothesis can be rejected if the test statistic is larger than  $t_1$  or smaller than  $t_2$  (two-tailed test) or if  $t > t_0$  (one-tailed test). The choice of the criterion type depends on the alternative hypothesis, since we want to increase the power against the chosen alternative.

If we want to detect both cases, when the contribution of a frequency is either larger or smaller than that for pure noise, then we choose a two-tailed test. If we want to detect only the excess of the frequency contribution, we choose a one-tailed test. This approach can be expressed in terms of prediction intervals. If we are interested to find the frequency with contribution larger than that of noise, then the one-sided prediction interval has the form  $[0, t_0]$  (in the general case,  $[-\infty, t_0]$ ; however, in our case the test statistic is non-negative). In the two-tailed case, the two-sided prediction interval is  $[t_1, t_2]$ .

## Multiple testing

The problem of multiple testing is well-known. The approach to the statistical testing described above is applicable for single tests only, since the probability of false discovery is controlled for each individual test only.

If we test several tests ( $m$ ) simultaneously, we are interested in the so-called family-wise error rate (FWER). FWER is the probability of false discovery in at least one of  $m$  tests. This probability can be vastly larger than the chosen small  $\alpha$ . Thus, we should not use a set of single tests with  $\alpha$  if we want to control FWER. Ideally, we should construct one multivariate test instead of several single tests. If this is technically hard, the Bonferroni correction is used (performing single tests with significance level  $\alpha/m$ ); this trick controls the FWER not larger than  $\alpha$  (usually, this does the family-wise type I error considerably smaller than  $\alpha$ ; that is, the multiple testing is conservative and therefore decreases the test power). If the single tests are independent, the Šidák correction (performing single tests with significance level  $1 - (1 - \alpha)^{1/m}$ ) provides the exact test; however, in the general case, testing with the Šidák correction can be liberal and therefore the Šidák correction is not applied.

### 3.3.2. Monte Carlo SSA

#### General comments

On the one hand, Monte Carlo SSA is a well-developed method. In [151], the foundation of the method is thoroughly described. On the other hand, many questions are still under investigation; they are, among others, the best way of estimating the noise parameters and taking into consideration the presence of the nuisance signal. In this section, we consider a most simple version of Monte Carlo SSA to concentrate on the problem of multiple testing.

In some cases, we will assume that the noise parameters are known. Also, we will show how can the criterion errors change if the parameters are estimated.

Recall that in SSA at the first stage, among other things, a basis is constructed from the left singular vectors  $\{U_i\}$  of the trajectory space of the series. Vectors from this basis are often used

in Monte Carlo SSA to estimate the contribution of the frequency corresponding to the vector. Sometimes a version of Toeplitz SSA for stationary time series is considered instead of Basic SSA. By the properties of the SVD,  $\lambda_i = \|\Pi_{\{U_i\}} \mathbf{X}\|^2 = \|\mathbf{X}^T U_i\|^2 = \sum_{j=1}^K (X_j^T U_i)^2$  can be interpreted as the total squared norm of projections of lagged vectors to  $\text{span}(U_i)$ .

There is a natural modification of Monte Carlo SSA, which does not use the SSA decomposition step except for the construction of the trajectory matrix (a part of the study will be performed for such a modification). However, the relation with SSA is essential, since the detected signal can be reconstructed through SSA if we keep the connection with the eigenvectors in SSA.

### Single test

Let  $\boldsymbol{\xi} = (\xi_1, \dots, \xi_N)$  be a red noise with parameters  $\boldsymbol{\varphi}$  and  $\boldsymbol{\delta}$ . Denote by  $L$  the window length and by  $\boldsymbol{\Xi} = \mathcal{T}(\boldsymbol{\xi})$  the trajectory matrix of  $\boldsymbol{\xi}$ . Let a vector  $\mathbf{W} \in \mathbb{R}^L$ ,  $\|\mathbf{W}\| = 1$ , be given. If we are interesting in a frequency contribution, then  $\mathbf{W}$  can be a sine wave with a given frequency. The total squared norm of the projection of the columns of  $\boldsymbol{\Xi}$  to the vector  $\mathbf{W}$  is calculated as

$$p = \|\boldsymbol{\Xi}^T \mathbf{W}\|^2 = \mathbf{W}^T (\boldsymbol{\Xi} \boldsymbol{\Xi}^T) \mathbf{W}.$$

The null hypothesis states that the observed time series  $\mathbf{X}$  is a realization of  $\boldsymbol{\xi}$  with some parameters  $\boldsymbol{\varphi}$  and  $\boldsymbol{\delta}$ . Denote  $\hat{p} = \|\mathbf{X}^T \mathbf{W}\|^2$ . If  $\mathbf{W}$  is an eigenvector of  $\mathbf{X} \mathbf{X}^T$ , then  $\hat{p}$  is the corresponding eigenvalue. Note that for a sinusoidal  $\mathbf{W}$ ,  $\hat{p}$  negligibly depends on the phase of this sinusoid, since for large  $K = N - L + 1$  the lagged vectors consist of many shifts.

Let  $\boldsymbol{\varphi}$  and  $\boldsymbol{\delta}$  be known. Under some assumptions, the distribution of  $p$  can be calculated theoretically. Then the prediction interval with confidence level  $\gamma$  is calculated as the interval between  $(1 - \gamma)/2$ - and  $(1 + \gamma)/2$ - quantiles for the two-tailed test and between zero and  $\gamma$ -quantile for the one-tailed test (upper-tailed test). In both cases,  $\hat{p}$  belongs to the constructed predicted interval with probability  $\gamma$ .

If the theoretical distribution is unknown, then these quantiles can be calculated by simulation of  $G$  samples  $\boldsymbol{\xi}^i$  of the random vector  $\boldsymbol{\xi}$  and the use of empirical (sample) quantiles for the obtained sample  $p_i = \|\boldsymbol{\Xi}_i^T \mathbf{W}\|^2$ ,  $i = 1, \dots, G$ . The probability that  $\hat{p}$  belongs to the empirical (Monte Carlo) prediction interval tends to  $\gamma$  as  $G$  tends to infinity.

Recall that the significance level  $\alpha$  is equal to  $1 - \gamma$  and therefore one can say that the probability of the type I error  $\alpha_I$  tends to  $\alpha$ .

For both theoretical considerations and simulations, the values of the parameters  $\boldsymbol{\varphi}$  and  $\boldsymbol{\delta}$  are used. Here we do not discuss the estimation of noise parameters in the presence of a signal. Note that the proposed approach to hypothesis testing is valid for various modifications of the MC-SSA technique.

**Remark 31.** *It is important to note that  $\mathbf{W}$  may be produced by the time series itself. However, as was discussed in [151], then the surrogate data should be projected also to the vectors produced*

by them. There is a temptation to consider the version when the projection vector is produced by the observed time series and the test is constructed by projection to this vector. However, then the type-I error is not controlled and the test is liberal.

## Choice of vectors for projection

In practice, we do not know the frequency of a possible periodic signal component. Therefore, the approach is to consider many vectors for projection, which correspond to a set of frequencies. For example, one can take a set of sine waves  $W_1, \dots, W_H \in \mathbb{R}^L$  with equidistant frequencies from some frequency interval  $[\omega_1, \omega_2] \subset (0, 0.5)$ . To obtain slightly dependent tests, the number of vectors should not exceed their dimension  $L$ .

The other choice is to take the set of eigenvectors produced by SSA (this is a common case in MC-SSA) with consideration to Remark 31.

The compromised version suggested in [151] is  $W_1, \dots, W_H$  to be eigenvectors of the general-population covariance matrix. That is, the sample covariance matrix is  $\mathbf{X}\mathbf{X}^T$ , where  $\mathbf{X}$  is the  $L$ -trajectory matrix, whereas the general-population correlation matrix of red noise has the  $(i, j)$ -th term  $\phi^{|i-j|}$ . They are close; however, the difference from the viewpoint of type-I error can be drastic. If one wants to include to the set of projection vectors the sine wave vector with a specific frequency  $\omega$ , then the eigenvectors of the matrix with the  $(i, j)$ -th term  $\phi^{|i-j|} + C\cos(2\pi\omega|i-j|)$  can be considered.

Thus, we will consider two choices:

1.  $W_1, \dots, W_H$  are the eigenvectors of the matrix

$$\{\phi^{|i-j|} + C\cos(2\pi\omega|i-j|)\}_{i,j=1}^L. \quad (3.38)$$

2.  $W_1, \dots, W_H$  are the eigenvectors of the matrix  $\mathbf{X}\mathbf{X}^T$ .

Since we study the version of Monte Carlo SSA, where the projections of surrogate data are performed on the fixed vectors, the second version is generally wrong. We will consider an example to show this.

## Multiple testing

In Monte Carlo SSA, the prediction intervals are constructed for the contribution of each projection vector independently. Let  $W_1, \dots, W_H$  be a set of projection vectors. Denote

$$\hat{p}_k = \|\mathbf{X}^T W_k\|^2, \quad k = 1, \dots, H.$$

For each vector  $W_k$ , the sample of squared projection norms is constructed:  $P_k = (p_{k1}, \dots, p_{kG})^T$ , where  $p_{ki}$  is calculated as

$$p_{ki} = \|\boldsymbol{\Xi}_i^T W_k\|^2, \quad i = 1, \dots, G; \quad (3.39)$$

here  $\mathbf{\Xi}_i = \mathcal{T}(\boldsymbol{\xi}^i)$  is the trajectory matrix of the  $i$ th sample of red noise  $\boldsymbol{\xi}^i = (\xi_1^i, \dots, \xi_N^i)$  with given parameters.

We can construct single prediction intervals for the contribution of each vector  $W_k$  as it is described in Section 3.3.2. The problem of multiple testing (the problem of FWER, which can be much larger than the given significance level  $\alpha$ ) can be solved by means of the Bonferroni correction. However, if the Bonferroni correction is used, the test becomes conservative (FWER is less than  $\alpha$ ). To obtain an exact test, an approach similar to that of Tukey's HSD applied to post hoc comparisons in ANOVA can be considered. That is, we can construct a test, which is based on the distribution of the maximum of the standardized contributions  $p_k$ . If this test rejects the null hypothesis, then all frequencies, which correspond to the projection vectors with the contribution lying outside the corrected prediction intervals, are considered significant. Thus, we can talk about the significance of a frequency if put into correspondence projection vectors and frequencies. Note that for this approach, FWER is equal to  $\alpha$ .

Let us describe the algorithm of constructing the prediction intervals with correction for multiple testing.

The first version is straightforward; this is the single test for each vector  $W_k$  with Bonferroni correction, that is, the significance level  $\alpha/H$  is taken instead of  $\alpha$ .

**ALGORITHM 3.4:** Single one-tailed test with Bonferroni correction

- 1: For each  $k$ ,  $k = 1, \dots, H$ , calculate the test statistic  $\hat{p}_k$  and the sample  $P_k = \{p_{ki}\}_{i=1}^G$ , see (3.39).
- 2: Find  $q_k$  as the sample  $(1 - \alpha/H)$ -quantile of  $P_k$ .
- 3: The null hypothesis, which states that the time series is pure red noise, is not rejected if for each  $k$  the inequality  $\hat{p}_k < q_k$  is valid; otherwise, the null hypothesis is rejected and a signal is detected.
- 4: If  $H_0$  is rejected, then post-hoc testing can be performed: the contribution of  $W_k$  (and of the corresponding frequency) is significant if  $\hat{p}_k$  exceeds  $q_k$ . Thus,  $[0, q_k]$  are considered as the corrected prediction intervals,  $k = 1, \dots, H$ .

**ALGORITHM 3.5:** Single two-tailed test with Bonferroni correction

- 1: For each  $k$ ,  $k = 1, \dots, H$ , calculate the test statistic  $\hat{p}_k$  and the sample  $P_k = \{p_{ki}\}_{i=1}^G$ , see (3.39).
- 2: Find  $q_k^{\text{low}}$  and  $q_k^{\text{up}}$  as the sample  $(0.5\alpha/H)$ - and  $(1 - 0.5\alpha/H)$ -quantiles correspondingly.
- 3: The null hypothesis, which states that the time series is pure red noise, is not rejected if for each  $k$   $q_k^{\text{low}} < \hat{p}_k < q_k^{\text{up}}$ ; otherwise, the null hypothesis is rejected and a signal is detected.
- 4: If  $H_0$  is rejected, then post-hoc testing can be performed: the contribution of  $W_k$  (and of the corresponding frequency) is significant if  $\hat{p}_k$  does not belong  $[q_k^{\text{low}}, q_k^{\text{up}}]$ . Thus,  $[q_k^{\text{low}}, q_k^{\text{up}}]$  are

considered as the corrected prediction intervals,  $k = 1, \dots, H$ .

---

Let us extend the multiple-testing approach, which was considered in [160] in the framework of Monte Carlo SSA. The approach is based on the distribution of  $\max_{1 \leq k \leq H} (p_{ki} - \mu_k) / \sigma_k$ , where  $\mu_k$  and  $\sigma_k$  are mean and standard deviation of  $P_k$ ,  $k = 1, \dots, H$ . Here  $\sigma_k$  reflects the size of the  $k$ -th prediction intervals. Normalization by  $\sigma_k$  keeps the same difference between the intervals sizes as in single tests. By construction, the size of a prediction interval is related to the test power against the existence of the signal at the corresponding frequency. Therefore, we can consider  $\tilde{\sigma}_k = w_k \sigma_k$ , where  $w_k$  is a weight; an increase of the weight corresponds to more expectation of a signal at the frequency matched to  $W_k$ . Some weights  $w_k$  can be zero; this means that the corresponding projection vectors do not participate in the testing and the test is disabled to detect the signal at the corresponding frequencies. Thus, it is the same, to set zero weights outside the frequency range (and unit weights within it) or just consider the subset of the projection vectors matched to the frequency interval. However, the approach with weights is much more flexible and allows to take arbitrary weights between 0 and 1.

---



---

ALGORITHM 3.6: Weighted multiple one-tailed test

---

- 1: For each  $k$ ,  $k = 1, \dots, H$ , calculate the test statistic  $\hat{p}_k$ , the sample  $P_k = \{p_{ki}\}_{i=1}^G$ , see (3.39), and calculate its mean  $\mu_k$  and standard deviation  $\sigma_k$ .
- 2: Calculate  $\boldsymbol{\eta} = (\eta_1, \dots, \eta_G)$ , where

$$\eta_i = \max_{1 \leq k \leq H} (p_{ki} - \mu_k) / \tilde{\sigma}_k, \quad \tilde{\sigma}_k = w_k \sigma_k, \quad i = 1, \dots, G.$$

- 3: Find  $q$  as the sample  $(1 - \alpha)$ -quantile of  $\boldsymbol{\eta}$ .
- 4: The null hypothesis, which states that the time series is pure red noise, is not rejected if

$$\max_{1 \leq k \leq H} (\hat{p}_k - \mu_k) / \tilde{\sigma}_k < q;$$

otherwise, the null hypothesis is rejected and a signal is detected.

- 5: If  $H_0$  is rejected, then post-hoc testing can be performed: the contribution of  $W_k$  (and of the corresponding frequency) is significant if  $\hat{p}_k$  exceeds  $\mu_k + qw_k \sigma_k$ . Thus,  $[0, \mu_k + qw_k \sigma_k]$  are considered as the corrected prediction intervals,  $k = 1, \dots, H$ .
- 

---

ALGORITHM 3.7: Weighted multiple two-tailed test

---

- 1: For each  $k$ ,  $k = 1, \dots, H$ , calculate the test statistic  $\hat{p}_k$ , the sample  $P_k = \{p_{ki}\}_{i=1}^G$ , see (3.39), and calculate its mean  $\mu_k$  and standard deviation  $\sigma_k$ .

2: Calculate  $\boldsymbol{\eta} = (\eta_1, \dots, \eta_G)$ , where

$$\eta_i = \max_{1 \leq k \leq H} |p_{ki} - \mu_k| / \tilde{\sigma}_k, \quad \tilde{\sigma}_k = w_k \sigma_k, \quad i = 1, \dots, G.$$

3: Find  $q$  as the sample  $(1 - \alpha)$ -quantile of  $\boldsymbol{\eta}$ .

4: The null hypothesis, which states that the time series is pure red noise, is not rejected if

$$\max_{1 \leq k \leq H} |\hat{p}_k - \mu_k| / \tilde{\sigma}_k < q;$$

otherwise, the null hypothesis is rejected and a signal is detected.

5: If  $H_0$  is rejected, then post-hoc testing can be performed for a fixed  $k$ : the contribution of  $W_k$  (and of the corresponding frequency) is significant if  $|\hat{p}_k - \mu_k| / \tilde{\sigma}_k$  exceeds  $q$ . Thus,  $[\mu_k - qw_k \sigma_k, \mu_k + qw_k \sigma_k]$  are considered as the corrected prediction intervals,  $k = 1, \dots, H$ .

---

Algorithms 3.5 and 3.7 correspond to two-sided prediction intervals (this is the conventional version of Monte Carlo SSA), while Algorithms 3.4 and 3.6 correspond to the one-sided case. This corresponds to hyper-rectangular and half hyper-rectangular prediction regions respectively.

**Remark 32.** *The choice of the vectors  $W_k$ ,  $k = 1, \dots, H$ , was discussed in Section 3.3.2. To increase the criterion power, the number  $H$  of the vectors should be as small as possible, e.g., only vectors with dominant frequencies from a given frequency range can be taken. If the vectors  $W_k$  are sine waves, the choice of vectors with frequencies from the given range is trivial. If  $W_k$  are the eigenvectors, then their dominant frequencies can be calculated by e.g. the ESPRIT method [30, Section 3.1]. This has a little sense for white noise, for which each eigenvector is a mixture of a lot of frequencies. However, for red noise, it has sense, since the eigenvectors correspond to narrow ranges of frequencies. As we mentioned above, increasing the power can be performed by setting the weights.*

### 3.3.3. Numerical investigation

Let us introduce abbreviations for the test versions. We consider 3-symbols abbreviations with an optional information

$$\{M,S,B\}\{T,E\}\{1,2\}[Est]. \quad (3.40)$$

The first letter M (Multiple) corresponds to Algorithms 3.6 and 3.7, S (Single) corresponds to Algorithms 3.4 and 3.5 without Bonferroni correction, while B means using this correction. The second letter is related to the way of generating the projection vectors: T means that the eigenvectors of the theoretical covariance matrix (3.38) of red noise ( $C = 0$ ) are considered; E means that the eigenvectors of the empirical covariance matrix  $\mathbf{X}\mathbf{X}^T$  are used. The last digit means what (one- or two-tailed) test is considered. We consider the method MT1 as the basic one, which

is exact and more powerful.

This abbreviation corresponds to the use of true parameters of red noise and unit weights  $w_k$ . If the noise parameters are estimated, then we will add ‘Est’ after the digit. If another modification is used, this is indicated separately.

R-scripts in [161] contain an implementation of the above algorithms.

Let us demonstrate the results of different versions of Monte Carlo SSA.

### Example

The model of an artificial time series is

$$x_n = A \sin(2\pi\omega n) + \xi_n, \quad (3.41)$$

where  $\xi_n$  is red noise with parameters  $\varphi$  and  $\delta$ ,  $n = 1, \dots, N$ . The case  $A = 0$  corresponds to the null hypothesis and the case  $A > 0$  yields the presence of signal, that is, corresponds to an alternative. Hereafter we consider the AR(1) parameters  $\varphi = 0.7$  and  $\delta = 1$ .

For illustrative examples, we take  $N = 1000$  and the signal parameters  $A = 0.4$  and  $\omega = 0.2$  in  $H_1$ . The parameters of MC-SSA are  $L = 40$  and  $G = 1000$ .

Let us consider a fixed significance level 0.2 (that is, the confidence level equals 0.8). To weaken the dependence on the time series length  $N$ , we consider  $\|\mathbf{X}\|^2 = \sum_{i=1}^N x_i^2/N$ . We used the true parameters of AR(1) for the creation of surrogate data. The continuous curve is the spectral density of AR(1) with the parameters that were used in the simulation. We calculated the dominant frequencies of  $W_k$  by the ESPRIT method with the rank  $r = 2$ .

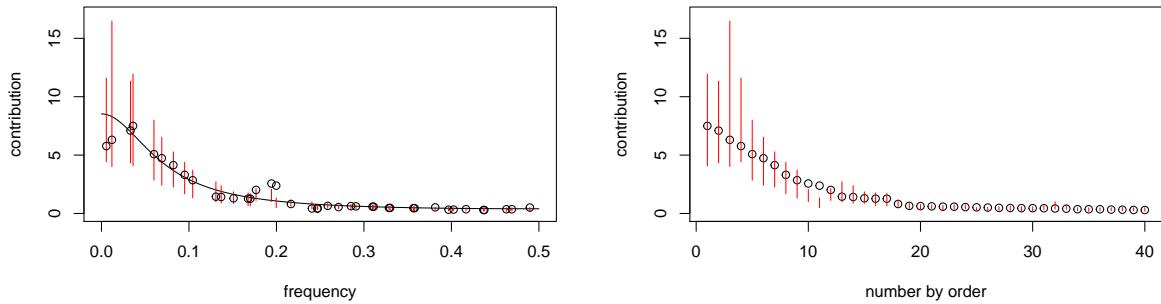


Figure 3.4: ME2. Left — ordered by frequency, right — ordered by projection value ( $\hat{p}_k$ ).

Although one-tailed versions of the tests are more powerful, on most of the figures we show two-sided prediction intervals for a clearer visual presentation.

The presence of a sine-wave signal mostly corresponds to exceeding the upper bound. Therefore, as a rule, for exact tests, if the upper bound is smaller, then the test is more powerful.

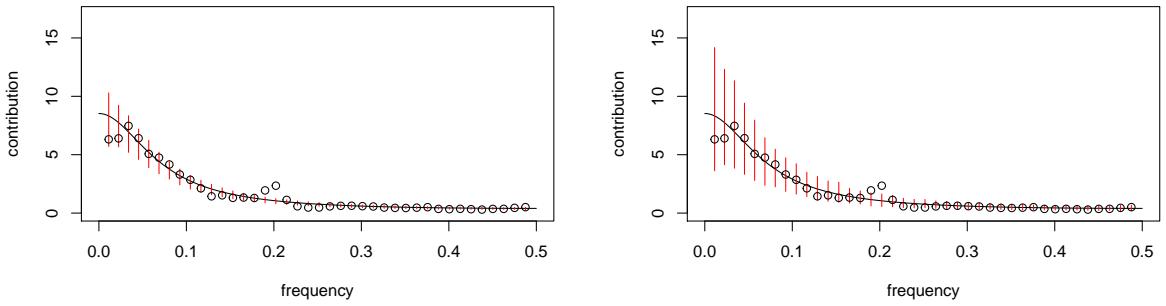


Figure 3.5:  $\{S,B\}T2$ . Left — without correction, right — with Bonferroni correction.

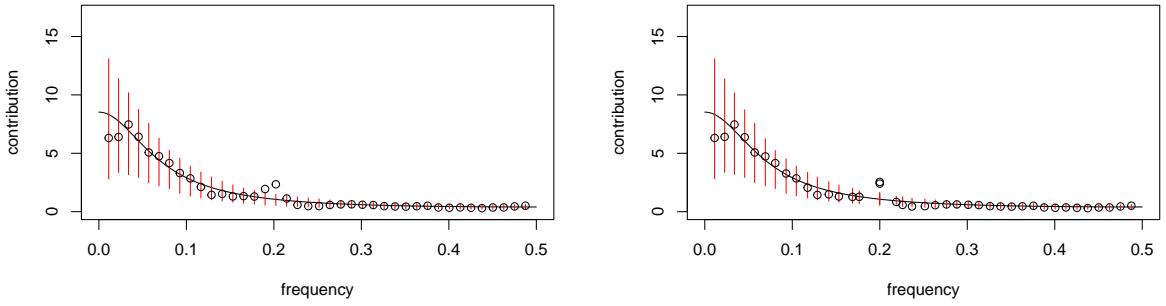


Figure 3.6: MT2. Left — eigenvectors of pure red-noise covariance matrix, right — eigenvectors of the covariance matrix of red-noise with an added sinusoid at frequency 0.2.

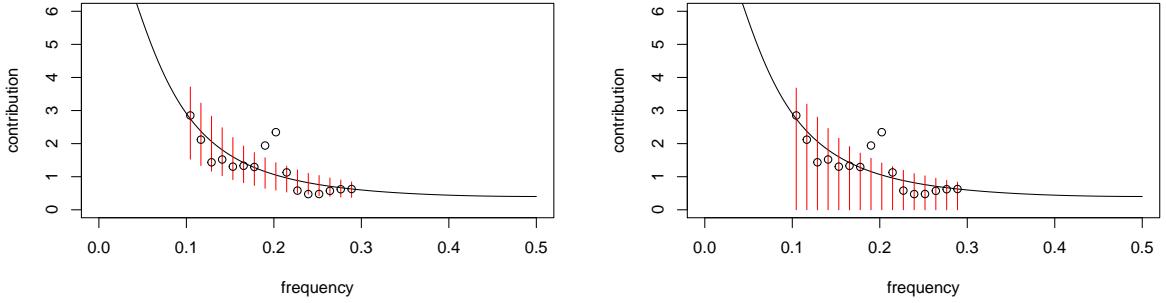


Figure 3.7: MT{1,2}, frequency range [0.1, 0.3]. Left — two-tailed test, right — one-tailed test.

Fig. 3.4 demonstrates generally a liberal test, which therefore cannot be used in practice as is. In this version the projection vectors are generated by the observed time series; this choice corresponds to a popular version of Monte Carlo SSA.

Fig. 3.5 shows prediction intervals for the single test with projection vectors chosen as

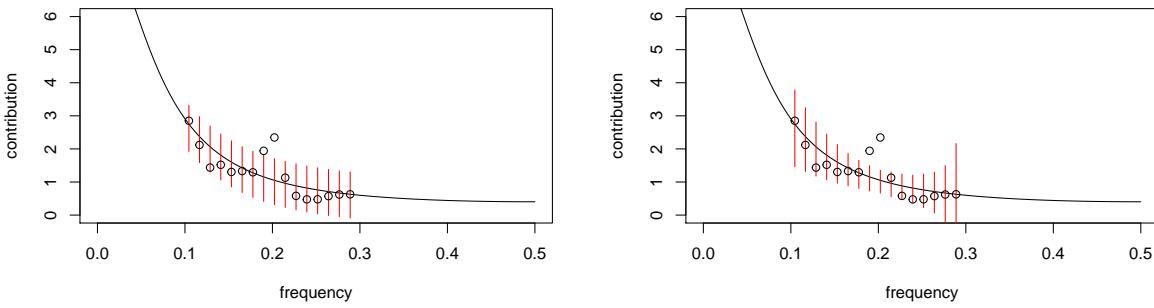


Figure 3.8: MT2, weighted, frequency range  $[0.1, 0.3]$ , two-tailed test. Left — the weights to obtain equal prediction intervals ( $w_k = 1/\sigma_k$ ), right — the weights to increase the power against frequencies around 0.2.

theoretical eigenvectors; the Bonferroni correction is used in Fig. 3.5 (right). The test in Fig. 3.5 (left) is strongly liberal, while Fig. 3.5 (right) is slightly conservative.

Fig. 3.6 contains multiple prediction intervals, which provide the exact test with theoretical eigenvectors as the projection vectors. In Fig. 3.6 (left), the eigenvectors of the matrix (3.38) with  $C = 0$  are taken as  $W_k$ . The upper bounds are slightly lower than in Fig. 3.5 (left). In Fig. 3.6 (right), the information about the presence of a signal of frequency  $\omega = 0.2$  is used and the eigenvectors of the matrix (3.38) with  $C = 1$  are taken as  $W_k$ .

Figs. 3.7 and 3.8 are related to the case of zero weights for the frequencies of the projection vectors outside the frequency interval  $[0.1, 0.3]$ . Fig. 3.7 demonstrates the difference between two-sided and one-sided prediction intervals. In the one-sided case, the upper bound is slightly lower. Fig. 3.8 shows how the weights influence the prediction intervals sizes. We set `weights = c(seq(6.25, 8, 0.25), 8:1)`, what means that the weights increase from 6.25 to 8 with step 0.25 and then decrease from 8 to 1. Influencing weights on the test power will be studied later.

## Study of statistical properties of MC-SSA

Let us describe the methodology of the study of statistical properties of the constructed criteria with bootstrapping. The key items are:

1. The first step is to simulate synthetic data according to the null hypothesis  $M$  times and estimate  $\alpha_I = \alpha_I(\alpha)$  as the proportion of the rejected null hypothesis for a given significant level  $\alpha$ . Check that the necessary condition  $\alpha_I \leq \alpha$  is fulfilled. It is necessary to find the sufficient size  $G$  of the surrogate data.
2. If the necessary condition is fulfilled, then the second step is to simulate synthetic data according to an alternative hypothesis  $M$  times and estimate the power  $1 - \alpha_{II}$  against this hypothesis as the proportion of the rejected null hypothesis for a given significant level  $\alpha$ .

3. Compare different criteria by power against the alternative hypothesis under interest and use the one with the larger power.
4. If there is no test with sufficient power, then it is possible to improve a test, where  $\alpha_I < \alpha$  or  $\alpha_I > \alpha$ . Then find  $\tilde{\alpha}$  such that  $\alpha_I(\tilde{\alpha}) \approx \alpha$  and use the significance level  $\tilde{\alpha}$  instead of  $\alpha$ .

Below we consider these items. As before, we consider the AR(1) parameters  $\varphi = 0.7$  and  $\delta = 1$ . In a case, when the AR(1) parameters are estimated, the maximum likelihood method was applied, where the conditional-sum-of-squares method was used to find starting values.

**Choice of  $G$ .** A sufficient value  $G$  in item 1 can be determined theoretically. The value of  $G$  should be considerably large to estimate the quantiles for the surrogate data. In the case of single prediction intervals with Bonferroni correction,  $(1 - \alpha/H)$ -quantiles should be estimated (let us consider one-tailed tests), while in the case of multiple testing, we need  $(1 - \alpha)$ -quantiles only. For example, if  $\alpha = 0.05$ , for multiple testing,  $G = 1000$  is enough, since the estimate is the 50-th value from the maximum in an ordered sample. However, for single testing and  $H = 100$  projection vectors,  $\alpha/100 = 0.005$  and therefore the 0.995-quantile will be underestimated (and therefore the test will be liberal). Table 3.1 contains the estimates of the type I errors for  $N = L = 200$  and  $\alpha = 0.2$  (then the Bonferroni correction gives the significance level 0.001);  $M = 10000$  simulations were used. Table 3.1 confirms that the multiple methods need smaller  $G$ . Hereinafter, the columns ‘2.5%’ and ‘97.5%’ show the bounds of 95%-confidence intervals for the estimated probabilities (the type I error or the power).

Table 3.1: Type-I error for different  $G$ .

Method	$G$	est. type I error	2.5%	97.5%
MT1	100	0.466	0.456	0.476
BT1	100	0.864	0.857	0.870
MT1	500	0.250	0.241	0.258
BT1	500	0.326	0.316	0.335
MT1	1000	0.219	0.211	0.227
BT1	1000	0.212	0.204	0.221

**Type I error.** The estimates of the family-wise type I errors are contained in Fig. 3.9; we use  $N = 100$ ,  $L = 10$ ,  $M = 1000$ , and  $G = 1000$ .

Fig. 3.9 (left) demonstrates the difference between the use of theoretical and empirical eigenvectors. The pluses and minuses of these versions are as follows.

1. Multiple test, empirical eigenvectors.

Plus: It is possible to reconstruct the signal on the basis of significant projection vectors (empirical eigenvectors).

Minus: The test is liberal.

2. Multiple test, theoretical eigenvectors.

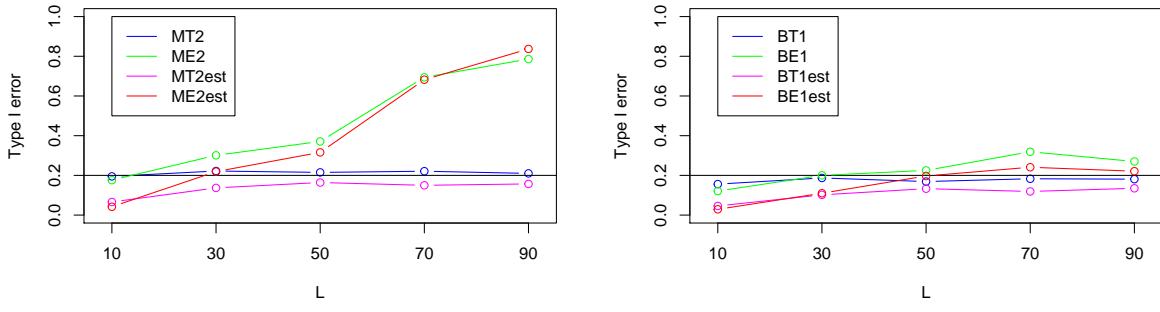


Figure 3.9: Type I error for different  $L$ , empirical and theoretical eigenvectors. Left — the multiple test, right — the single test with the Bonferroni correction.

Plus: The test is exact.

Minus: If the SSA reconstruction of a detected signal should be performed, a paring is needed to put into correspondence theoretical and empirical eigenvectors.

Fig. 3.9 (right) shows the difference between the use of multiple and single tests. The pluses and minuses of these versions are as follows.

### 1. Multiple test

Plus: The test is exact.

Minus: Recalculation is necessary to consider a subset of the projection vectors.

### 2. Single test

Plus: It is easy to consider subsets of projection vectors without recalculation.

Minus: For the version without correction, it is a very liberal test. For the version with Bonferroni correction, it is a slightly conservative test (this is visible for small  $L$ ) and large  $G$  is needed (visible for large  $L$ ).

In addition, Fig. 3.9 shows that if the noise parameters are estimated, then the MT and BT tests becomes very conservative (the type I error is considerably smaller than the given level  $\alpha = 0.2$ ). Also, the conservativeness of Bonferroni correction can compensate for the liberality of the test {M,B}E; it can be a reason why the BE test is less liberal than the ME one in Fig. 3.9.

**Power.** Let us estimate the power of different versions of the Monte Carlo SSA tests. Consider the presence of a signal in  $H_1$  with  $A = 1$  and study the dependence of test power on the signal frequency. We used a standard method of estimation of AR(1) parameters, which ignores a possible presence of a signal.

The estimates of power are depicted in Fig. 3.10. One can see that both the one-tailed version and a narrow frequency interval can increase the test power.

**Correction of type I error.** Let us illustrate item 4 of the described scheme with the help of the example (3.41), where  $H_0$  corresponds to  $A = 0$  and  $H_1$  corresponds to  $A = 1$ ,  $\omega = 0.1$ . As before, we consider a fixed significance level 0.2.

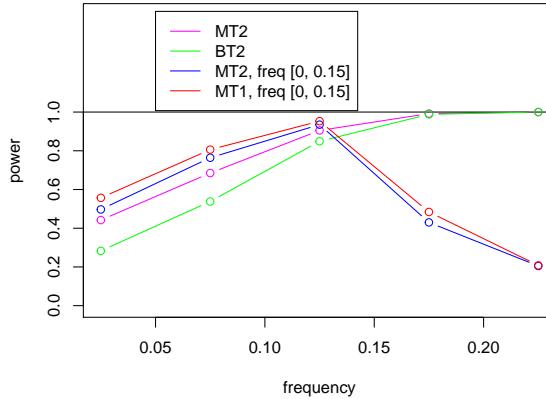


Figure 3.10: Power for alternatives with sinusoidal signals of different frequencies.

If the noise parameters are estimated, then the criterion becomes very conservative (the type I error is two times smaller than the given  $\alpha = 0.2$ ). Changing  $\alpha$ , we can find that for  $\tilde{\alpha} = 0.35$  we obtain  $\alpha_I \approx 0.2 = \alpha$  (Table 3.2, the first three rows).

The criterion with the adjusted formal significance level is more powerful than the original conservative criterion. The last three rows of Table 3.2 show that the use of the conservative test MT1est decreases the power in comparison with that of MT1, while after correction of the significance level to  $\tilde{\alpha} = 0.35$ , the power of MT1est considerably increases.

Table 3.2: Improvement of the power of a conservative test.

	est. type I error	2.5%	97.5%
MT1	0.200	0.176	0.226
MT1est( $\tilde{\alpha} = 0.2$ )	0.069	0.054	0.086
MT1est( $\tilde{\alpha} = 0.35$ )	0.208	0.183	0.234
	est. power	2.5%	97.5%
MT1	0.800	0.774	0.824
MT1est( $\tilde{\alpha} = 0.2$ )	0.575	0.544	0.606
MT1est( $\tilde{\alpha} = 0.35$ )	0.731	0.702	0.758

Note that the dependence of the test power on the significance level shows the test strength against a chosen alternative hypothesis. To find how the test would work if the formal significance level is corrected to obtain a given probability of type I error the dependence of the test power on the probability of type I error should be studied; this dependence is a kind of ROC curve.

### 3.4. Choice of parameters

#### 3.4.1. Introduction

Statements of problems within the framework of SSA can differ in the following aspects.

- A1 Features of interest: we can be interested either in the signal  $S_N$  as a whole or in some of its characteristics. In particular, if  $S_N$  has finite rank, then it has a known parametric form and we can be interested in parameter estimation. The most elaborated problem is the estimation of damping factors and frequencies of exponentially damped sinusoids in noise. For solving this estimation problem it is sufficient to know only  $r$  leading eigenvectors in (1.12), more specifically, the subspace spanned by these eigenvectors (i.e., the estimated subspace of the signal  $S_N$ ); see e.g. ESPRIT-like methods.
- A2 Type of residuals: the residual  $R_N$  is either deterministic or stochastic (or it has both random and deterministic components). These cases correspond to different properties of the SSA decomposition and cause different characteristics of estimators. For example, a finite-rank series  $S_N$  will be the leading component if the deterministic residual is bounded by some constant, while white noise can have any variance for large time series lengths  $N$  and window lengths  $L$ . Also, the structure of the stochastic noise (e.g., white or red) can influence the behavior of estimation errors.
- A3 Choice of the window length: either we can vary the window length  $L$  or  $L$  is fixed. In the former case, the problem of the optimal choice of  $L$  arises. Then, the asymptotic behavior depends on whether  $L$  tends to infinity as  $N \rightarrow \infty$  or not. If we consider a matrix  $L \times K$  with a fixed number of rows as an input, then the only way is to fix  $L$  and to consider  $K$  tending to infinity. The other possible reason for choosing a not very large  $L$  is the computational cost. However, there are recent computational advances, which make calculations very fast, see [113].

Following to [5], we consider different statements of the problem within Aspect A1 and then analyze errors and parameter choice rules following Aspects A2 and A3.

The main information about the time series structure that we obtain within the framework of SSA is contained in the set of eigentriples  $(\sqrt{\lambda_i}, U_i, V_i)$ . Consequently, we obtain not only the reconstructed signal but also much additional information about  $S_N$ . In addition to the problem of reconstructing  $S_N$ , we consider the problems of signal forecasting and signal parameter estimation.

Note that the SVD is determined only by the set of eigenvectors  $U_i$ , since  $\lambda_i = \|\mathbf{X}^T U_i\|^2$  and  $V_i = \mathbf{X}^T U_i / \sqrt{\lambda_i}$ , where  $\mathbf{X}$  is the trajectory matrix of the observed time series. Consequently,  $\tilde{\mathbf{S}} = \sum_{i=1}^r U_i U_i^T \mathbf{X} = \mathbf{P}_r \mathbf{X}$ , where  $\mathbf{P}_r$  is the orthogonal projection on  $\mathcal{L}_r$ . Therefore, we can say that the set of eigenvectors (along with the time series  $\mathbf{X}_N$ ) completely determines the whole SVD expansion and, therefore, the results of forecasting and parameter estimation. Thus it is natural to start the

investigation with the estimation of eigenvectors or, equivalently, the eigenspace estimation. Note that the problem of estimation of the factor vectors  $V_i$  becomes the problem of estimation of the eigenvectors  $U_i$  by changing the window length from  $L$  to  $N - L + 1$ .

Let us remark that the transformations applied to the eigenvectors in order to obtain, for example, the frequency estimates can tremendously change the structure of estimation errors.

Nevertheless, in Section 3.4.2 we consider the errors of signal subspace estimation as the starting point of the investigation. Sections 3.4.3 and 3.4.4 contain the results on reconstruction and forecasting based on the chosen subspace. Section 3.4.5 is devoted to parameter estimation within the framework of the subspace-based methods of signal processing including ESPRIT. In Sections 3.4.2–3.4.5 we give general recommendations on the choice of the window length  $L$  which are based on simulations and known theoretical results. In Section 3.4.6 we consider the convergence rate in different conditions (a fixed window length  $L$  or a window length proportional to the time series length) for the problems investigated in previous sections.

Let us remark that the results of Sections 3.4.2–3.4.6 are valid under the condition of strong separability of the signal from the residual. Section 3.4.7 deals with several examples, in which there is no strong separability. In Section 3.4.8 we consider several versions of Basic SSA and demonstrate some examples of application of the versions designed for the stationary time series to non-stationary ones.

In Section 3.4.9 we briefly describe some origins of the SVD providing the key step of the SSA algorithm. We are interested in these origins, since they imply different views on the problem statement and on the parameter choice.

### 3.4.2. Signal subspace

We will generally rely on the results of the paper [162], which is devoted to the discussion of convergence and also contains the main error terms and their upper bounds.

As a measure of the error for the subspace approximation, we consider the spectral norm of the difference of projectors on the true subspace and the estimated subspace. Note that this norm is equal to the sine of the largest principal angle between these subspaces. The aim of this section is to investigate the dependence of the approximation error on the window length.

Let  $\mathbf{S}_N$  be a signal of rank  $r$ . By  $\mathbf{P}_r^{(s)}$  we denote the orthogonal projector on the signal subspace  $\mathcal{L}_r^{(s)}$ , which is spanned by the left singular vectors  $U_1^{(s)}, \dots, U_r^{(s)}$  of the signal trajectory matrix  $\mathbf{S}$ , and by  $\mathbf{P}_r$  we denote the orthogonal projector on the estimated signal subspace  $\mathcal{L}_r = \text{span}(U_1, \dots, U_r)$ , where  $U_1, \dots, U_r$  are the  $r$  leading left singular vectors of the trajectory matrix of the observed time series  $\mathbf{X}_N$ . Note that we can easily calculate the estimation error  $\|\mathbf{P}_r^{(s)} - \mathbf{P}_r\|$ , since the cosine of the largest principal angle between  $\mathcal{L}_r^{(s)}$  and  $\mathcal{L}_r$  is equal to the  $r$ -th eigenvalue of the matrix  $\mathbf{U}_r^{(s)} \mathbf{U}_r^T$ , where  $\mathbf{U}_r^{(s)} = [U_1^{(s)} : \dots : U_r^{(s)}]$  and  $\mathbf{U}_r = [U_1 : \dots : U_r]$  (see e.g. [163, p. 18]).

Let us consider five examples of time series  $x_n = s_n + r_n$ ,  $n = 1, \dots, N$ ,

$$s_n = 1, r_n = -c(-1)^n, \quad (3.42)$$

$$s_n = b^n \cos(2\pi n/10), r_n = c, \quad (3.43)$$

$$s_n = b^n \cos(2\pi n/10), r_n = \sigma \varepsilon_n, \quad (3.44)$$

$$s_n = b^n \cos(2\pi n/10), r_n = (\sigma \varepsilon_n + c)/\sqrt{2}, \quad (3.45)$$

$$s_n = b^n \cos(2\pi n/10), r_n = \sigma \eta_n. \quad (3.46)$$

Here  $\varepsilon_n$  is a white gaussian noise with variance 1 and  $\eta_n$  is the autoregressive process of order 1 (red noise) with parameter  $\alpha$  and variance  $\mathbf{D}\eta_n = 1$ , that is,  $\eta_n = \alpha \eta_{n-1} + \varepsilon_n$ , where  $\varepsilon_n$  has variance  $1 - \alpha^2$ . In this section, we set  $c = \sigma = 0.1$ ,  $\alpha = 0.5$ ,  $b = 1$ , and  $N = 100$ .

We choose the level of noise in the time series (3.43)–(3.46) to have the same signal-to-noise ratio (SNR), which is conventionally determined as the ratio of the average of squared signal values to the average of squared residual values (or to the variance of residuals if they are random).

Generally, the SNR does not determine the size of the errors of estimates obtained by the SSA processing, since the SNR does not take into consideration the time series length. In fact, the SNR can be used to compare the quality of processing of time series of equal lengths. However, we cannot say that SSA separates signal and noise only if the SNR is larger than a specific value; for example, for any small SNR a sine-wave signal is asymptotically separated from white noise as  $N \rightarrow \infty$  and  $L \sim \beta N$ ,  $0 < \beta < 1$ .

The time series (3.42) is included, since we can compare the results with those in [162, Section 4.2.1]. It appears that the main term of perturbation found in [162] is almost equal to the whole error. Moreover, the behavior of errors depends on whether the lengths of the window and the time series are even or odd. In addition, this is an example of a time series that produces the projector error having the first-order term with respect to the perturbation level which is not the main term of the error as  $N \rightarrow \infty$ .

The time series (3.43)–(3.45) differ in the structure of residuals: deterministic, random, or combined. The time series (3.46) is used to consider a noise that differs from the white noise.

In the case of random residuals, we compute either MSD (mean square deviation) or RMSE (square root of mean square error) as a measure of accuracy. Generally, these criteria yield very similar results. The difference between them is that we can compute the square root before averaging (MSD) or after averaging (RMSE) for the simulation results. In the examples of this section, we estimate MSD using 100 simulations.

We present the results of simulation study for the time series (3.42)–(3.46) in Figures 3.11–3.14. Figure 3.11 shows the errors in the estimation of the projector  $P_r^{(s)}$  on the signal subspace for the examples with deterministic residuals. One can see a tendency for the errors to increase as the window length increases. However, for window lengths  $L$  that are divisible by periods of the time series components, errors generally decrease. This reflects the influence of the multiplicity of  $L$

and/or  $K$  on the periods of time series components. Note that if both  $L$  and  $K$  are divisible by the periods (by 2 in the first example and by 10 in the second one), then the projector perturbation is equal to 0. This corresponds to the case of bi-orthogonality of the trajectory matrices of  $S_N$  and  $R_N$  and therefore to the case of exact separability. If only  $L$  (or  $K$ ) is divisible by the period, then this case can be called left (or right) orthogonality. Thus, if the residual is deterministic and  $S_N$  contains a periodic component, then we observe two effects: the specific behavior of errors in the case of window lengths divisible by the period and a periodic behavior of errors in the general case.

It is clear that if the residual  $R_N$  contains noise, then the exact orthogonality cannot be achieved. Figure 3.12 demonstrates that the decrease of errors for special window lengths does not occur.

In the case of combined residuals (Fig. 3.13), the behavior of projector errors inherits the properties of errors for both pure random and deterministic residuals. Below we show that this feature is valid for other kinds of problems.

To show that the fact that the noise is red (rather than white) does not interfere with the extraction of the signal, we consider the time series (3.46) with a red noise, see Fig. 3.13 (right). Fig. 3.14 compares MSD for different structures of residuals (recall that SNR is the same). One can see that a red noise yields a slightly worse accuracy. Errors for the time series (3.45) lie between those for the time series (3.43) and the time series (3.44).

To summarize, in this section we demonstrated the behavior of errors of projector estimates for different types of residuals. However, we are usually interested in certain features of the signal space, rather than in the projector itself. Therefore, the results of this section provide just the basic information, which can be used for the explanation of further results.

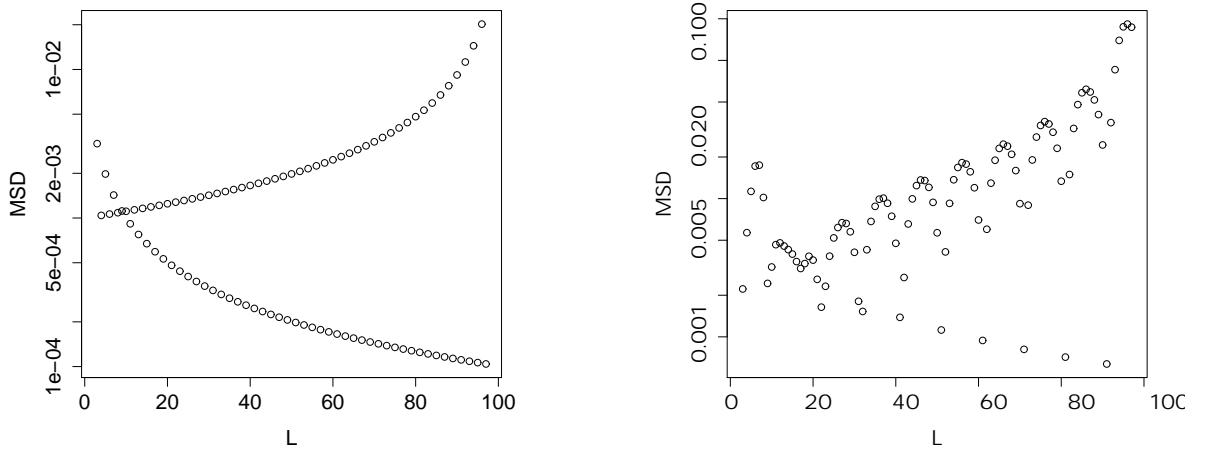


Figure 3.11: MSD of projector estimates: deterministic residuals (log-scale); t.s. (3.42) (left) and t.s. (3.43) (right).

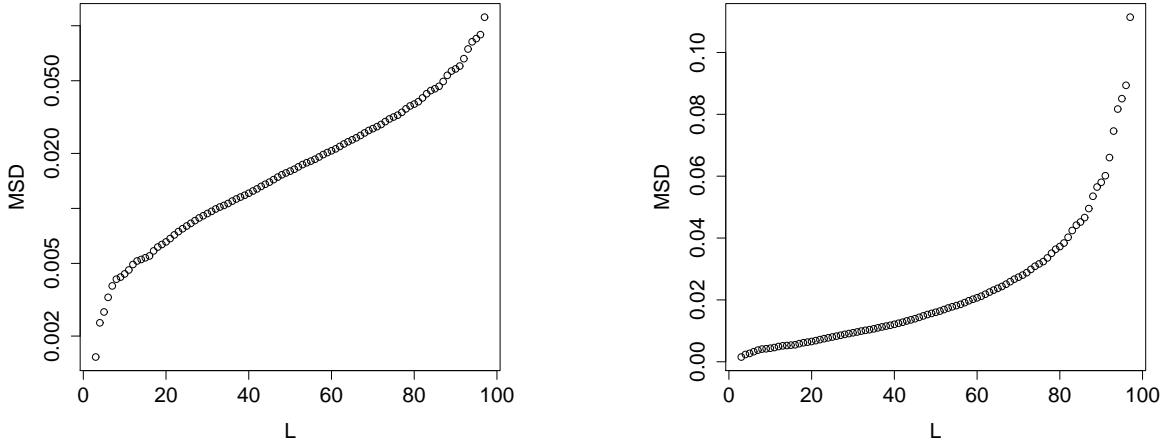


Figure 3.12: MSD of projector estimates: deterministic residuals, t.s. (3.44); log-scale (left) and original scale (right).

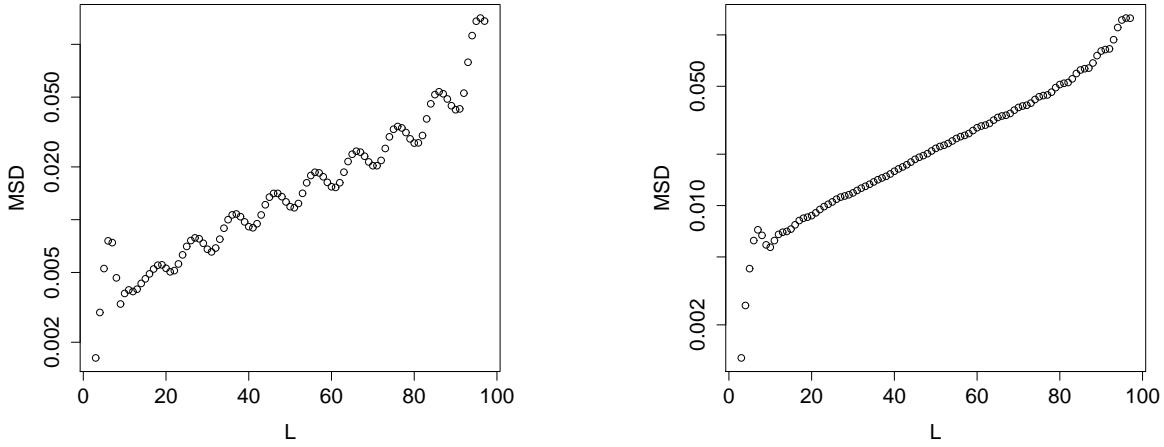


Figure 3.13: MSD of projector estimates: deterministic residuals (log-scale); mixed residuals, t.s. (3.45) (left) and red-noise residuals, t.s. (3.46) (right).

### 3.4.3. Signal extraction

Recall that the reconstructed signal is obtained by applying the diagonal averaging to the reconstructed matrix calculated by the formula  $\tilde{\mathbf{S}} = \mathbf{U}_r \boldsymbol{\Lambda}_r^{1/2} \mathbf{V}_r^T$ , where  $\mathbf{U}_r = [U_1 : \dots : U_r]$ ,  $\mathbf{V}_r = [V_1 : \dots : V_r]$ , and  $\boldsymbol{\Lambda}_r = \text{diag}(\lambda_1, \dots, \lambda_r)$ ;  $U_i$ ,  $V_i$  and  $\lambda_i$  are defined in (1.12). Note that the columns of the matrix  $\mathbf{U}_r$  form a basis of the perturbed signal subspace for the window length  $L$  while the columns of  $\mathbf{V}_r$  form a basis of the perturbed signal subspace for the window length  $N - L + 1$ . First, this means that the results are the same for the window lengths  $L$  and  $N - L + 1$ . Then, as is shown in Figures 3.11–3.14, the signal-subspace perturbation grows as  $L$  increases and decreases as  $N - L + 1$  increases. Thus, the resultant errors are caused by these contradictory tendencies. Fig. 3.12 (right) demonstrates that the growth rate of errors in the projector estimation

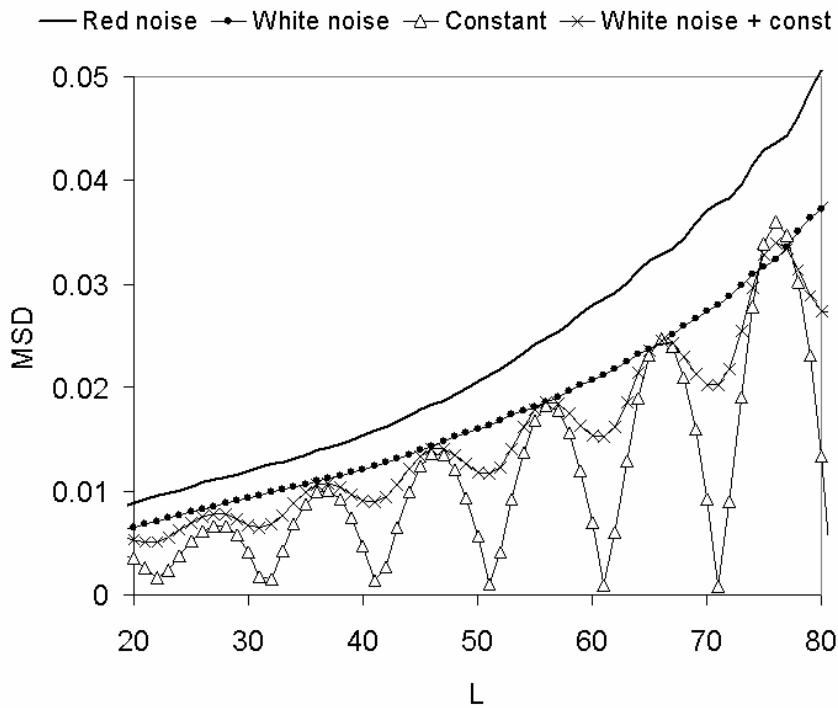


Figure 3.14: MSD of projector estimates: different types of residuals, t.s. (3.43)–(3.46)

is larger for large  $L > N/2$ . Therefore, there is no surprise that the reconstruction errors are large for small window lengths due to large errors in  $\mathbf{V}_r$ . However, the question of the optimal window lengths remains open.

Further, we consider the dependence of the reconstruction error  $\tilde{s}_l - s_l$  on the window length in several examples. Note that an explicit asymptotic form is known for the example of a noisy constant signal with  $s_n \equiv c$  [38]. The paper [38] contains an explicit expression for the variance of the first-order reconstruction errors, where the first order is given with respect to the perturbation of the signal by  $\mathbf{R}_N$  and not with respect to  $N$ . Strictly speaking, the first-order error doesn't need to be the main term of the error as  $N \rightarrow \infty$ . It has been checked by computer simulations that in the case of pure random noise we can consider the first-order error as the main error term (it is not true in the general case, see [162]). Computer simulations confirm that the qualitative results for the constant signal are valid for many other types of signals as well, including oscillations. To describe these results let us present the formula for the dependence of the asymptotic errors on the window length for a constant signal [38].

Let the window length  $L \sim \beta N$ ,  $0 < \beta \leq 1/2$ , and  $l$  be the index of the time series point,  $l \sim \gamma N/2$ ,  $0 \leq \gamma \leq 1$ , as  $N \rightarrow \infty$ . The value  $\gamma = 1$  corresponds to the middle of the time series; consequently, we present the formula for the first half of the time series with window lengths smaller than one-half of the time series length. Then the variance of the first-order errors has the

following asymptotic form:

$$Ds_l^{(1)} \sim \frac{\sigma^2}{N} \begin{cases} D_1(\beta, \gamma), & 0 \leq \gamma \leq 2\min(\beta, 1-2\beta), \\ D_2(\beta, \gamma), & 2\min(\beta, 1-2\beta) < \gamma < 2\beta, \\ D_3(\beta, \gamma), & 2\beta \leq \gamma \leq 1, \end{cases} \quad (3.47)$$

as  $N \rightarrow \infty$ , where

$$\begin{aligned} D_1(\beta, \gamma) &= \frac{1}{12\beta^2(1-\beta)^2} \left( \gamma^2(1+\beta) - 2\gamma\beta(1+\beta)^2 + 4\beta(3-3\beta+2\beta^2) \right), \\ D_2(\beta, \gamma) &= \frac{1}{6\beta^2(1-\beta)^2\gamma^2} \left( \gamma^4 + 2\gamma^3(3\beta-2-3\beta^2) + 2\gamma^2(3-9\beta+12\beta^2-4\beta^3) + 4\gamma(-1+4\beta-3\beta^2-4\beta^3+4\beta^4) + (8\beta-56\beta^2+144\beta^3-160\beta^4+64\beta^5) \right), \\ D_3(\beta, \gamma) &= \frac{2}{3\beta}. \end{aligned}$$

The points of change between the cases in (3.47) correspond to  $l = K - L$  (i.e.,  $\gamma = 2(1-2\beta)$ ) and  $l = L$  ( $\gamma = 2\beta$ ). The former point of change is present if  $K < 2L$  ( $\beta > 1/3$ ). Note that formula (3.47) can be extended to the window lengths  $2 < L < N-1$  ( $0 < \beta < 1$ ) and to the indices of time series points  $0 \leq l \leq N-1$  ( $0 \leq \gamma \leq 2$ ) due to the symmetry of errors with respect to the middle of the time series and by the equivalence of results under the substitution of  $L$  for  $K$  ( $\beta \leftrightarrow 1-\beta$ ).

When we solve the problem of minimizing RMSE of estimation of  $s_l$  at a fixed point  $l$ , the optimal window length varies from  $N/3$  to  $N/2$ , see [38]. This means that even in the case with a constant signal the optimal window length, which minimizes the reconstruction errors as a whole, depends on the importance (weights) of each point of the time series. In any case, the general recommendation is to choose a window length slightly less than one-half of the time series length  $N$ . Note that the optimal window length provides a considerable improvement in the error rate (with respect to the choice  $L = N/2$ ) at the edge time series points, that is, for  $l/N \approx 0$ .

It has been shown for the projectors in Section 3.4.2 that for a noisy sine-wave signal (i.e. if the residuals do not contain a deterministic component), the divisibility of the window length by the sine-wave period is not an important issue. The presence of a deterministic component in the residual makes this divisibility important. A similar effect takes place for the reconstruction errors  $\|\tilde{S}_N - S_N\|$ , see Fig. 3.15 for the time series (3.43)–(3.46) with the same parameters.

To study the influence of the window length on RMSE, we consider RMSE for the reconstruction of the ten last points of the signal. Comparison of graphs in Fig. 3.16 shows that the impact of the divisibility of the window length by the sine-wave period is stronger for the edge points.

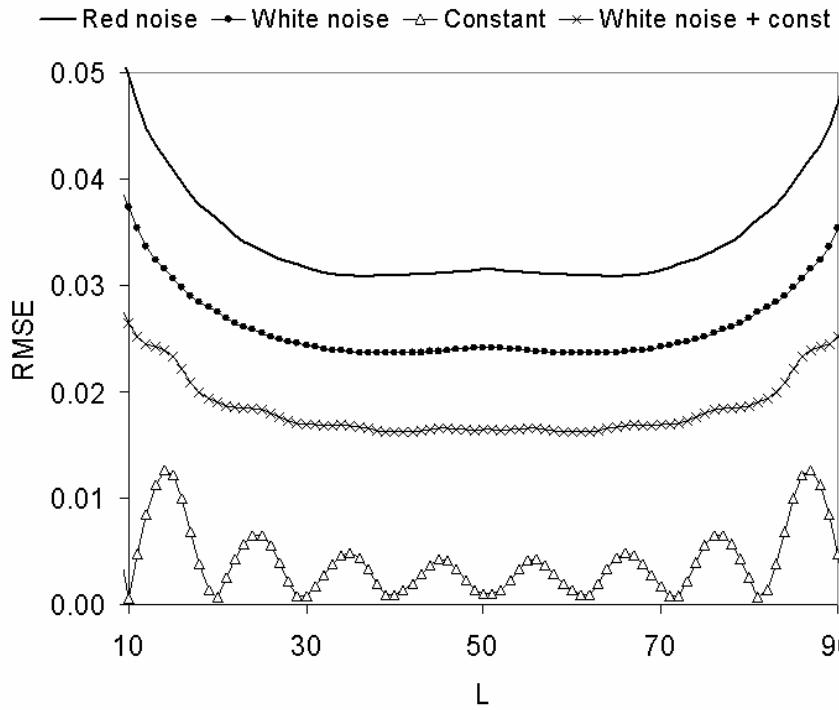


Figure 3.15: RMSE of signal estimates: different types of residuals, t.s. (3.43)–(3.46)

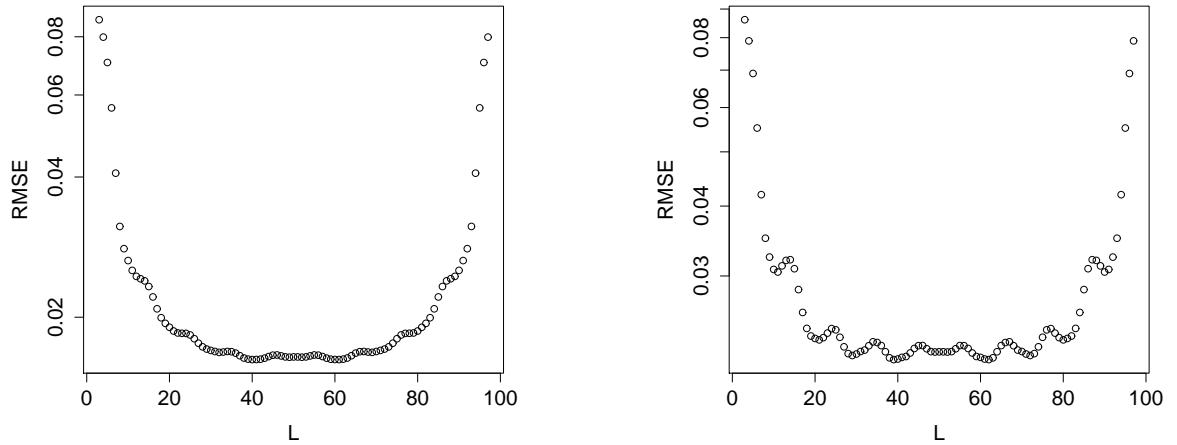


Figure 3.16: RMSE of signal estimates: mixed residuals, t.s. (3.45); the whole series (left) and the last 10 points (right).

In Fig. 3.15 we observe that the optimal window length is close to  $0.4N$  in the case of random residual. However, the divisibility of the window length by the period (if the residuals contain a deterministic component) can be more important than the adjustment, e.g., the transition from  $N/2$  to  $0.4N$ .

### 3.4.4. Recurrent SSA forecast

#### Dependence on the window length

Let us consider the dependence of the forecast accuracy on the window length  $L$  (we consider the signal rank  $r$  as given in advance). Note that the forecasting procedure uses two objects: the LRR itself and the initial data for this LRR taken from the last points of the reconstructed signal  $\tilde{S}_N$ . Let us denote the vector constructed from the  $L - 1$  last signal points by  $V$  and the vector constructed from the  $L - 1$  last reconstructed (i.e., perturbed) signal points by  $V + \Delta V$ . Likewise, we denote the vector of coefficients of the true min-norm LRR by  $A$  and the vector of the estimated LRR coefficients by  $A + \Delta A$ . Then the forecast error is  $A^T \Delta V + (\Delta A)^T V + (\Delta A)^T \Delta V$ .

Therefore, the first-order error consists of two kinds of errors:

1. the errors in the LRR coefficients that are caused by an error of the projection onto the signal subspace  $(\Delta A)^T V$ ;
2. the errors of signal reconstruction  $A^T \Delta V$ .

Let us investigate these two error sources separately. To do this, we apply the LRR that was estimated with the window length  $L_{\text{LRR}}$  to the true signal values and apply the true LRR to the estimated signal values that were reconstructed with the window length  $L_{\text{rec}}$ .

We consider the time series (3.44) with  $\sigma = 0.1$ ,  $N = 399$  and  $\ln b = 0, 0.01, -0.01$ . Estimation (by 1000 simulations) of RMSE of the one-term ahead forecast is depicted in Fig. 3.17 for  $L_{\text{rec}} = 200$ . Values of  $L_{\text{LRR}}$  are varied from 20 to 380 with increment 20. Similar graphs for different  $L_{\text{rec}}$  are presented in Fig. 3.18. The labels of the x-axis correspond to the values of  $L_{\text{rec}}$ .

The line marked ‘total’ shows the accuracy (RMSE) of the forecast with the corresponding window lengths. The line marked ‘LRR’ corresponds to  $(\Delta A)^T V$ . The ‘LRR’ errors look like the errors of the estimator  $P_r$  of the projector on the signal space (see Fig. 3.12 (right)). This is not surprising since the coefficients of the LRR are proportional to the projection  $\mathbf{U}_{r+1,L} \mathbf{U}_{r+1,L}^T \mathbf{e}_L = P_r^\perp \mathbf{e}_L$ . Naturally, the ‘LRR’ errors do not depend on the window length  $L_{\text{rec}}$  used for reconstruction. The typical behavior of this part of the error is as follows: the larger the window length, the larger the error.

The line marked ‘Rec’ corresponds to  $A^T \Delta V$ . We can see that the larger the window length, the smaller the error. This can be interpreted in the following way: extraneous roots that are located close to the uniform distribution on a circle compensate one another (see [112, 107] for several results of this kind).

Figures 3.17–3.18 show that the accuracy of forecasts is stable within a wide range of window lengths. In particular,  $L_{\text{LRR}}$  and  $L_{\text{rec}}$  slightly smaller than  $N/2$  are quite appropriate. Also, we can take either small  $L_{\text{LRR}}$  and  $L_{\text{rec}} \sim N/2$  or  $L_{\text{LRR}} \sim N/2$  and small  $L_{\text{rec}}$ . The former is the preferred choice, since the errors are smaller and more robust to the changes in the window length, see Fig. 3.18.

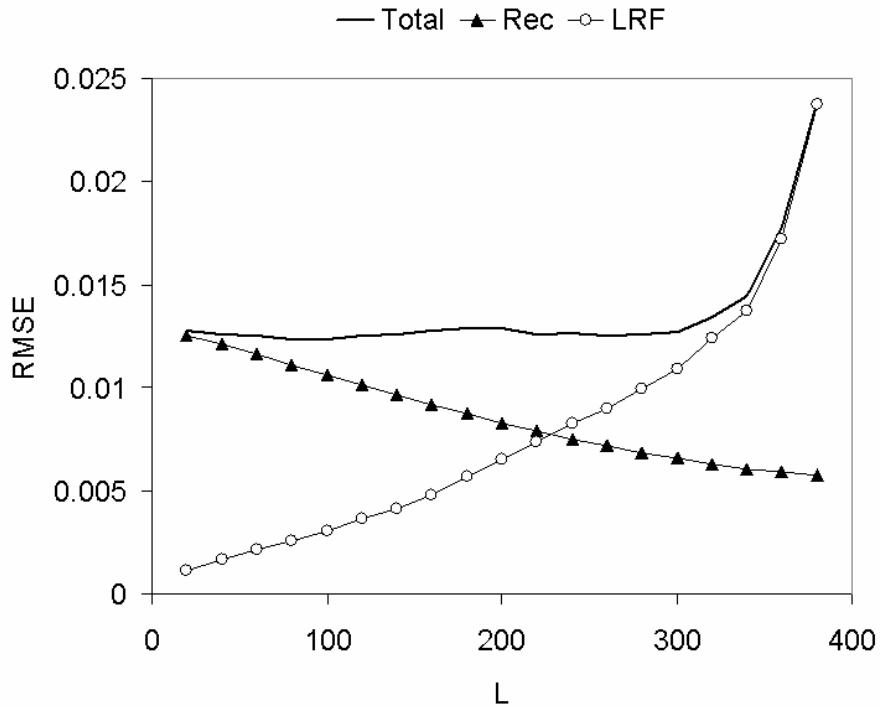


Figure 3.17: RMSE of forecast as a function of  $L_{\text{LRR}}$ : t.s. (3.44) with  $b = 1$ ,  $L_{\text{rec}} = 200$

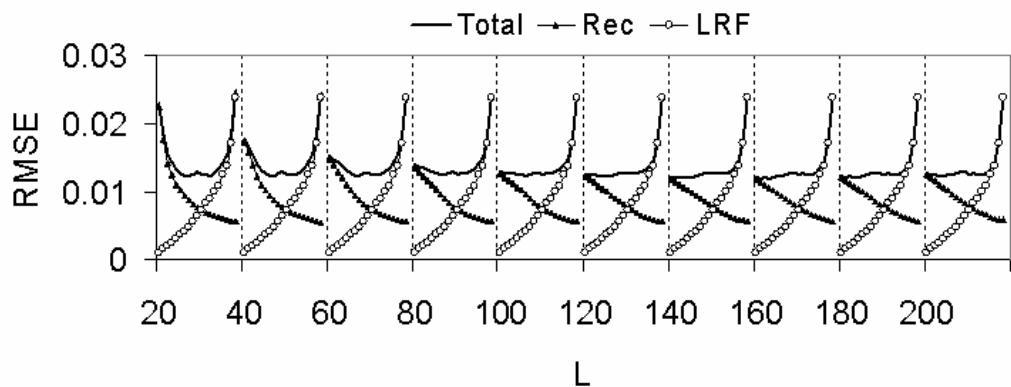


Figure 3.18: RMSE of forecast: t.s. (3.44) with  $b = 1$ ,  
different  $L_{\text{rec}}$

Recommendations on the window length choice naturally depend on the forms of the signal and the residual. Figures 3.19 and 3.20 contain RMSE for damped sine waves. The interpretation of figures and the RMSE behavior are similar.

The deterministic residual can provide a specific behavior of errors. Specifically, for the time series (3.42) we obtain that the choice of an even  $N$  and odd values of  $L_{\text{LRR}}$  provides a decrease in the errors of projectors as the window length increases (see Fig. 3.11 (left)). Therefore, for such choice of  $L$  and  $N$ , we observe similar behavior for two sources of forecast errors and the optimal choice is that the window length should be as large as possible.

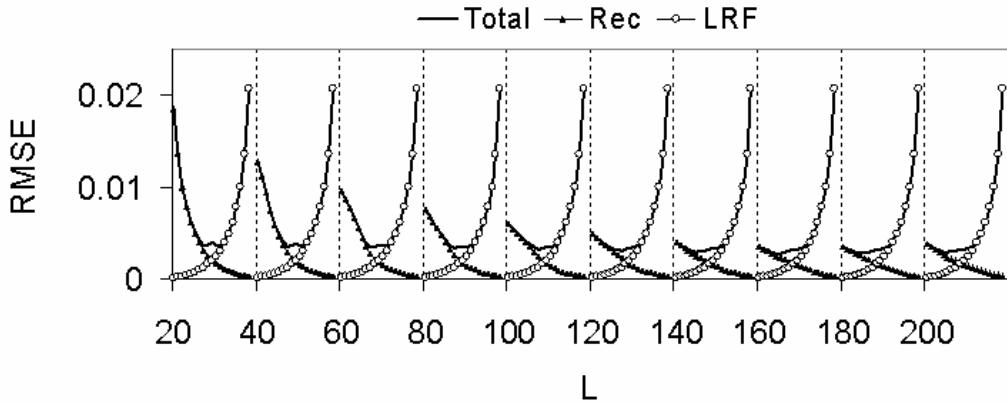


Figure 3.19: RMSE of forecast: t.s. (3.44) with  $b < 1$ , different  $L_{\text{rec}}$

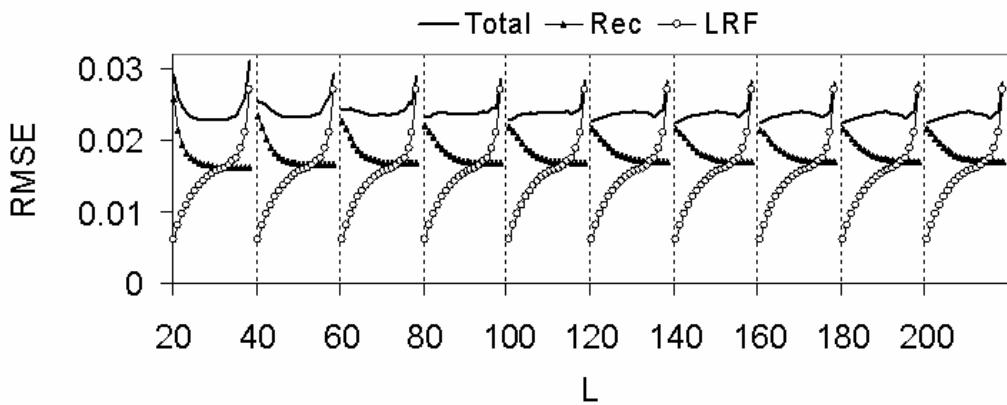


Figure 3.20: RMSE of forecast: t.s. (3.44) with  $b > 1$ , different  $L_{\text{rec}}$

### 3.4.5. Subspace-based methods of parameter estimation

Let us consider the choice of the SSA parameters for the problem of estimating the time series parameters such as t.s. frequency.

The paper [164] contains the following theoretical result: while estimating the frequency of a noisy sinusoid, the asymptotic ( $N \rightarrow \infty$ ) variance of the first-order error has order  $1/(K^2L)$  and the choice of window length is symmetric with respect to  $N/2$ . Therefore, the asymptotic optimal window length is equal to  $N/3$  or  $2N/3$ . Numerical experiments confirm this conclusion. Let us remark that it is not always true that the first-order (with respect to the perturbation level) error is the main-term error as the time series length tends to infinity. Therefore, it is better to check the correspondence between the first-order error and the total error through simulation.

In [165] an explicit form of the asymptotic variance of the first-order error is derived in the general case of damped complex exponentials. In the case of undamped complex exponentials, the derived form coincides with that in [164]. As for damped complex exponentials, the result is that the optimal window length lies between  $N/3$  and  $N/2$  and approaches  $N/2$  as the damping factor increases. It is shown in [165] that for  $s_n = \exp((\alpha + i\beta)n)$ ,  $i = \sqrt{-1}$ , the first-order variances of

the ESPRIT estimates of  $\alpha$  and  $\beta$  are equal. Therefore, the optimal window lengths are the same for estimators of the damping factor  $\alpha$  and of the frequency  $\beta$ .

In the previous sections, we demonstrated that the separability of the signal from deterministic and stochastic residuals has a different nature and therefore leads to different consequences. Let us consider how this difference reveals itself in the problem of frequency estimation by ESPRIT. We perform simulations for the time series (3.43)–(3.46) with  $c = \sigma = 0.1$ ,  $b = 1$ ,  $N = 100$ .

Fig. 3.21 contains the results for the deterministic perturbation, including the specific behavior of RMSE in the case of one-sided (left) orthogonality. Fig. 3.22 contains RMSE of frequency estimates for different kinds of residuals. The behavior of errors is very similar to that in the signal reconstruction, see Fig. 3.15. Also, the errors of frequency and exponential rate (damping factor) estimates are approximately equal if  $L$  is proportional to  $N$ . The main difference from the reconstruction errors is in the size of the errors, which is much smaller. Therefore we use 1000 instead of 100 simulated series to estimate RMSE with sufficient accuracy.

Fig. 3.23 focuses attention on the error behavior for small window lengths. One can see that the dominance of errors for the time series (3.43) with deterministic residuals is a distinctive feature.

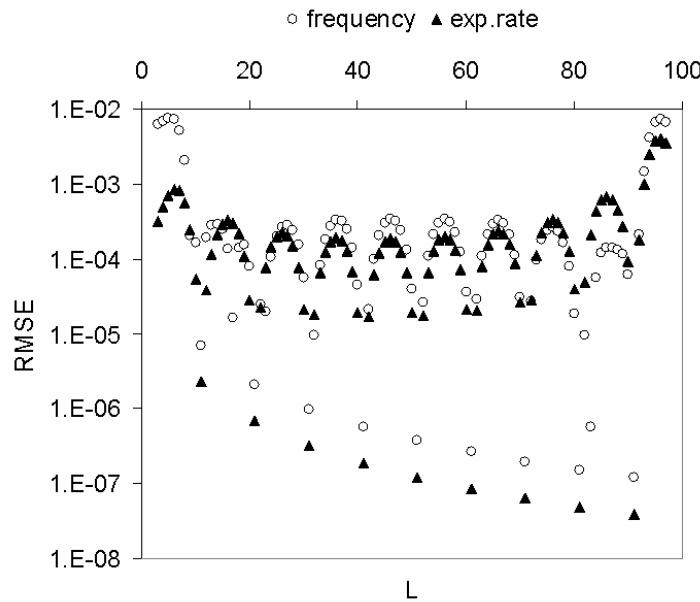


Figure 3.21: RMSE of frequency and exp.rate estimates: t.s. (3.43) (log-scale)

### 3.4.6. The rate of convergence

The paper [162] contains theoretical results on convergence (as the time series length  $N$  tends to infinity) for the methods that are based on the estimation of the signal subspace. Here we investigate the rate of convergence by means of examples.

Let us consider two time series with lengths  $N_1$  and  $N_2$  such that  $N_2 = 4N_1$ . Let RMSE be the measure of accuracy. If the residual  $R_N$  is random, then we perform simulations to estimate

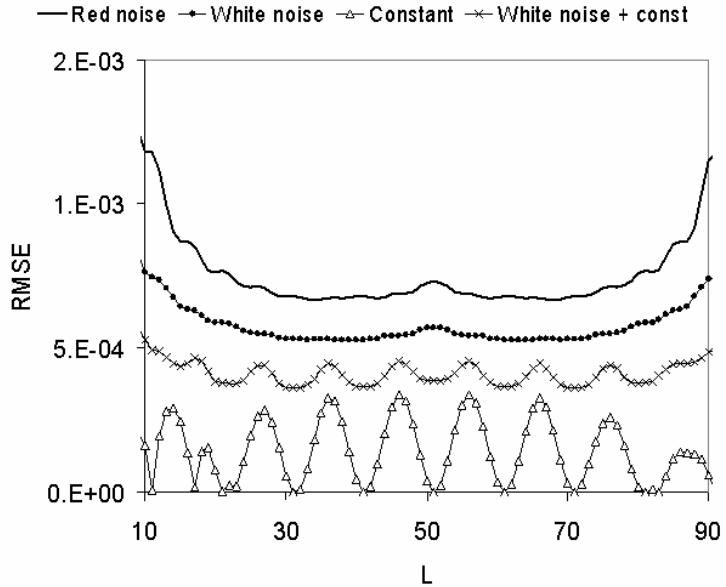


Figure 3.22: RMSE of frequency estimates: different types of residuals, t.s. (3.43)–(3.46),  $L \sim N/2$

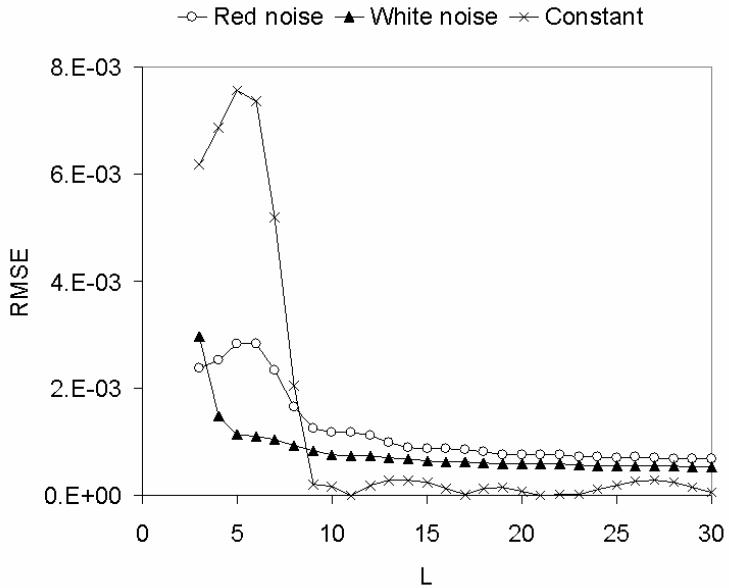


Figure 3.23: RMSE of frequency estimates: different types of residuals, t.s. (3.43)–(3.46),  $L$  is small

RMSE. We denote the ratio of RMSEs for the window lengths  $N_1$  and  $N_2$  by  $\Delta = \text{RMSE}_1/\text{RMSE}_2$ . Then,  $\Delta = 8$  indicates the rate of convergence  $1/N^{1.5}$ ,  $\Delta = 2$  corresponds to the rate of convergence  $1/N^{0.5}$ , and  $\Delta = 1$  means that there is no convergence at all. To estimate  $\Delta$ , we use  $N_1 = 6399$  and  $N_2 = 25599$  (we chose odd time series lengths to consider  $(N+1)/2$  as one of window lengths)

We discuss examples with the following three kinds of perturbation of the signal: by a constant, by noise and by a sum of noise and a constant. Also, we consider two types of random noise, white and red. We perform numerical experiments for the time series (3.43)–(3.46) with  $\sigma = 0.1$ ,  $b = 1$ ,  $\alpha = 0.5$ . In what follows we calculate the frequency and exponential base estimates using the LS-ESPRIT method (the difference with the results of TLS-ESPRIT is small and does not

Table 3.3: The rate of convergence of reconstruction: estimated  $\Delta$

$L$	c	c+wn	wn	rn
$r+1$	1.0	1.0	1.0	1.0
20	—	1.0	1.0	1.0
25	1.0	1.0	1.0	1.0
$(N+1)/2 - 5$	8.0	1.9	2.0	2.1
$(N+1)/2$	—	1.9	2.0	2.1

Table 3.5: The rate of convergence of expon. base estimates: estimated  $\Delta$

$L$	c	c+wn	wn	rn
$r+1$	—	4.1	4.1	4.0
20	—	4.0	4.1	3.9
25	1.0	2.8	4.1	3.9
$(N+1)/2 - 5$	16.0	8.2	8.2	8.1
$(N+1)/2$	—	8.2	8.2	8.1

influence the conclusions).

Tables 3.3–3.6 include the results on convergence based on 1000 simulations. Column ‘c’ corresponds to the time series (3.43) with constant residuals, columns ‘wn’ and ‘rn’ contain the results for white-noise (the time series (3.44)) and red-noise (the time series (3.46)) residuals respectively, and the column ‘c+wn’ includes estimates of  $\Delta$  for the time series (3.45) with combined perturbation.

It would appear reasonable that the convergence rates for fixed window lengths and for window lengths proportional to  $N$  differ. Also, the multiplicity of window lengths to the period of the sine-wave signal (10 in the considered examples) can be important. Therefore, we analyze two sets of window lengths. The first set includes fixed window lengths: the minimal  $L = r+1$ , where  $r = 2$  is the rank of sinusoid,  $L = 20$  is divisible by 10, and  $L = 25$  is a common case. The second set contains two window lengths close to  $N/2$ :  $L = (N+1)/2$  is divisible by 10 and  $L = (N+1)/2 - 5$  is a common case. Note that if there is the exact separability (that is possible in the case of deterministic residuals only), then the ratio  $\Delta$  cannot be calculated (the sign ‘—’ in the tables). We do not consider the windows length  $L > (N+1)/2$ , since these values of  $L$  lead to either the same or worse convergence rates in comparison with the window length  $L \leq (N+1)/2$ .

Let us discuss the results presented in Tables 3.3–3.6 for window lengths tending to infinity and for fixed window lengths separately.

**Window length  $L \sim N/2$**  The simulations provide stable estimates of the convergence rate for the window length  $L$  equal to one-half of the time series length (and more generally, for the window lengths that are proportional to  $N$ ):

(A) “signal+noise” (the time series (3.44), (3.46))

Table 3.4: The rate of convergence of projector estimates: estimated  $\Delta$

$L$	c	c+wn	wn	rn
$r+1$	1.0	1.0	2.2	1.0
20	—	3.1	2.8	1.1
25	1.0	1.0	3.0	1.1
$(N+1)/2 - 5$	4.0	2.0	2.0	2.0
$(N+1)/2$	—	2.0	2.0	2.0

Table 3.6: The rate of convergence of frequency estimates: estimated  $\Delta$

$L$	c	c+wn	wn	rn
$r+1$	1.0	1.0	2.0	1.0
20	—	2.7	2.5	1.0
25	1.0	1.0	2.7	1.0
$(N+1)/2 - 5$	16.0	8.2	8.4	8.2
$(N+1)/2$	—	8.2	8.4	8.2

- (a) the convergence rate of the projector on the signal subspace is  $1/N^{0.5}$ ,
- (b) the convergence rate of the reconstruction of the whole signal (average error) is  $1/N^{0.5}$ ,
- (c) the convergence rate of the frequency and exponential rate estimates is  $1/N^{1.5}$ ;

**(B)** “signal+constant” (the time series (3.43))

- (a) the convergence rate of the projector on the signal subspace is nearly  $1/N$ ,
- (b) the convergence rate of the reconstruction of the whole signal (average error) is nearly  $1/N^{1.5}$ ,
- (c) the convergence rate of the frequency and exponential rate estimates is nearly  $1/N^2$ .

Theoretical results on the reconstruction errors in a particular case of a noisy constant signal [38], see (3.47), provide support for a part of the conclusions derived from the simulations. Results on RMSE for the frequency and exponential base estimates are confirmed in [164], where the case of a noisy sinusoid is considered. Let us remark that the convergence rate  $1/N^{1.5}$  is not surprising, since the Cramér-Rao lower bound for the variance of the frequency estimates has the same order (see, e.g. [166, 167]). On the other hand, the Cramér-Rao lower bound for the variance of estimates of the sinusoid amplitude has the order  $1/N^{0.5}$ , which corresponds to the convergence rate of the reconstruction of the signal. Simulations for the time series (3.46) confirm that for  $L \sim N/2$  the red-noise residuals provide the same convergence rate as the white-noise residuals.

One can see that the convergence rate in the examples with pure random residuals is much worse than that for the example with deterministic (constant) residuals. As one might expect, the example with combined residuals inherits the worst case. Simulations confirm that in the case of a constant residual mixed with a stochastic component (the time series (3.45)) and the window length  $L$  proportional to  $N$ , the rate of convergence is the same as for the case (A).

**Fixed window length  $L = L_0$**  Let us consider the case of a fixed window length  $L$  and  $N \rightarrow \infty$ . The behavior of the rate of convergence is more complicated than the behavior described above.

Analysis of the *reconstruction errors* shows the following behavior:

**(A)** for “signal+noise” (the time series (3.44), (3.46)), there is no convergence to the signal, even for the window lengths divisible by the signal (or noise) periods (divisible by 10 in the considered example);

**(B)** for “signal+constant” (the time series (3.43)), the convergence holds only if  $L$  (or  $K = N - L + 1$ ) is divisible by the period; in general, there is no convergence.

Consequently, if residuals contain noise, there is no convergence. Thus, in general, small window lengths are not suitable for the problems of signal reconstruction.

Let us now consider the *errors of projector and parameter estimates*. The difference with the behavior of reconstruction errors consists in the presence of convergence for small  $L$  and white-noise residuals. However, the simulations for the time series (3.46) with red-noise residuals demonstrate the absence of convergence for projection and frequency estimation. In Tables 3.4 and 3.6, this absence of convergence corresponds to the values ‘1.0’ and ‘1.1’ in the column entitled ‘rn’. Convergence of the exponential base estimates still takes place, see Table 3.5.

The question is how the deterministic (constant, in our examples) compound of the white-noise residuals influences the convergence for small  $L$ . Unfortunately, the errors for the projector and parameter estimates converge to 0 for  $L = L_0$  only if the window length is divisible by the signal periods.

Thus, we can conclude that using small window lengths for frequency estimation is possible only if the residuals are pure white noise, that is, they do not contain deterministic components and are independent. Otherwise, there is no convergence.

Let us compare the estimation error for white and red noise residuals regardless of the absence of convergence in the latter case. Fig. 3.24 shows the absolute values of the estimation errors for  $L = 10$ . The first symbols of the line titles mean types of the estimated objects: projector ('proj'), exponential base ('base'), or frequency ('freq'); the last symbols designate types of residuals. One can see that the sizes of errors for red and white noise residuals are comparable for the considered time series lengths. This effect is stable enough for different parameters of the time series model. Thus we can perform an estimation with good accuracy even in the case of the absence of convergence. This can be explained by the results of [162, formula (2.15)], where the upper bounds for the projector errors are derived. The main term of the upper bound (it is a constant depending on the sinusoid frequency, the parameters of red noise, and the window length) yields the proper order of errors. Moreover, [162, formula (2.9)] provides approximately the same small magnitude  $1.3 \cdot 10^{-3}$  as presented in Fig. 3.24, the line marked 'proj\_rn'. Calculations confirm that this magnitude is approximately equal to  $K(\mathbf{S}\mathbf{S}^T)^\dagger \boldsymbol{\Sigma} (\mathbf{I} - \mathbf{U}_r^{(s)}(\mathbf{U}_r^{(s)})^T)$ , where  $\boldsymbol{\Sigma}$  is the autocovariance  $L \times L$  matrix of the considered red noise. Note that this term does not converge to 0 as the time series length tends to infinity.

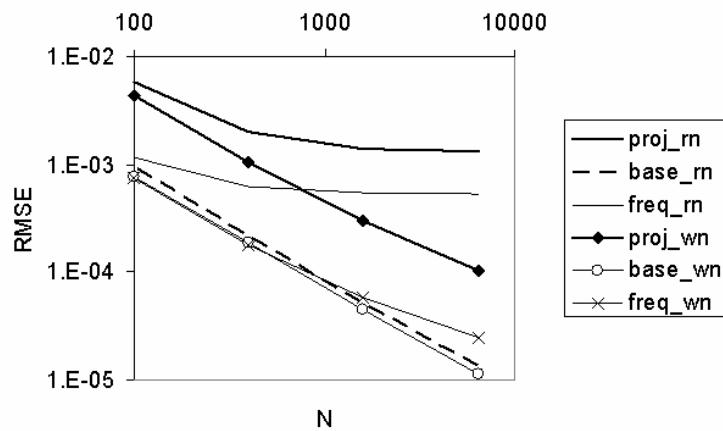


Figure 3.24: Comparison of RMSE for red-noise and white-noise residuals

### 3.4.7. Choice of the window length and separability

#### Modulated sinusoid

Note that the ability of SSA-like methods to extract exponentially modulated (damped) sinusoids implies that the SSA method is not simply a spectral method even if we do not apply SSA for trend extraction. Note that both damped and undamped sinusoids have SSA-rank 2 (or rank 1 if cisoids are considered in the complex-valued case). This feature of SSA significantly extends the set of time series that are suitable for the SSA analysis. Most of the classical methods (e.g. Fourier analysis, seasonal decomposition) deal with either constant amplitudes or with amplitudes proportional to the trend (if any) and can be reduced to constant amplitudes by transferring time series to the logarithmic scale. However, time series consisting of several damped sinusoids cannot be reduced to periodic time series or to multiplicative periodicity. Let us consider, for example, a seasonal component containing both yearly and quarterly oscillations and let the amplitude of yearly periodicity increase whereas the amplitude of quarterly oscillations decrease. Then many classical methods fail whereas the SSA-like methods can easily extract such seasonality.

If the behavior of modulations is more complex than the exponential one, then the SSA-like methods can encounter difficulties. These methods can still extract such oscillations, however the question of the proper choice of the window length arises.

Let us formulate the question in a more specific form. Consider the signal in the form of  $s_n = A(n) \cos(2\pi n \omega)$ , where  $A(n)$  is a slowly (in comparison with  $\omega$ ) varying function. The question: are there any examples when the choice of the window length close to  $N/2$  is not good.

We consider the time series  $x_n = s_n + r_n$  with

$$s_n = \cos(2\pi n/19) + \cos(2\pi n/21), \quad r_n = \varepsilon_n. \quad (3.48)$$

Here  $s_n = A(n) \cos(2\pi n/20)$  with  $A(n) = 2 \cos(\pi n(1/19 - 1/21))$  is a modulated sinusoid of frequency 1/20. The signal has rank 4 and is asymptotically separable from noise, constant residual and others. We have two alternative possibilities. The first possibility is to take  $L$  close to  $N/2$  (e.g., between  $N/3$  and  $N/2$ ) and extract the signal by four leading eigentriples. The second alternative is to take a window length  $L$  so small that the amplitude of the signal is almost constant within the limits of subseries of length  $L$ , and then to extract the signal by two leading components. In the latter case, the left singular vectors are close to the undamped sinusoids and the modulation is caught by the right singular vectors.

Numerical simulations (see Fig. 3.25 for the time series lengths equal to 99, 199, 399, and 999) show that there is no clear choice between the described alternatives. If the time series length is large enough for approximate separability, then the choice  $L \sim N/2$  is better. Otherwise, window lengths close to a couple of periods provide better accuracy. The drawback of the latter choice is that usually we do not know the period and therefore cannot guess the proper window length.

Thus, under the conditions of approximate separability (one can check if these conditions are

met by means of the analysis of the decomposition results, see [1]) the best signal reconstruction uses the window length close to  $N/2$  and the number of the corresponding eigentriples is equal to the signal rank. The advantages of this choice are better accuracy and the independence of the window length choice from the unknown values of periods. In the considered example, for  $L = N/2$ , the reconstruction by four eigentriples performs well starting from  $N = 160$ . However, the choice of  $L = N/2$  is appropriate even for  $N < 120$  if we produce the reconstruction by two eigentriples. Therefore, there is a limited range of time series lengths (approximately from 120 to 160), where the choice of the window length close to  $N/2$  cannot provide an adequate accuracy of signal reconstruction.

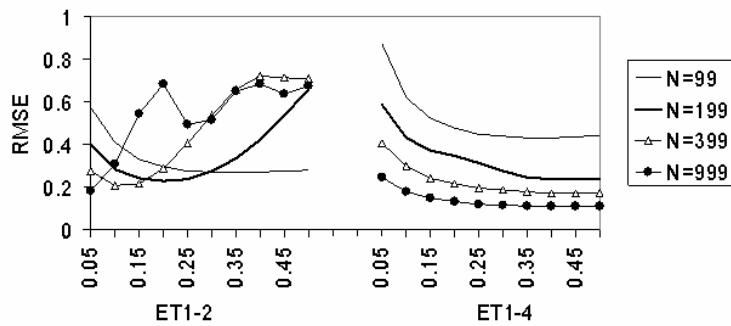


Figure 3.25: RMSE of reconstruction: dependence on  $L/N$  for t.s. (3.48), ET1–2 and ET1–4

Table 3.7 contains RMSE for signal reconstruction using two window lengths:  $L = 40$  (two periods) and  $L = N/2$ . The values in bold indicate smaller errors.

Table 3.7: RMSE of reconstruction: different parameter choices for t.s. (3.48)

$N$	$L = 40$ 2ET	$L = N/2$ 2ET	$L = N/2$ 4ET
99	<b>0.27</b>	<b>0.27</b>	0.45
159	<b>0.23</b>	0.42	<b>0.24</b>
199	<b>0.22</b>	0.65	<b>0.25</b>
399	<b>0.20</b>	0.70	<b>0.16</b>
999	0.19	0.68	<b>0.11</b>

**The case of complex-form modulation** Let us consider the case where the modulated signal is not a signal of finite rank. Let  $N = 399$  and  $x_n = s_n + r_n$ , where

$$s_n = A(n) \cos(2\pi n/20), \quad r_n = \sigma \varepsilon_n, \quad (3.49)$$

$A(n) = \cos(2\pi n^2/10^5)$ . Figures 3.26 and 3.28 show the initial time series with different levels of noise and Figures 3.27 and 3.29 contain the errors of reconstruction by 2, 4, 6, and 8 leading eigentriples. The choice of the window length for signal reconstruction is not crucial for low levels of noise, since we are able to achieve a good accuracy by choosing eigentriples for reconstruction

properly (the larger the window length  $L \leq N/2$ , the larger the number of eigentriples chosen for reconstruction).

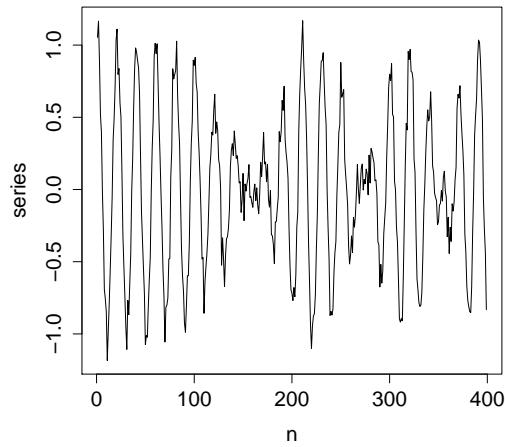


Figure 3.26: Initial time series: t.s. (3.49),  $\sigma = 0.1$

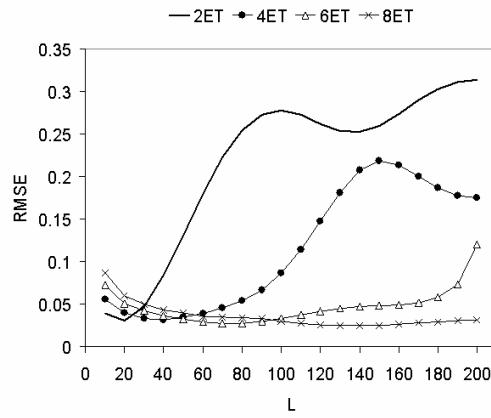


Figure 3.27: RMSE of reconstruction: t.s. (3.49),  $\sigma = 0.1$

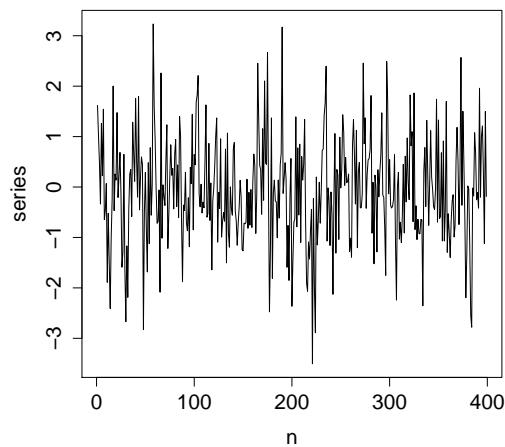


Figure 3.28: Initial time series: t.s. (3.49),  $\sigma = 1$

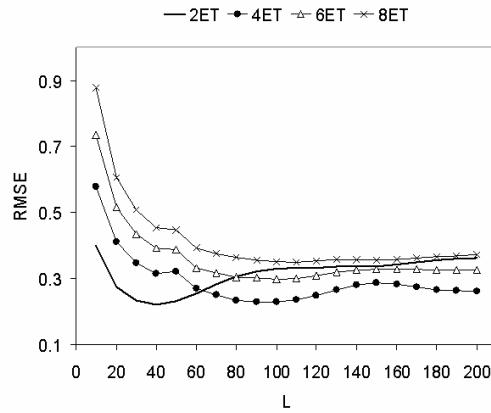


Figure 3.29: RMSE of reconstruction: t.s. (3.49),  $\sigma = 1$

Fig. 3.29 shows that for a high level of noise, the choice of the window length close to  $N/2$  does not provide the best accuracy. In particular, using  $L = 200$  with reconstruction by four leading eigentriples is slightly worse than using  $L = 40$  (two basic periods) and two leading eigentriples. However, if we take, say,  $L = 80$ , then reconstruction by two leading eigentriples is less accurate. Thus, the choice of  $L \sim N/2$  is quite appropriate and can be improved only with the help of additional information (like the value of period) about the analyzed time series.

Note that in this subsection we considered the choice of window lengths for signal reconstruction. Estimation of the basic frequency (e.g., by ESPRIT) is a separate problem, which requires special statements and approaches to its solution.

### The problem of mixing

One of the problems that SSA encounters is a possible lack of strong separability [1] under conditions of weak separability. This problem is caused by equal singular values in SVDs of trajectory matrices of the signal and the residual.

Let us consider some deterministic slowly varying component (i.e., a trend) as a signal. The signal does not necessarily have a finite rank. More likely, it can be approximated by a time series of finite rank  $r_{\text{appr}}$ . Let  $Q$  be the share of  $r_{\text{appr}}$  leading squared singular values in the SVD of the trajectory matrix of the trend. Let us denote by  $\lambda_{\max}^{\text{resid}}$  the maximal singular value generated by the trajectory matrix of the residual and by  $\lambda_{\min}^{\text{trend}}$  the  $r_{\text{appr}}$ -th singular value of the trajectory matrix of the trend.

The case of approximate separability corresponds to large  $Q$  under the condition  $\lambda_{\max}^{\text{resid}} < \lambda_{\min}^{\text{trend}}$ . If  $Q$  is fixed, then, in general, the larger the window length, the larger the rank of the approximating time series and the smaller is  $\lambda_{\min}^{\text{trend}}$ . This observation can lead to the optimal window lengths being considerably smaller than  $N/2$ .

We consider the example

$$s_n = \cos(2\pi n^2/10^5), \quad r_n = \varepsilon_n + 0.5 \cos(2\pi n/10), \quad (3.50)$$

$N = 199$  (Fig. 3.30). It is easy to check numerically that for  $L = 100$  we can get  $r_{\text{appr}} = 2$  with  $Q = 99.5\% = 92\% + 7.5\%$  (the singular values are equal to 74 and 21), whereas for  $L = 30$ ,  $r_{\text{appr}} = 1$  with  $Q = 98\%$  (the singular value is 41). However, the residual produces the maximal singular value equal approximately to 25 for  $L = 100$  and to 15 for  $L = 30$ . This means that we have no strong separability for  $L = 100$  with  $Q = 99.5\%$ . Thus, we can reach  $Q = 92.5\%$  for  $L = 100$  and  $Q = 98\%$  for  $L = 30$  (certainly, these are just rough measures) to satisfy  $\lambda_{\max}^{\text{resid}} < \lambda_{\min}^{\text{trend}}$ , that is confirmed by Fig. 3.31. Note that after extracting the trend with  $L_1 = 30$  the periodicity can be extracted from the residual with a window length close to  $L_2 = 100$ . In [1] this technique is called Sequential SSA.

#### 3.4.8. SSA processing of stationary time series

There are special recommendations concerning the choice of parameters for stationary time series. It is traditionally recommended to perform the centering procedure for the stationary time series before processing (i.e., to subtract the average over the time series) and then to use the

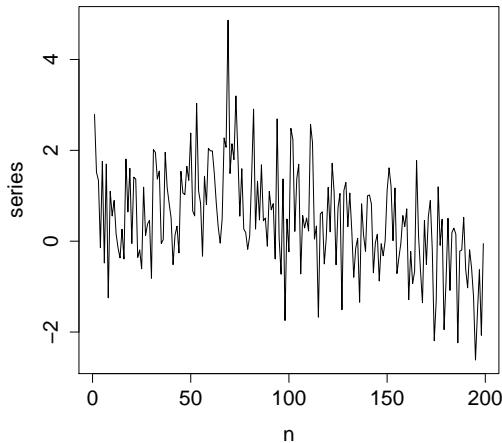


Figure 3.30: Initial time series:  
t.s. (3.50)

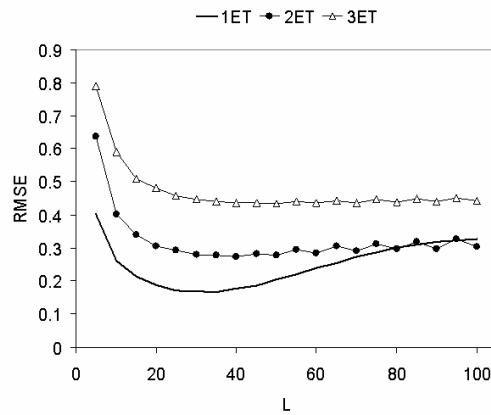


Figure 3.31: RMSE of reconstruction:  
t.s. (3.50)

Toeplitz autocovariance matrix  $\tilde{\mathbf{C}}$  with entries

$$\tilde{c}_{ij} = \frac{1}{N - |i - j|} \sum_{m=0}^{N-|i-j|-1} x_m x_{m+|i-j|}$$

instead of  $\mathbf{C} = \mathbf{X}\mathbf{X}^T$  at the decomposition stage (see Section 1.8.6 and [1] for details of the Toeplitz SSA algorithm). The use of  $\tilde{\mathbf{C}}$  does not provide us with the SVD of the trajectory matrix and therefore with the SVD optimality.

The papers related to the SSA analysis of climate time series (e.g. [168]) consider the Toeplitz SSA as the main version and state that the Basic and Toeplitz versions only slightly differ. Our investigation shows that Toeplitz SSA provides more stable results than Basic SSA (reconstruction, forecast, estimates). However, these results can be inadequate and can have a considerable bias if the time series we analyze is not stationary. It seems that using the Toeplitz version of the SSA algorithm is unsafe if the time series contains a trend or oscillations with increasing or decreasing amplitudes.

Here we can apply a well-known principle: if the method assumes a model, then it gives more precise results when the model is valid; otherwise, the method can produce completely wrong results.

Centering the time series is a less risky procedure than using  $\tilde{\mathbf{C}}$  and can either slightly improve the SSA results or worsen them. Let us note that centering usually increases the rank of the signal making the signal structure more complicated.

Let us recall that besides SSA with centering as preprocessing there are the so-called “Single centering SSA”, which is appropriate for time series with a constant trend, and also the so called “Double centering SSA”, which works well for time series with linear trends (see Section 2.1.1).

In the following subsections we demonstrate the examples of the application of the centering

procedure and the Toeplitz SSA algorithm to non-stationary time series.

### Centering as preprocessing

Let  $N = 199$  and  $x_n = s_n + r_n$  with

$$s_n = 1.005^n, \quad r_n = \varepsilon_n, \quad (3.51)$$

see Fig. 3.32.

The rank of the exponential series is 1 and therefore for its extraction we should choose just one eigentriple. After centering this time series, we obtain a new one  $\mathbf{Y}_N = (y_1, \dots, y_N)$  with  $y_n = 1.005^n - c + \varepsilon_n$ . Therefore, we should choose two eigentriples to extract  $1.005^n - c$ , i.e., we artificially create a more complex structure of the signal. The results of simulations (see Fig. 3.33,  $L$  changes from 5 to 100 with increment 5; reconstructing by ET1 is performed with no centering and also reconstructing ET1–2 is performed with the preliminary centering) confirm that the thickening of the signal structure ends in the increase of errors.

Practically, the centering procedure can increase the errors of reconstruction even of undamped sinusoids in short time series if the time series length is not divisible by the sine-wave period. The explanation is similar to that for the exponential time series: if the time series length is not divisible by the sine-wave period, then the average over the time series is not equal to 0. Therefore, after subtracting this average we transform the signal of rank 2 to a signal of rank 3. For long time series, the effect of the rank increase is diminished because the time series average is almost zero. However, in such a case centering has no sense.

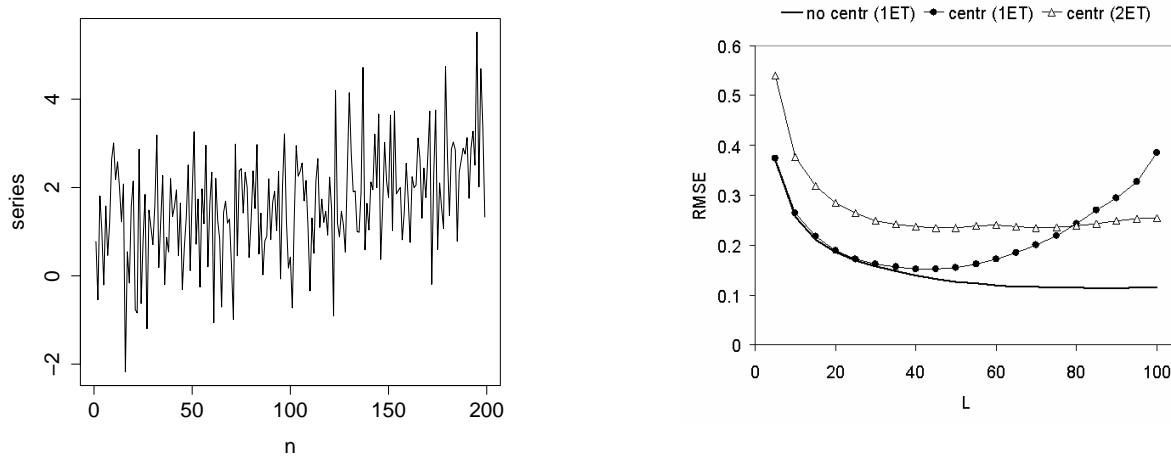


Figure 3.32: Initial time series:  
t.s. (3.51).

Figure 3.33: RMSE of reconstruction:  
t.s. (3.51).

**Remark 33** (About filling in the missing data). Let us mention that the method of the imputation of missing values that is introduced in Section 3.1 does not imply centering (although centering can be used). In the method from [133] the centering procedure is essential, since the missing values are replaced with zeros at the first iteration. It seems that this peculiarity is caused by the fact that the first applications of SSA have been oriented at stationary time series. Certainly, to generalize the method proposed in [133], we can fill in the missing values using the average over the whole time series or use an interpolation method based on neighboring non-missing values.

**Remark 34** (About eigenvalues). Eigenvalues play an important role in the analysis of stationary time series (see, for example, papers devoted to Monte Carlo SSA for detecting a signal in red noise [151]). For stationary time series, the share of eigentriples corresponding to the signal represents the contribution of the corresponding component. However, in the general case of arbitrary time series, there is little point in the eigentriples share, since it depends on absolute values of the times series: in Basic SSA, the distribution of eigenvalues (mostly, leading) depends on the constant compound of the time series while the structure of the time series should not depend on addition or subtraction of a constant. In particular, if the leading eigentriple takes 99.9%, this does not mean that it is enough to take only one eigentriple to approximate the time series with high accuracy. Certainly, we can use information on eigenvalues to identify pairs of eigentriples generated by sinusoids (one sinusoid produces two close or equal eigenvalues). However, the same identification can be done with more powerful tools.

## Toeplitz SSA

Let us demonstrate the consequences of improper use of Toeplitz SSA. First, the application of Toeplitz SSA to non-stationary signals of finite rank  $r$  generally increases the number of nonzero eigenvalues from  $r$  to the maximal possible value equal to  $\min(L, K)$ , that is, the structure can be lost. This means that the number of eigentriples required for accurate reconstruction increases. Also, the constructed approximation can have a wrong structure and, for example, lead to a wrong forecast.

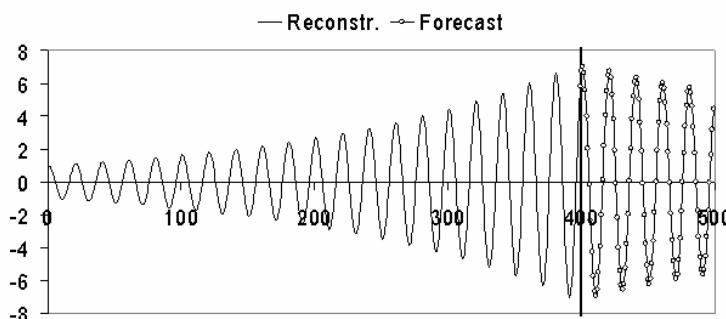


Figure 3.34: Toeplitz SSA forecast for  $x_n = 1.005^n \cos(2\pi n/20)$ ,  $N = 399$ ,  $L = 200$ , ET1–14

We consider two examples of time series of finite rank,  $x_n = 1.005^n \cos(2\pi n/20)$  (rank 2) and  $x_n = 1.005^n$  (rank 1). To get an accurate approximation of the last time series points, we perform

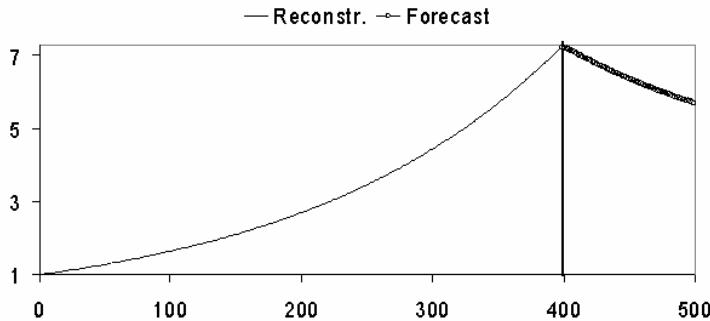


Figure 3.35: Toeplitz SSA forecast for  $x_n = 1.005^n$ ,  $N = 399$ ,  $L = 200$ , ET1–14

Toeplitz-SSA reconstruction with  $L = 200$  based on ET1–14 (ET1–2 for the first example and ET1 for the second one do not provide good accuracy). Figures 3.34 and 3.35 demonstrate that the forecast based on the chosen components is inadequate (moreover, this conclusion does not depend on the chosen window length  $L$  and the number of components).

One can see that the Toeplitz-SSA forecast is completely wrong while the Basic SSA forecast of finite-rank time series is precise.

### 3.4.9. SVD-origins of SSA and the choice of SSA parameters

The key step in SSA is the Singular Value Decomposition (SVD) of the trajectory matrix. The SVD is used for solving different problems, including statistical methods in data analysis. Therefore, the logic of these procedures can also be extended to SSA. Let us briefly describe several origins of SVD-related ideas and views on the SSA-parameter choice generated by them. We do not aim to review the literature on SVD but instead just emphasize the relation between origins and the methodology of SSA.

**Principal Component Analysis (PCA)** This origin is characterized by different kinds of manipulations with variables and with cases in data, i.e., with rows and columns of the trajectory matrix. In particular, the conventional manipulations are centering and standardization of variables. This induces different views on eigenvectors and factor vectors (vectors of principal components), where the latter is interpreted as components of the original time series (maybe, just a little shorter than the original time series). In addition, greater attention to eigenvalues' contribution is transferred from PCA to SSA. This attitude to the trajectory matrix is appropriate when we apply Single centering SSA and when the number of rows ( $L$ ) is fixed and is smaller than the number of columns ( $K$ ). In the general case, the structure of the trajectory matrix does not depend on the transposition of the trajectory matrix, the interpretation of eigenvalues is not so important and, in particular, Double centering is often more natural than Single one.

**Hankel rank-deficient matrices** The relation between such matrices and time series governed by linear recurrence relations has long been known (see e.g. [80]). This technique allows us to analyze noisy time series governed by LRRs. The main application of this approach refers to the signal processing with its approaches to the choice of the method's parameters and consists of

the analysis of a noisy sum of damped/undamped cisoids and the estimation of their parameters (mostly, frequencies). However, this approach can be applied to parameter estimation of arbitrary signals governed by LRRs.

**Spectral analysis [79]** The name of this approach is close to Singular Spectrum Analysis. However, the name ‘Spectral analysis’ means the analysis of stationary time series and their frequency characteristics. On the other hand, ‘Singular Spectrum’ is related to the spectrum of linear operators, i.e. singular values of trajectory matrices in the case of SSA. Thus, the analysis of the singular spectrum does not imply stationarity. As has been stated in [65], the name SSA does not reflect the multifaceted entity of SSA and is traditionally used. The peculiarity of this origin consists in centering the time series before processing and subsequent application of the Toeplitz version of SSA. As was demonstrated in Section 3.4.8, if we apply this technique to non-stationary time series, we likely obtain either imprecise or even meaningless results. Within the framework of the spectral analysis approach, special attention is paid to the red noise (autoregression of order 1) and testing the null hypothesis of the absence of a signal in red noise. In some sense, this corresponds to the problem with a weak signal and strong noise.

**Karhunen-Loève expansion [76]** This expansion is conventionally used in the theory of stochastic processes and originally assumes zero expectation of elements of the considered stochastic process (or subtracting the known averages). In [76], estimation of the process average is performed by using the moving average. Then the errors from this estimation are included in the centered stochastic process and this allows one to apply the method to processes with trends. Thus, the algorithm appears to be very close to PCA and in fact coincides with Single centering SSA. It seems that this origin is used by researchers who are well familiar with the stochastic process techniques and therefore the association with KL expansion helps them to understand the SSA method.

**Dynamical systems [60, 61]** This origin is related to special problems in the theory of dynamical systems with a specific approach to the choice of parameters. However, the contribution of these papers is considerable, since the described algorithm served as an origin of SSA ideas in several applied areas.

## Chapter 4

# SSA for multivariate time series

In this chapter, we consider the problem of simultaneous decomposition, reconstruction and forecasting for a collection of time series from the viewpoint of SSA. The main method of this chapter is usually called either Multichannel SSA or Multivariate SSA, shortly MSSA. The principal idea of the algorithm is the same as for Basic SSA, the difference is in the way of how the trajectory matrix is constructed. MSSA aims to take into consideration the combined structure of a multivariate series to obtain more accurate results.

MSSA is usually considered as an extension of 1D-SSA. However, the algorithm of MSSA was published even earlier than the algorithm of 1D-SSA; see [169], where MSSA was named Extended Empirical Orthogonal Function (EOF) analysis. The MSSA algorithm in the framework of SSA was formally formulated in [170]. Here we consider the algorithm of MSSA for the analysis and forecasting of multivariate time series following the approach described in [1, Chapter 2] for one-dimensional series and in [35] for multidimensional ones.

Whereas there is a more or less canonic description of 1D-SSA, algorithms of analysis and forecasting by MSSA are considerably different (although they are equivalent). Therefore, in Sections 4.1 and 4.3, we review different notations and write down the versions that we consider as canonic ones according to [12] and [31].

In Section 4.1 we expand the methodology of Chapter 2 for the SSA analysis of a system of several time series. It is important to note that there are two main ways of stacking individual trajectory matrices into a joint trajectory matrix: horizontal stacking and vertical stacking. For the MSSA analysis, these two stacking procedures are equivalent.

In Section 4.2, we gather pieces of the theory related to separability, ranks and form of decomposition of sinusoids, which is used in the automatic grouping.

Section 4.3 considers forecasting in MSSA. There are four main variants of MSSA forecasting: recurrent column forecasting, recurrent row forecasting, vector column forecasting and vector row forecasting. We carefully describe the commonalities and differences between these four variants and make their comparison on simulated data.

Section 4.4 contains the extensions of the methods for automatic grouping, which are described in Section 2.5 for one-dimensional time series.

### 4.1. MSSA analysis

#### 4.1.1. Method

Consider a multivariate time series; that is, a collection  $\{\mathbf{X}^{(p)} = (x_j^{(p)})_{j=1}^{N_p}, p = 1, \dots, s\}$  of  $s$  time series of length  $N_p$ ,  $p = 1, \dots, s$ .

Denote  $\mathbf{X} = (\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(s)})$  the initial data for the MSSA algorithm. The generic scheme of the algorithm described in Section 1.2 holds for MSSA; we only need to define the embedding operator  $\mathcal{T}_{\text{MSSA}}(\mathbf{X}) = \mathbf{X}$ .

**Embedding** Let  $L$  be an integer called window length,  $1 < L < \min(N_p, p = 1, \dots, s)$ . For each time series  $\mathbf{X}^{(p)}$ , we form  $K_p = N_p - L + 1$   $L$ -lagged vectors  $X_j^{(p)} = (x_j^{(p)}, \dots, x_{j+L-1}^{(p)})^T$ ,  $1 \leq j \leq K_p$ . Denote  $K = \sum_{p=1}^s K_p$ . The *trajectory matrix* of the multidimensional series  $\mathbf{X}$  is a matrix of size  $L \times K$  and has the form

$$\mathcal{T}_{\text{MSSA}}(\mathbf{X}) = \mathbf{X} = [X_1^{(1)} : \dots : X_{K_1}^{(1)} : \dots : X_1^{(s)} : \dots : X_{K_s}^{(s)}] = [\mathbf{X}^{(1)} : \dots : \mathbf{X}^{(s)}], \quad (4.1)$$

where  $\mathbf{X}^{(p)} = \mathcal{T}_{\text{1D-SSA}}(\mathbf{X}^{(p)})$  is the trajectory matrix of the one-dimensional series  $\mathbf{X}^{(p)}$  defined by (1.1). Thus, the trajectory matrix of a system of time series has *stacked Hankel* structure. Note that

$$\mathcal{T}_{\text{MSSA}}^{-1}(\mathbf{X}) = [\mathcal{T}_{\text{SSA}}^{-1}(\mathbf{X}^{(1)}) : \dots : \mathcal{T}_{\text{SSA}}^{-1}(\mathbf{X}^{(s)})]. \quad (4.2)$$

**Comments to the form of Embedding step** In MSSA (recall that MSSA is an abbreviation for Multivariate/Multichannel SSA), the trajectory matrix is constructed from the stacked trajectory matrices of time series from the considered collection. The stacking can be either vertical or horizontal. We consider the horizontal stacking as the basic version. In some papers, including papers on climatological applications of MSSA (see e.g. [60, 152, 171]), the trajectory matrix is transposed; that is, the trajectory matrices  $\mathbf{X}^{(p)}$  are stacked vertically.

In the case of vertical stacking (for equal series lengths),

$$\mathcal{T}_{\text{MSSA}}(\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(s)}) = \begin{pmatrix} \mathbf{X}^{(1)} \\ \dots \\ \mathbf{X}^{(s)} \end{pmatrix}. \quad (4.3)$$

The vertical stacking is not fundamentally different from the horizontal stacking; the left and right singular vectors interchange (after interchanging  $L$  and  $N - L + 1$ ) and therefore, the way of stacking influences no more than terminology, the choice of parameters and possibly computational costs. It is important to bear in mind that there are many different notations in literature on MSSA, and some of these notations are somewhat controversial.

With the horizontal stacking, we fix the number of rows; that is, the dimension of the column space of the trajectory matrix. In this case, the increase of the time series lengths leads to the increase in the number of columns of the trajectory matrix. Recall that for the horizontal stacking,

the lengths of different time series can differ and this has no influence on the column dimension.

For a simple illustration, consider the horizontal and vertical stackings for two time series,  $\mathbf{X}^{(1)} = (11, 12, 13, 14, 15)$  and  $\mathbf{X}^{(2)} = (21, 22, 23, 24)$ . For the horizontal stacking with  $L = 2$  we have

$$\left( \begin{array}{cccc|ccc} 11 & 12 & 13 & 14 & 21 & 22 & 23 \\ 12 & 13 & 14 & 15 & 22 & 23 & 24 \end{array} \right).$$

The vertical stacking is possible only if the number of columns in the individual trajectory matrices is the same; in our example, therefore, we have to use different window lengths for different time series; set  $L_1 = 2$  and  $L_2 = 1$ :

$$\left( \begin{array}{cccc} 11 & 12 & 13 & 14 \\ 12 & 13 & 14 & 15 \\ \hline 21 & 22 & 23 & 24 \end{array} \right).$$

Let us explain the difference between the vertical staking in the form (4.3) and in the form used in [169]. Let  $\mathbf{X}^{(j)} = (j1, j2, j3, j4, j5)$ , where  $j = 1, 2, 3$  are numbers of some geographic points, the second digit corresponds to time. Then the rows of the trajectory matrix  $\mathbf{X}$  for MSSA with vertical stacking are ordered by series numbers, whereas the rows of the trajectory matrix  $\mathbf{X}_{WN}$  used in [169] are ordered by time:

$$\mathbf{X} = \left( \begin{array}{cccc} 11 & 12 & 13 & 14 \\ 12 & 13 & 14 & 15 \\ \hline 21 & 22 & 23 & 24 \\ 22 & 23 & 24 & 25 \\ \hline 31 & 32 & 33 & 34 \\ 32 & 33 & 34 & 35 \end{array} \right), \quad \mathbf{X}_{WN} = \left( \begin{array}{cccc} 11 & 12 & 13 & 14 \\ 21 & 22 & 23 & 24 \\ \hline 31 & 32 & 33 & 34 \\ 12 & 13 & 14 & 15 \\ \hline 22 & 23 & 24 & 25 \\ 32 & 33 & 34 & 35 \end{array} \right). \quad (4.4)$$

However, the SVD decompositions of the two trajectory matrices in (4.4) are in fact the same and differ only in the order of appearance of the vector components. The left singular vectors of size  $6 = 2 \times 3$  can be split into two parts shifted in time. Each part consists of three geographic points. In [169], the left singular vectors of length  $L_s$  are called EEOFs (extended empirical orthogonal functions). Each EEOF is divided into two parts corresponding to two time lags (the number of lags is equal to  $L = 2$ ) and then each part is depicted as a surface.

**Decomposition** The conventional rank-one matrix decomposition at Decomposition step of MSSA is constructed by applying the SVD to the trajectory matrix; that is, the standard MSSA is an extension of Basic SSA and therefore it can be called Basic MSSA.

Oblique modifications of MSSA are the same as in the 1D case; that is, one can perform nested decompositions by Iterative O-SSA and Filter-adjusted O-SSA. The use of these nested variations is the same as in the 1D case and we refer the reader to Sections 2.2 and 2.3.1 for details.

**Reconstruction** Since  $\mathcal{M}_{L,K}^{(H)}$  in MSSA is the set of stacked Hankel matrices, the orthogonal projector  $\Pi_{\text{stacked } \mathcal{H}}$  to  $\mathcal{M}_{L,K}^{(H)}$  has the form

$$\Pi_{\text{stacked } \mathcal{H}}(\mathbf{Y}) = [\Pi_{\mathcal{H}}(\mathbf{Y}^{(1)}) : \dots : \Pi_{\mathcal{H}}(\mathbf{Y}^{(s)})], \quad (4.5)$$

where  $\Pi_{\mathcal{H}}$  is the projector to the set of Hankel matrices. The equality (4.5) follows from the general form of the projection described in Section 1.2.2. Similar to the 1D case, the reconstructed series are obtained by means of the composition of  $\mathcal{T}_{\text{MSSA}}^{-1}$  and  $\Pi_{\text{stacked } \mathcal{H}}$ .

**Comments** Let us make some comments concerning characteristic features of MSSA.

The eigenvectors  $\{U_i\}$  in the SVD of the trajectory matrix  $\mathbf{X} = \sum_i \sqrt{\lambda_i} U_i V_i^T$  form the common basis of the column trajectory spaces of all time series from the system. The factor vectors  $\{V_i\}$  consist of parts related to each time series separately; that is,

$$V_i = \begin{pmatrix} V_i^{(1)} \\ \vdots \\ V_i^{(s)} \end{pmatrix}, \quad (4.6)$$

where the  $p$ th factor subvector  $V_i^{(p)} \in \mathbb{R}^{K_p}$  belongs to the row trajectory space of the  $p$ th series.

The eigenvectors  $U_i$  reflect the common features of time series, while the factor subvectors  $V_i^{(p)}$  show how these common features appear in each series. It is natural to transform a factor vector to a *factor system* of factor subvectors  $V_i^{(p)}$ . Then the form of transformed factor vectors will be similar to the initial system of series.

Similarly to the one-dimensional case, the main result of application of MSSA is a decomposition of the multivariate time series into a sum of  $m$  multivariate series; the parameters are the window length  $L$  and the way of grouping. For the frequently used case of two groups, we denote by  $\tilde{\mathbf{X}}^{(k)} = (\tilde{x}_j^{(k)})_{j=1}^N$ ,  $k = 1, \dots, s$ , the reconstructed series (usually, the signal) corresponding to the first group of eigentriples  $I_1$ .

**Remark 35.** 1. *The indexing of time points  $1, \dots, N_p$  ( $p = 1, \dots, s$ ) starting from 1 does not mean that all  $s$  series start at the same time; they can also finish at different times. The resultant decomposition obtained by the MSSA algorithm does not depend on the shift between the one-dimensional series and therefore this indexing is only a formality. In particular, MSSA decompositions of two one-dimensional series measured at the same time interval and at disjoint time intervals do not differ.*

2. *The original time ranges for series  $X^{(p)}$  can be useful for depicting and interpreting them. The reconstructed series have the same time ranges as the original series. Factor subvectors from the factor system can also be synchronized for plotting based on the ranges of the initial series; although factor vectors are shorter than the initial series, their time shifts are the*

same.

3. For simultaneous analysis of several time series, it is recommended to transfer them into the same scale. Otherwise, the structure of one particular time series will have too much influence on the MSSA results. To balance the time series, they can be either standardized (centered and normalized, in additive models) or only normalized (in multiplicative models). From the other viewpoint, scaling of individual series can be used to influence the importance of a particular series of the system when, for example, this particular series is more important or has a smaller noise component.
4. The MSSA algorithm can be modified in the same ways as the 1D-SSA algorithm. For example, Toeplitz MSSA and MSSA with projection (including centering) can be considered. (However, these options are not implemented in the current version of RSSA.) Nested oblique variations of 1D-SSA (Iterative O-SSA and Filter-adjusted O-SSA) are implemented in RSSA.
5. In climatology, the SVD of the transposed (vertically-stacked) trajectory matrix defined in (4.1) is traditionally considered [171]. Therefore, the eigenvectors  $\{U_i\}$  correspond to normalized extended principal components in [171], while the factor vectors  $\{V_i\}$  are often called extended empirical orthogonal functions (EOF) in climatology applications, starting from [169].
6. The computational cost of the SVD at Decomposition step depends on the size of the matrix  $\mathbf{X}\mathbf{X}^T$  and hence this computational cost may be significantly different for the horizontally-stacked and vertically-stacked versions of MSSA.

#### 4.1.2. Algorithm

The algorithm of MSSA decomposition differs from Algorithm 1.1 of Basic SSA decomposition only by the form of the embedding operator.

##### ALGORITHM 4.1: MSSA: decomposition

*Input:* Collection  $\{\mathbf{X}^{(p)} = (x_j^{(p)})_{j=1}^{N_p}, p = 1, \dots, s\}$  of  $s$  time series of length  $N_p$ ,  $p = 1, \dots, s$ , window length  $L$ .

*Output:* Decomposition of the trajectory matrix on elementary matrices  $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d$ , where  $\mathbf{X}_i = \sqrt{\lambda_i} U_i V_i^T$ .

- 1: Construct the trajectory matrix  $\mathbf{X} = \mathcal{T}_{\text{MSSA}}(\mathbf{X})$ , where  $\mathcal{T}_{\text{MSSA}}$  is defined by (4.1).
- 2: Compute the SVD  $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d$ ,  $\mathbf{X}_i = \sqrt{\lambda_i} U_i V_i^T$ .

Reconstruction stage is also very similar to Algorithm 1.2 for Basic SSA reconstruction.

---

ALGORITHM 4.2: MSSA reconstruction

---

*Input:* Decomposition  $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d$ , where  $\mathbf{X}_i = \sigma_i U_i V_i^T$  and  $\|U_i\| = \|V_i\| = 1$ , grouping  $\{1, \dots, d\} = \bigsqcup_{j=1}^m I_j$ .

*Output:* Decomposition of the time series system on identifiable components  $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_m$ .

- 1: Construct the grouped matrix decomposition  $\mathbf{X} = \mathbf{X}_{I_1} + \dots + \mathbf{X}_{I_m}$ , where  $\mathbf{X}_I = \sum_{i \in I} \mathbf{X}_i$ .
  - 2:  $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_m$ , where  $\mathbf{X}_i = \mathcal{T}_{\text{MSSA}}^{-1} \circ \Pi_{\text{stacked } \mathcal{H}}(\mathbf{X}_{I_i})$ .
- 

Recall that  $\mathcal{T}_{\text{MSSA}}$  and  $\Pi_{\text{stacked } \mathcal{H}}$  can be expressed through  $\mathcal{T}_{\text{SSA}}$  and  $\Pi_{\mathcal{H}}$  introduced in Chapter 2 for 1D-SSA (see Section 4.1.1).

## 4.2. Elements of MSSA theory

We put into this section several propositions related to the MSSA theory [33, 34].

### 4.2.1. Separability

Separability is the key notion in the SSA theory, since separability of series means the ability of the method to extract them from the given sum-total series. Notion of separability for multidimensional time series is analogous to that for one-dimensional series, which is briefly commented in Section 1.2.2 and is thoroughly described in [1, Sections 1.5 and 6.1]. There is weak separability, which means orthogonality of the trajectory spaces, and strong separability, that means empty intersection of the sets of singular values produced by the separated series.

Generally, conditions of separability of multidimensional time series are more restrictive than that for one-dimensional series. The following sufficient condition of weak separability is valid.

**Proposition 19.** *If time series  $F^{(1)}$  and  $F^{(2)}$ ,  $G^{(1)}$  and  $G^{(2)}$ ,  $F^{(1)}$  and  $G^{(2)}$ , and also  $G^{(1)}$  and  $F^{(2)}$  are weakly  $L$ -separable by SSA, then the two-dimensional time series  $(F^{(1)}, F^{(2)})$  and  $(G^{(1)}, G^{(2)})$  are weakly  $L$ -separable by MSSA.*

The proof is directly follows from the separability definition.

Proposition 19 can be extended to an analogous result for asymptotic separability ( $N_i \rightarrow \infty$ ) and therefore for approximate separability for fixed  $N_i$ .

The following example is a corollary of Proposition 19, considering the known separability conditions for one-dimensional series.

**Example 2.** Consider the example of four harmonic real-valued time series  $F^{(1)}$ ,  $F^{(2)}$ ,  $G^{(1)}$  and  $G^{(2)}$  of length  $N$

$$f_k^{(1)} = A_1 \cos(2\pi\omega_1 k + \varphi_1), \quad f_k^{(2)} = B_1 \cos(2\pi\omega_1 k + \varphi_2),$$

$$g_k^{(1)} = A_2 \cos(2\pi k \omega_2 k + \phi_1), \quad g_k^{(2)} = B_2 \cos(2\pi k \omega_2 k + \phi_2),$$

$\omega_1 \neq \omega_2$ ,  $k = 0, \dots, N-1$ ,  $A_1, A_2, B_1, B_2 \neq 0$ . If  $L\omega_i$  and  $K\omega_i$ ,  $i = 1, 2$ , are integer, then  $(F^{(1)}, F^{(2)})$  and  $(G^{(1)}, G^{(2)})$  are weakly  $L$ -separable by MSSA.

Note that if  $L\omega_i$  or  $K\omega_i$  is not integer, then the series become approximately separable.

Weak separability is not enough for extraction of time series components. Therefore, let us look at strong separability related to eigenvalues produced by time series components. It appears that each time series, multivariate or separate, can produce different eigenvalues in SSA and MSSA. Therefore, the application of a multidimensional modification of SSA can improve (or worsen) the strong separability.

#### 4.2.2. Ranks and subspaces

Consider a system of  $s$  infinite time series  $X = (X^{(1)}, X^{(2)}, \dots, X^{(s)})$ , choose the window length  $L$  and denote  $\mathcal{X}^{(1)}, \dots, \mathcal{X}^{(s)}$  the column trajectory spaces of the series. Let  $\mathcal{X} = \text{span}(\mathcal{X}^{(1)}, \dots, \mathcal{X}^{(s)})$  be the column trajectory space of the collection of time series  $X$ . As well as for one-dimensional time series, we call the dimension of the trajectory space (equal to the rank of the trajectory matrix of the series collection) the rank of the series collection, see Section 1.2.2 for short description of general notions.

Denote the ranks of  $X^{(l)}$  by  $r_l = \dim \mathcal{X}^{(l)} \leq L$ ,  $l = 1, \dots, s$ . For each time series  $X^{(l)}$  we can write out the minimal LRR governing the time series:

$$x_{j+r_l}^{(l)} = \sum_{k=1}^{r_l} a_k^{(l)} x_{j+r_l-k}^{(l)}, \quad \text{where } a_{r_l}^{(l)} \neq 0, \quad l = 1, \dots, s. \quad (4.7)$$

The corresponding characteristic polynomials of the LRR (4.7) are

$$P_{r_l}^{(l)}(\mu) = \mu^{r_l} - \sum_{k=1}^{r_l} a_k^{(l)} \mu^{r_l-k}, \quad l = 1, \dots, s. \quad (4.8)$$

Recall that the roots of the characteristic polynomial of the minimal LRR governing the series are called characteristic roots.

Let

- $p^{(l)}$  be the number of different roots of the polynomial  $P_{r_l}^{(l)}(\lambda)$ ,
- $\mu_m^{(l)}$  be the  $m$ th root of the polynomial  $P_{r_l}^{(l)}(\lambda)$ ,
- $k_m^{(l)}$  be the multiplicity of the root  $\mu_m^{(l)}$ .

Then, from standard theory of time series of finite rank [1], we have that

$$k_1^{(l)} + \dots + k_{p^{(l)}}^{(l)} = r_l, \quad l = 1, \dots, s.$$

The characteristic roots determine the series behavior. For example, if  $k_m^{(l)} = 1$ , then the time series

has the form

$$x_n^{(l)} = \sum_{j=1}^{r_l} C_j^{(l)} (\mu_j^{(l)})^n.$$

Also let

$\mu_1, \dots, \mu_p$  be the pooled set of roots of all the polynomials  $P_{r_1}^{(1)}, \dots, P_{r_s}^{(s)}$ ,  
 $k_1, \dots, k_p$  be the multiplicities of the roots  $\mu_1, \dots, \mu_p$ ,

where multiplicity of a root in the pooled set is equal to the maximal multiplicity of the corresponding root across all the polynomials.

Since the roots are determined by the structure of the trajectory space, the following proposition can be proved.

**Proposition 20.** *The MSSA rank of the multi-dimensional time series  $X = (X^{(1)}, X^{(2)}, \dots, X^{(s)})$  is equal to  $r = \sum_{i=1}^p k_i$ , for  $L > r$ .*

*Proof.* Let us denote for a complex value  $\mu = \rho e^{i2\pi\omega}$ ,  $-1/2 < \omega \leq 1/2$ , and  $j \geq 0$

$$Q(\mu, j) = \begin{pmatrix} \mu^0 0^j \\ \mu^1 1^j \\ \vdots \\ \mu^{L-1} (L-1)^j \end{pmatrix}, \quad P(\mu, j) = \begin{cases} Q(\mu, j), & \text{if } \omega = 0 \text{ or } \omega = 1/2, \\ \Re(Q(\mu, j)), & \text{if } 0 < \omega < 1/2, \\ \Im(Q(\mu, j)), & \text{if } -1/2 < \omega < 0. \end{cases}$$

Here  $\Re(X)$  and  $\Im(X)$  are the real and imaginary parts of  $X$ ,  $Q(\mu, j) \in \mathbb{C}^L$  and  $P(\mu, j) \in \mathbb{R}^L$ . By construction, the vectors  $P(\mu_m, j)$ , where  $m = 1, \dots, p$ ,  $j = 0, \dots, k_m^{(l)} - 1$ , form a spanning set of the trajectory space  $\mathcal{X}$  of  $X$ . Since they are linearly independent for  $L > r$ , the proof is finished.  $\square$

**Matching of series** Simultaneous analysis of several time series is usually performed to identify their inter-relation and to extract their common structure. Recall that for 1D-SSA, a time series has a structure if and only if the trajectory matrix of this series is rank-deficient. Certainly, for a typical real-world series, the trajectory matrix has full rank. Therefore, in what follows we talk about the rank of signal (a part of times series with structure) or its components.

Consider a system of signals  $H = (H^{(1)}, H^{(2)})$  with a rank-deficient trajectory matrix. The structure of a series is reflected in its trajectory space. We can say that two time series have the same structure if their trajectory spaces coincide. For example, for two sine waves with equal periods their trajectory spaces coincide, whatever the values of their amplitudes and phases. This follows from the fact that the trajectory space is the span of subseries of length  $L$  of the initial series. On the other hand, sine waves with different frequencies have entirely different structure and the combined trajectory space of their system is a direct sum of the series trajectory spaces.

If two time series are fully matched, then the trajectory space of one time series can be used for reconstructing or forecasting of the second series. If two series are unrelated and have totally

different structure, then neither series contains any useful information about the other series for the MSSA analysis.

For MSSA, any shift between two time series and any difference between phases of two matched sine waves have no influence on the result of analysis. Therefore, one cannot say anything about the direction of causality. Moreover, asymmetry of influence of one time series to the another series can be caused by different levels of noise. However, the time series  $\mathbf{X}^{(2)}$  can be called *supportive* for the time series  $\mathbf{X}^{(1)} = \mathbf{H}^{(1)} + \mathbf{R}^{(1)}$  if the accuracy of either reconstruction or forecasting of  $\mathbf{H}^{(1)}$  improves if we analyze the system of two series  $\mathbf{X} = (\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$  rather than the series  $\mathbf{X}^{(1)}$  alone.

Numerical experiments confirm that for two matching signals the series with any level of noise, which is not larger than the other one, is always supportive (see e.g. Section 4.3.3).

#### 4.2.3. Decomposition of e-m harmonics

Consider *multidimensional exponentially modulated (e-m) harmonic series*  $\mathbf{S} = (\mathbf{S}^{(1)}, \dots, \mathbf{S}^{(s)})$ , where elements of the  $p$ -th series has the form

$$s^{(p)}(n) = e^{\alpha n} a_p \cos(2\pi\omega n + \phi_p), \quad (4.9)$$

$$0 \leq \phi_p < 2\pi, a_p \neq 0, 0 < \omega \leq 0.5, \sin \phi_p \neq 0 \text{ for } \omega = 0.5, n = 1, \dots, N_p, p = 1, \dots, s.$$

Denote  $L$  a window length,  $K_p = N_p - L + 1$ ,  $K = \sum_{p=1}^s K_p$ ,  $\Lambda^{(1)}$  and  $\Lambda^{(2)}$  are the subspaces of rows and columns of the trajectory matrix of the series  $\mathbf{S}$ .

**Proposition 21.** 1. *L-Rank d of the series S, whose elements are given by (4.9), is 1 if  $\omega = 0.5$ ; in other cases  $d = 2$ .*

2. *If  $\omega \neq 0.5$ , then the subspace  $\Lambda^{(1)}$  has the basis*

$$\{(1, e \cos(2\pi\omega), \dots, e^{\alpha(L-1)} \cos(2\pi(L-1)\omega))^T, \\ (0, e \sin(2\pi\omega), \dots, e^{\alpha(L-1)} \sin(2\pi(L-1)\omega))^T\}.$$

3. *If  $\omega \neq 0.5$ , then the subspace  $\Lambda^{(2)}$  has the basis*

$$\{(c_1^{(1)}, \dots, c_{K_1}^{(1)}; \dots; c_1^{(s)}, \dots, c_{K_s}^{(s)})^T, (d_1^{(1)}, \dots, d_{K_1}^{(1)}; \dots; d_1^{(s)}, \dots, d_{K_s}^{(s)})^T\},$$

where

$$c_j^{(p)} = a_p \cos(2\pi j\omega + \phi_p) \quad \text{and} \quad d_j^{(p)} = a_p \sin(2\pi j\omega + \phi_p), \\ j = 1, \dots, K_p, \quad p = 1, \dots, s.$$

4. *In the case  $d = 1$ , the one-dimensional subspaces  $\Lambda^{(1)}$  and  $\Lambda^{(2)}$  span the vectors  $(1, c, \dots, c^{L-1})^T$  and  $(1, c, \dots, c^{K-1})^T$  respectively, where  $c = -e^\alpha$ .*

**Proposition 22.** Let  $\alpha = 0$ ,  $\omega \neq 0.5$  and  $L\omega$  and  $K\omega$  are integers,  $s = 2$ . Then the eigenvalues of the SVD of the trajectory matrix of the two-dimensional series  $S$ , whose elements are given by (4.9), are the same and have the form  $\lambda_1 = \lambda_2 = (A^2 + B^2)LK/4$ .

**Example 3.** Let

$$s_k^{(1)} = A \cos(2\pi\omega k + \varphi_1), \quad s_k^{(2)} = B \cos(2\pi k\omega k + \varphi_2).$$

If  $L\omega$  and  $K\omega$  are integer, then  $(S^{(1)}, S^{(2)})$  produces two equal eigenvalues in MSSA:  $\lambda_1 = \lambda_2 = (A^2 + B^2)LK/4$ . Note that the series  $S^{(1)}$  itself produces two eigenvalues equal to  $A^2LK/4$ .

*Proof.* 1. Denote  $\psi = \varphi_1/2$ . Then  $\mathbf{X} = P_1 Q_1^T + P_2 Q_2^T$ , where

$$\begin{aligned} p_{1k} &= \cos(2\pi\omega(k-1) + \psi), & p_{2k} &= -\sin(2\pi\omega(k-1) + \psi), \\ q_{1m} &= A \cos(2\pi\omega(m-1) + \psi), & q_{2m} &= A \sin(2\pi\omega(m-1) + \psi), \end{aligned}$$

for  $1 \leq k \leq L$ ,  $1 \leq m \leq K$ .

Set  $p = \|P_1\|/\|P_2\|$ ,  $q = \|Q_1\|/\|Q_2\|$ ,  $S = \|P_1\| \|P_2\| \|Q_1\| \|Q_2\|$  and

$$c_p = \frac{(P_1, P_2)}{\|P_1\| \|P_2\|}, \quad c_q = \frac{(Q_1, Q_2)}{\|Q_1\| \|Q_2\|}.$$

It follows from [1, Proposition 5.2] that  $\tilde{\lambda}_1 = \lambda_1/S$  and  $\tilde{\lambda}_2 = \lambda_2/S$  are the eigenvalues of the matrix

$$\mathbf{B} = \begin{pmatrix} pq + c_p c_q & qc_p + p^{-1} c_q \\ pc_q + q^{-1} c_p & p^{-1} q^{-1} + c_p c_q \end{pmatrix}.$$

Therefore,  $\tilde{\lambda}_1$  and  $\tilde{\lambda}_2$  can be found as the roots of

$$\tilde{\lambda}^2 - (pq + (pq)^{-1} + 2c_p c_q)\tilde{\lambda} + 1 + c_p^2 c_q^2 - c_p^2 - c_q^2 = 0.$$

If  $L\omega$  and  $K\omega$  are integers, then  $c_p = c_q = 0$ ,  $\|P_1\|^2 = \|P_2\|^2 = L/2$  and  $\|Q_1\|^2 = \|Q_2\|^2 = A^2K/2$ . Thus, we have the quadratic equation  $\tilde{\lambda}^2 - 2\tilde{\lambda} + 1 = 0$  with one multiple root  $\tilde{\lambda}_1 = \tilde{\lambda}_2 = 1$ . Therefore,  $\lambda_1 = \lambda_2 = A^2LK/4$ .

In the asymptotic case ( $N$  tends to infinity) the quadratic equation is  $\tilde{\lambda}^2 - b(N)\tilde{\lambda} + c(N) = 0$ , where the coefficients  $b(N) \rightarrow 2$  and  $c(N) \rightarrow 1$  for  $N \rightarrow \infty$ . Since  $S$  is equivalent to  $A^2LK/4$ ,  $\lambda_1 \rightarrow A^2LK/4$ ,  $\lambda_2 \rightarrow A^2LK/4$ .

2. The trajectory matrix of  $(S^{(1)}, S^{(2)})$  can be expressed as  $\mathbf{X} = P_1 Q_1^T + P_2 Q_2^T$ , where  $P_1$  and  $P_2$

are the same as in the one-dimensional case and

$$Q_1 = \begin{pmatrix} A \cos(2\pi\omega_0 + \varphi_1) \\ \vdots \\ A \cos(2\pi\omega(K-1) + \varphi_1) \\ B \cos(2\pi\omega_0 + \varphi_2) \\ \vdots \\ B \cos(2\pi\omega(K-1) + \varphi_2) \end{pmatrix}, Q_2 = \begin{pmatrix} A \sin(2\pi\omega_0 + \varphi_1) \\ \vdots \\ A \sin(2\pi\omega(K-1) + \varphi_1) \\ B \sin(2\pi\omega_0 + \varphi_2) \\ \vdots \\ B \sin(2\pi\omega(K-1) + \varphi_2) \end{pmatrix}.$$

Then  $\|Q_1\|^2 = \|Q_2\|^2 = (A^2 + B^2)K/2$ , if  $L\omega$  and  $K\omega$  are integer, or tend to this number in the asymptotic case. The next steps of the proof are the same as for onedimensional time series.

□

#### 4.2.4. Comments on 1D-SSA and MSSA

##### Covariance structure

Consider in more detail the case of two time series  $\mathbf{X} = (\mathbf{F}, \mathbf{G})$  and let  $\mathbf{F}$  and  $\mathbf{G}$  be the trajectory matrices of  $\mathbf{F}$  and  $\mathbf{G}$  correspondingly. Then, since in MSSA we stack the individual trajectory matrices horizontally, the trajectory matrix of  $\mathbf{X}$  is  $\mathbf{X} = [\mathbf{F} : \mathbf{G}]$ . In accordance with (1.12), the SVD of  $\mathbf{X} = \mathbf{X}^{(H)}$  is  $\mathbf{X} = \sum_i \sqrt{\lambda_i} U_i V_i^T$ , where  $\lambda_i$  and  $U_i$  are eigenvalues and eigenvectors of the matrix  $\mathbf{S} = \mathbf{S}_{\text{MSSA}}^{(H)} = \mathbf{X} \mathbf{X}^T = \mathbf{F} \mathbf{F}^T + \mathbf{G} \mathbf{G}^T$  and  $V_i = \mathbf{X}^T U_i / \sqrt{\lambda_i}$ .

Consider now the vertical stacking of the trajectory matrices of  $\mathbf{F}$  and  $\mathbf{G}$  in the trajectory matrix:  $\mathbf{X}^{(V)} = \begin{pmatrix} \mathbf{F}^T \\ \mathbf{G}^T \end{pmatrix} = (\mathbf{X}^{(H)})^T$ .

The SVD of  $\mathbf{X}^{(V)}$  is then  $\mathbf{X}^{(V)} = \sum_i \sqrt{\lambda_i} V_i U_i^T$ , the transposed SVD of  $\mathbf{X}$ . Here  $\lambda_i$ ,  $V_i$  and  $U_i$  are the same as above but now they have different interpretation: in particular,  $V_i$  (they are often called EOFs, see item 5 of Remark 35 in Section 4.1.1) are the eigenvectors of

$$\mathbf{S}_{\text{MSSA}}^{(V)} = \mathbf{X}^{(V)} (\mathbf{X}^{(V)})^T = \begin{pmatrix} \mathbf{F}^T \mathbf{F} & \mathbf{F}^T \mathbf{G} \\ \mathbf{G}^T \mathbf{F} & \mathbf{G}^T \mathbf{G} \end{pmatrix}.$$

The last formula clearly demonstrates the relation between the two versions (horizontal stacking and vertical stacking) of MSSA and shows that MSSA takes into consideration cross-covariances of time series (more precisely, we obtain the cross-covariances if centering of the one-dimensional series is done at the preprocessing stage).

##### Choice of the window length

The choice of the window length for one-dimensional SSA was reviewed in Section 3.4. The problem of the choice of the window length in MSSA is more complicated than that in 1D-SSA.

Until now there is no in-depth study of the problem of the choice of the optimal window length for analysis and, to an even greater extent, for forecasting of multidimensional time series. Moreover, the choice of the best window length for MSSA forecasting differs for different types of forecasting methods; some numerical investigation of this problem is performed in Section 4.3.3.

By analogy with the one-dimensional case, we can formulate some key principles for the choice of  $L$ . The main principle is the same as for 1D-SSA and states that the choice of  $L$  should provide (approximate) separability of series. However, the MSSA case has additional features. Different approaches to the choice of the window length can be partly explained as follows. In 1D-SSA, it makes sense to constraint the window length to the interval  $2 \leq L \leq [(N+1)/2]$ , since the SVD expansions for window lengths  $L$  and  $N-L+1$  coincide. For the MSSA-analysis of more than one time series, the expansions for all possible window lengths  $2 \leq L \leq \min_i N_i - 1$  are generally different. In particular, while in the 1D-SSA analysis it makes no sense to take  $L > (N+1)/2$ , in the MSSA analysis it makes perfect sense choosing large  $L$  (and hence small  $K_i = N_i - L + 1$ ) for trend extraction and smoothing.

Since  $L$  in 1D-SSA does not exceed half of the time series length, the divisibility of  $L = \min(L, K)$  on possible periods of oscillations is recommended in 1D-SSA. In MSSA,  $\min(L, K_i)$  is not necessary equal to  $L$  and therefore one also has to pay attention to the values of  $K_i$ .

In 1D-SSA, the most detailed decomposition can be obtained if the trajectory matrix  $\mathbf{X}$  has maximal rank. In the general case of SSA-family methods, this corresponds to the case of a square trajectory matrix. Thus, for a system of  $s$  time series of length  $N$  the window length in MSSA providing the square trajectory matrix  $\mathbf{X}$  is approximately  $sN/(s+1)$ . For the case of two time series this corresponds to  $2L/3$  for MSSA.

Numerical investigations show that the formula  $L \simeq sN/(s+1)$  is appropriate for the decomposition of a small number of time series (see simulation results in Section 4.3.3), but does not look suitable for the system of many short series (the values of  $K_i$  become too small for achieving separability). Generally, the choice  $L \simeq N/2$  is still appropriate for MSSA.

Various special techniques can be transferred from 1D-SSA to MSSA, such as Sequential SSA. Sequential 1D-SSA is based on successive application of 1D-SSA with different window lengths, see [29, Section 2.5.5] for more details. Sequential MSSA can be applied in a similar manner. In addition to the reasons which are similar to the 1D case, we may find extra arguments in favour of Sequential MSSA. In particular, if trends of different one-dimensional series are of different structure, a smaller window length can be applied to achieve similarity of eigenvectors and to improve separability. After that, the residuals with a common structure (e.g., containing the seasonality) can be simultaneously decomposed with a larger window length.

### 4.3. MSSA forecasting

Recall from Section 1.9.3 that forecasting in 1D-SSA is performed for a signal component which can be separated by 1D-SSA and is governed, perhaps approximately, by an LRR. For

brevity, we will talk about forecasting of the whole signal. 1D-SSA provides an estimate of the signal subspace and thereby an estimate of one of LRRs governing the signal. The recurrent 1D-SSA forecasting continues the estimated signal by the estimated LRR. The vector 1D-SSA forecasting continues the reconstructed vectors in the given subspace. Methods of one-dimensional SSA forecasting in a given subspace are described in Section 1.9.3.

As in 1D-SSA, methods of MSSA forecasting can be subdivided into recurrent and vector forecasting. In contrast with 1D-SSA, rows and columns of the trajectory matrix in MSSA have different structure. Therefore, there exist two kinds of MSSA forecasting: row forecasting and column forecasting; this depends on which of the two spaces the forecasting is made (row or column space respectively). In total, there are four main variants of MSSA forecasting: recurrent column forecasting, recurrent row forecasting, vector column forecasting and vector row forecasting.

There are different names for the same forecasting methods. In [35], column and row forecasting are called  $L$ - and  $K$ - forecasting. In [172], these methods are called horizontal and vertical forecasts and the trajectory matrix is transposed. In Sections 4.1.1 and 4.2.4 we have explained the choice of orientation of the MSSA trajectory matrix and the connection between the horizontally-stacked and vertically-stacked trajectory matrices of separate time series. We use the name ‘column’ and ‘row’ with respect to the horizontally-stacked trajectory matrices as defined in Section 4.1.1. Probably, the first version of recurrent row MSSA forecasting was proposed in [32].

In the column forecasting methods, each time series in the system is forecasted separately but in a given common subspace (i.e., using the common LRR). In the row forecasting methods, each series is forecasted with the help of its own LRR applied to the whole set of series from the system. Let us describe all four variants of MSSA forecasting.

#### 4.3.1. Method

##### Common notation

First, we introduce some common notation used for description of all the variants of MSSA forecasting.

Denote by  $\bar{A} \in \mathbb{R}^{Q-1}$  the vectors consisting of the last  $Q - 1$  coordinates of  $A \in \mathbb{R}^Q$ ; that is, the vectors with the first coordinate removed are indicated by the line on the top of the vector. Denote by  $\underline{A} \in \mathbb{R}^{Q-1}$  the vectors consisting of the first  $Q - 1$  coordinates of  $A$ ; by  $\pi(A)$  we denote the last coordinate of the vector. For a matrix  $\mathbf{A} = [A_1 : \dots : A_r]$ , we denote  $\bar{\mathbf{A}} = [\bar{A}_1 : \dots : \bar{A}_r]$  and  $\underline{\mathbf{A}} = [\underline{A}_1 : \dots : \underline{A}_r]$  and let  $\boldsymbol{\pi}(\mathbf{A}) = (\pi(A_1), \dots, \pi(A_r))^T$  be the last row of the matrix  $\mathbf{A}$ .

Consider the following form of  $B \in \mathbb{R}^K$ , where  $K = \sum_{i=1}^s K_i$ , induced by the structure of the

row trajectory space:

$$B = \begin{pmatrix} B^{(1)} \\ B^{(2)} \\ \vdots \\ B^{(s)} \end{pmatrix}, \quad \underline{\underline{B}} = \begin{pmatrix} \underline{B}^{(1)} \\ \underline{B}^{(2)} \\ \vdots \\ \underline{B}^{(s)} \end{pmatrix}, \quad \bar{\bar{B}} = \begin{pmatrix} \bar{B}^{(1)} \\ \bar{B}^{(2)} \\ \vdots \\ \bar{B}^{(s)} \end{pmatrix}, \quad (4.10)$$

where  $B^{(j)} \in \mathbb{R}^{K_j}$ , and let  $\boldsymbol{\mu}(B) = (\boldsymbol{\pi}(B^{(1)}), \dots, \boldsymbol{\pi}(B^{(s)}))^T$ . Also, for  $\mathbf{B} = [B_1 : \dots : B_r]$  let  $\underline{\underline{\mathbf{B}}} = [\underline{\underline{B}}_1 : \dots : \underline{\underline{B}}_r]$  and  $\mathbf{B}^{(j)} = [B_1^{(j)} : \dots : B_r^{(j)}]$ .

Assume that the group  $I$  corresponding to the forecasted component is given by the set of the leading components at Decomposition step of Algorithm 4.1; this assumption is made just for simplifying the formulas. Thus, let  $r$  leading eigentriples  $(\sqrt{\lambda_j}, U_j, V_j)$  be identified and chosen as related to the signal of rank  $r$  so that  $I = I_1 = \{1, \dots, r\}$ ,  $\mathbf{U} = [U_1 : \dots : U_r]$ ,  $\mathbf{V} = [V_1 : \dots : V_r]$ . The reconstructed series  $\tilde{\mathbf{X}}$ , its trajectory matrix  $\tilde{\mathbf{X}}$  and the reconstructed matrix  $\hat{\mathbf{X}}$  are defined in Section 1.2.1. Define  $\mathcal{L}^{\text{col}} = \text{span}(U_i, i \in I)$ ,  $\mathcal{L}^{\text{row}} = \text{span}(V_i, i \in I)$ . The reconstructed matrix  $\hat{\mathbf{X}} = [\hat{\mathbf{X}}_1 : \dots : \hat{\mathbf{X}}_K]$  consists of the column vectors which are the projections of the column vectors of the trajectory matrix on the chosen subspace  $\mathcal{L}^{\text{col}}$ .

To avoid repeating the transpose sign, denote  $\tilde{\mathbf{Y}} = [\tilde{\mathbf{Y}}_1 : \dots : \tilde{\mathbf{Y}}_L] = \tilde{\mathbf{X}}^T$ ,  $\hat{\mathbf{Y}} = [\hat{\mathbf{Y}}_1 : \dots : \hat{\mathbf{Y}}_L] = \hat{\mathbf{X}}^T$ ,  $\hat{\mathbf{Y}}_k = \hat{\mathbf{X}}_k^T$ .

## Recurrent MSSA forecast

We denote the vector of forecasted signal values for each time series by  $R_N = (\tilde{x}_{N_1+1}^{(1)}, \tilde{x}_{N_2+1}^{(2)}, \dots, \tilde{x}_{N_s+1}^{(s)})^T$ . Recurrent forecasting is closely related to missing data imputation for components of vectors from the given subspace and in fact uses Proposition 14. Following [35], we will write the forecasting formulas for two versions of the recurrent MSSA forecast: column (generated by  $\{U_j\}_{j=1}^r$ ) and row (generated by  $\{V_j\}_{j=1}^r$ ). These one-term ahead forecasting formulas can be applied for  $M$ -term ahead forecasting by using the recurrence.

The column recurrent forecasting performs forecast by an LRR of order  $L - 1$  applied to the last  $L - 1$  points of the reconstructed signal; that is, the same LRR and different initial data. The row recurrent forecasting constructs  $s$  different linear relations, each is applied to the set of  $K_i - 1$  last points of series; that is, the LRRs are different but the initial data for them is the same.

**Column forecast** Denote by  $\mathbf{Z}$  the matrix consisting of the last  $L - 1$  values of the reconstructed signals:

$$\mathbf{Z} = \begin{pmatrix} \tilde{x}_{N_1-L+2}^{(1)} & \dots & \tilde{x}_{N_1}^{(1)} \\ \tilde{x}_{N_2-L+2}^{(2)} & \dots & \tilde{x}_{N_2}^{(2)} \\ \vdots & \vdots & \vdots \\ \tilde{x}_{N_s-L+2}^{(s)} & \dots & \tilde{x}_{N_s}^{(s)} \end{pmatrix},$$

$v^2 = \sum_{j=1}^r \pi(U_j)^2$ . If  $v^2 < 1$ , then the column MSSA forecast is uniquely defined and can be calculated by the formula

$$R_N = \mathbf{Z}\mathcal{R}_L, \quad \text{where} \quad \mathcal{R}_L = \frac{1}{1-v^2} \sum_{j=1}^r \pi(U_j) \underline{U}_j \in \mathbb{R}^{L-1}. \quad (4.11)$$

Note that (4.11) implies that the forecasting of all individual signals is made using the same LRR which is generated by the whole system.

**Row forecast** Introduce the vectors of the last  $K_m - 1$  values of the reconstructed signals

$$\mathbf{Z}^{(m)} = (\tilde{x}_{N-K_m+2}^{(m)}, \dots, \tilde{x}_{N_m}^{(m)})^T, \quad m = 1, \dots, s,$$

and denote

$$\mathbf{Z} = \begin{pmatrix} \mathbf{Z}^{(1)} \\ \mathbf{Z}^{(2)} \\ \vdots \\ \mathbf{Z}^{(s)} \end{pmatrix}, \quad \mathbf{S} = [\boldsymbol{\mu}(V_1) : \dots : \boldsymbol{\mu}(V_r)].$$

In this notation,  $\mathbf{Z} = \overline{\overline{Y}}_L$ .

If the inverse matrix  $(\mathbf{I}_s - \mathbf{S}\mathbf{S}^T)^{-1}$  exists and  $r \leq K - s$ , then the row MSSA recurrent forecast exists and can be calculated by the formula

$$R_N = \mathcal{R}_K \mathbf{Z}, \quad \text{where} \quad \mathcal{R}_K = (\mathbf{I}_s - \mathbf{S}\mathbf{S}^T)^{-1} \underline{\mathbf{S}\mathbf{V}^T}. \quad (4.12)$$

Note that (4.12) implies that the forecasting of the individual signals is made using the LRRs which are different for different series. The forecasting value generally depends on the last values of all time series from the system of time series.

### Vector MSSA forecasting

Denote  $\underline{\mathcal{L}}^{\text{col}} = \text{span}(\underline{U}_1, \dots, \underline{U}_r)$  and  $\underline{\mathcal{L}}^{\text{row}} = \text{span}(\underline{V}_1, \dots, \underline{V}_r)$ . Let  $\Pi^{\text{col}}$  be the orthogonal projector of  $\mathbb{R}^{L-1}$  on  $\underline{\mathcal{L}}^{\text{col}}$  and  $\Pi^{\text{row}}$  be the orthogonal projector of  $\mathbb{R}^{K-s}$  on  $\underline{\mathcal{L}}^{\text{row}}$ .

An explicit form of the matrices of the column and row projectors is given in Proposition 15. However, the calculation by that formula is time-consuming. A fast algorithm of calculation is presented in Section 4.3.2.

**Column forecast** We have mentioned above that for a given subspace ( $\mathcal{L}^{\text{col}}$  in our case) the column forecast is performed independently for each time series. Define the linear operator

$\mathcal{P}_{\text{Vec}}^{\text{col}} : \mathbb{R}^L \mapsto \mathcal{L}^{\text{col}}$  by the formula

$$\mathcal{P}_{\text{Vec}}^{\text{col}} Z = \begin{pmatrix} \Pi^{\text{col}} \bar{Z} \\ \mathcal{R}_L^T \bar{Z} \end{pmatrix}, \quad (4.13)$$

where  $\mathcal{R}_L$  is defined in (4.11).

The *vector forecasting algorithm* for  $j$ th series is as follows.

1. In the notation above, define vectors  $Z_i$  as follows:

$$Z_i = \begin{cases} \hat{X}_i^{(j)} & \text{for } i = 1, \dots, K_j, \\ \mathcal{P}_{\text{Vec}}^{\text{col}} Z_{i-1} & \text{for } i = K_j + 1, \dots, K_j + M + L - 1. \end{cases} \quad (4.14)$$

2. By constructing the matrix  $\mathbf{Z} = [Z_1 : \dots : Z_{K_j+M+L-1}]$  and making its diagonal averaging we obtain the series  $z_1, \dots, z_{N_j+M+L-1}$ .
3. The numbers  $z_{N_j+1}, \dots, z_{N_j+M}$  form the  $M$  terms of the vector forecast.

**Row forecast** Define the linear operator  $\mathcal{P}_{\text{Vec}}^{\text{row}} : \mathbb{R}^K \mapsto \mathcal{L}^{\text{row}}$  by the formula

$$\mathcal{P}_{\text{Vec}}^{\text{row}} Z = A, \quad (4.15)$$

such that  $\underline{A} = \Pi^{\text{row}} \bar{Z}$  and  $\mu(A) = \mathcal{R}_K \bar{Z}$ , where  $\mathcal{R}_K$  is defined in (4.12).

The *vector forecasting algorithm* is as follows.

1. In the notation above, define vectors  $Z_i$  as follows:

$$Z_i = \begin{cases} \hat{Y}_i & \text{for } i = 1, \dots, L, \\ \mathcal{P}_{\text{Vec}}^{\text{row}} Z_{i-1} & \text{for } i = L + 1, \dots, L + M + K^* - 1, \end{cases} \quad (4.16)$$

where  $K^* = \max(K_i, i = 1, \dots, s)$ .

2. By constructing the matrix  $\mathbf{Z} = [Z_1 : \dots : Z_{L+M+K^*-1}]$  and making Reconstruction step we obtain the series  $z_1^{(j)}, \dots, z_{N_j+M+K^*-1}^{(j)}, j = 1, \dots, s$ .
3. The numbers  $z_{N_j+1}^{(j)}, \dots, z_{N_j+M}^{(j)}, j = 1, \dots, s$ , form the  $M$  terms of the vector forecast.

**Remark 36.** For the  $M$ -step ahead vector forecast,  $M + K^* - 1$  new lagged vectors for the row forecasting and  $M + L - 1$  ones for the column forecasting are constructed. The reason for this is to make the  $M$ -step forecast inheriting the  $(M - 1)$ -step forecast as its part. This specific feature of the vector forecasting provides its stability and accuracy if the accurately extracted component of finite rank is forecasted; that is, if a long-term forecast is appropriate. Otherwise (if the MSSA approximation is inadequate), the long-term vector forecasting can be misleading and even a short-term vector forecasting can be inaccurate for large  $K^*$  or  $L$  correspondingly.

### 4.3.2. Fast vector forecasting algorithm

It is mentioned in [29, p. 76] that the vector forecasting is time-consuming, while recurrent forecasting is fast. However, this is so if one implements Algorithm 1.9 directly. It appears that it is possible to considerably accelerate the vector forecasting. Moreover, in the current implementation in RSSA the vector forecasting is slightly faster than the recurrent one. Let us describe the fast implementation of the vector forecasting algorithm following [12, Section 6.3].

Denote by  $\dagger$  the pseudo-inversion of a matrix.

#### Column MSSA forecast

As it was shown in Section 4.3.1, column vector MSSA forecasting is reduced to performing  $s$  1D vector forecasts in the same subspace  $\mathcal{L}^{\text{col}}$ . Next, we describe the algorithm for fast vector forecasting of 1D time series in a given subspace  $\mathcal{L}^{\text{col}}$  (the fast version of vector forecasting in Basic SSA, see Algorithm 1.9).

Consider the forecasting in the subspace  $\mathcal{L}^{\text{col}}$  given by a basis  $\{P_1, \dots, P_r\}$ . Denote  $\mathbf{P} = [P_1 : \dots : P_r]$ . Each reconstructed vector  $\widehat{X}_k$  of the 1D time series belongs to  $\mathcal{L}^{\text{col}}$ ; hence, there exist coefficients  $W_k \in \mathbb{R}^r$  such that  $\widehat{X}_k = \mathbf{P}W_k$ . Denote  $\mathbf{W} = [W_1 : \dots : W_K]$ . In fact, the input for the algorithm is the minimal decomposition of  $\widehat{\mathbf{X}}$  into the sum of elementary matrices of rank 1 in the form  $\widehat{\mathbf{X}} = \mathbf{P}\mathbf{Q}^\top$  and  $\mathbf{W} = \mathbf{Q}^\top$ .

Note that if we have a singular value decomposition of  $\widehat{\mathbf{X}} = [\widehat{X}_1 : \dots : \widehat{X}_K]$ , then the left singular vectors provide the basis of the subspace, while  $W_k$  are determined by the right singular vectors and singular values:  $\mathbf{P} = [U_1 : \dots : U_r]$  and  $\mathbf{Q} = [\sqrt{\lambda}_1 V_1 : \dots : \sqrt{\lambda}_r V_r]$ .

In vector forecasting, we extend the reconstructed matrix as  $\mathbf{Z} = [\widehat{\mathbf{X}} : Z_{K+1} : \dots : Z_{K+M+L-1}]$ , where the vectors  $Z_k$ ,  $k \geq K+1$ , are obtained as  $Z_k = \mathcal{P}_{\text{Vec}}^{\text{col}} Z_{k-1}$  (and  $Z_k = \widehat{X}_k$  for  $k = 1, \dots, K$ ). Since all  $Z_k$  belong to  $\mathcal{L}^{\text{col}}$ , there exist  $W_k \in \mathbb{R}^r$ ,  $k \geq K+1$  such that

$$\mathbf{Z} = \mathbf{P}[W_1 : \dots : W_{K+M+L-1}].$$

Thus, there exist a matrix  $\mathbf{D}$  such that  $W_k = \mathbf{D}W_{k-1}$  for  $k \geq K+1$ . This observation leads to the following algorithm for vector forecasting.

#### ALGORITHM 4.3: Fast column vector forecasting

*Input:*  $\mathbf{P}, \mathbf{W}$ .

*Output:* the forecasted values  $z_{N+1}, \dots, z_{N+M}$ .

- 1: Compute the matrix  $\mathbf{D} = \underline{\mathbf{P}}^\dagger \overline{\mathbf{P}}$  using the QR-decomposition [173].
- 2: For  $k = K+1, \dots, K+M+L-1$  compute  $W_k = \mathbf{D}W_{k-1}$ .
- 3: Perform the fast rank-one hankelization algorithm [12, Algorithm 2] for the matrix  $\mathbf{Z} = \mathbf{P}[\mathbf{W} : W_{K+1} : \dots : W_{K+M+L-1}]$ , which is explicitly expressed as a sum of rank-one matrices, and obtain

the series  $z_1, \dots, z_{N+M+L-1}$ .

- 4: The numbers  $z_1, \dots, z_{N+M}$  form updated reconstructed and forecasted series.
- 

In order to prove the correctness of the algorithm, it remains to show that the formula  $\mathbf{D} = \underline{\mathbf{P}}^\dagger \bar{\mathbf{P}}$  is correct, which will be discussed at the end of this section. Finally, there is also a further improvement on the computation of  $\mathbf{D}$ .

**Remark 37.** 1. Item 1 in Algorithm 4.3 provides exactly the shift matrix from the LS-ESPRIT method for frequency estimation briefly described in Section 1.9.2.

2. If  $\{P_i\}$  is orthonormal system, then  $\mathbf{D}$  can be computed without the QR decomposition as:

$$\mathbf{D} = \left( \mathbf{I}_r - \frac{1}{1 - \boldsymbol{\pi}^\top \boldsymbol{\pi}} \boldsymbol{\pi} \boldsymbol{\pi}^\top \right) \underline{\mathbf{P}}^\top \bar{\mathbf{P}},$$

where  $\boldsymbol{\pi} = \boldsymbol{\pi}(\mathbf{P})$ ,  $\mathbf{I}_r$  is the  $r \times r$  identity matrix.

## Row MSSA forecast

Row vector forecast is slightly different from the column one, but the idea is the same. We deal with forecasting in the row subspace  $\mathcal{L}^{\text{row}} = \text{span}(Q_1, \dots, Q_r)$  and continue the sequence of the row vectors  $\widehat{Y}_k$  of  $\widehat{\mathbf{X}}$ . If the row vectors are equal to  $\widehat{Y}_k = \mathbf{Q}W_k$ ,  $k = 1, \dots, L$ , then  $\widehat{\mathbf{X}}^\top = \mathbf{Q}\mathbf{W}$ , where  $\mathbf{W} = [W_1 : \dots : W_L]$ . In the following algorithm, as in column forecasting, the vectors  $W_k$  are continued instead of the vectors  $Y_k$ .

---

### ALGORITHM 4.4: Fast row vector forecasting

*Input:*  $\mathbf{Q}, \mathbf{W}$ .

*Output:* the forecasted values  $z_{N_p+1}^{(p)}, \dots, z_{N_p+M}^{(p)}$ ,  $p = 1, \dots, s$ .

- 1: Compute the matrix  $\mathbf{D} = \underline{\underline{\mathbf{Q}}}^\dagger \bar{\bar{\mathbf{Q}}}$  using the QR decomposition.
  - 2: For  $k = L+1, \dots, L+M + \max_{p=1, \dots, s} K_p - 1$  compute  $W_k = \mathbf{D}W_{k-1}$ .
  - 3: Perform the fast rank-one hankelization algorithm [12, Algorithm 2] for each of  $s$  matrices  $\mathbf{Z}^{(p)} = \mathbf{Q}^{(p)}[\mathbf{W} : W_{L+1} : \dots : W_{L+M+K_p-1}]$  for  $p = 1, \dots, s$  and obtain  $s$  series  $z_1^{(p)}, \dots, z_{N_p+M+L-1}^{(p)}$ .
  - 4: The numbers  $z_1^{(p)}, \dots, z_{N_p+M}^{(p)}$  form updated reconstructed and forecasted series.
- 

**Remark 38.** If  $\{Q_i\}$  is orthonormal system, then  $\mathbf{D}$  from Algorithm 4.4 can be expressed as

$$\mathbf{D} = \left( \mathbf{I}_r - \mathbf{S}^\top (\mathbf{I}_s - \mathbf{S}\mathbf{S}^\top)^{-1} \mathbf{S} \right) \underline{\underline{\mathbf{Q}}}^\top \bar{\bar{\mathbf{Q}}},$$

where  $\mathbf{S} = [\boldsymbol{\mu}(Q_1) : \dots : \boldsymbol{\mu}(Q_r)]$ ,  $\mathbf{I}_r$  and  $\mathbf{I}_s$  are the  $r \times r$  and  $s \times s$  identity matrices.

## Proof of the algorithms correctness

For simplicity, we will consider the one-dimensional case, that is, SSA vector forecast, which coincides with the column MSSA forecast for one-dimensional series. The proof for row forecasting is analogous.

We need to prove that  $Z_{k+1} = \mathbf{P}W_{k+1}$ , where  $W_{k+1} = \mathbf{D}W_k$ ,  $\mathbf{D} = \underline{\mathbf{P}}^\dagger \bar{\mathbf{P}}$ . It is sufficient to prove  $\underline{Z}_{k+1} = \underline{\mathbf{P}}W_{k+1}$ , since the last coordinate of the vector is uniquely defined. In the standard formulation of vector forecasting algorithm given in Algorithm 1.9,  $\underline{Z}_{k+1}$  is the projection on the column space of  $\underline{\mathbf{P}}$ , that is,  $\underline{Z}_{k+1} = \underline{\mathbf{P}}\underline{\mathbf{P}}^\dagger \bar{Z}_k$ . Since  $\bar{\mathbf{P}}W_k$  is exactly  $\bar{Z}_k$  by definition of  $W_k$ , we have that  $\underline{Z}_{k+1} = \underline{\mathbf{P}}(\underline{\mathbf{P}}^\dagger \bar{\mathbf{P}})W_k = \underline{\mathbf{P}}W_{k+1}$ , and the equivalence of the standard and fast vector forecasting algorithms is proved.

**Remark 39.** *The speedup of the algorithms implementation is explained by two reasons:*

1. *multiplication by matrices of small size  $r \times r$  at each step instead of multiplication by matrices of much larger size, and,*
2. *the form of the matrix to be hankelized is suitable for application of the fast rank-one hankelization algorithm.*

### 4.3.3. Simulated example: numerical comparison

In this section, we demonstrate how the accuracy of MSSA is related to the structure of the multivariate time series. The aim is to compare accuracy for separate analysis and forecasting of time series with simultaneous processing of the series system.

In the study below, we consider the case  $s = 2$  and examine the following SSA methods: (a) 1D-SSA applied twice and (b) MSSA. The investigated model examples include the least favorable and the most favorable cases for MSSA.

Let us assume that we observe  $(\mathbf{X}^{(1)}, \mathbf{X}^{(2)}) = (\mathbf{H}^{(1)}, \mathbf{H}^{(2)}) + (\mathbf{N}^{(1)}, \mathbf{N}^{(2)})$ , where  $(\mathbf{H}^{(1)}, \mathbf{H}^{(2)})$  is a two-dimensional signal consisting of two harmonic time series,  $\mathbf{N}^{(1)}$  and  $\mathbf{N}^{(2)}$  are realizations of independent white Gaussian noises. Then we can use the standard simulation techniques to obtain estimates of the mean square errors (MSE) for the reconstruction and forecasting of  $(\mathbf{H}^{(1)}, \mathbf{H}^{(2)})$  by the indicated SSA methods. The resultant MSE is calculated as the mean of  $\text{MSE}^{(1)}$  and  $\text{MSE}^{(2)}$  for  $\mathbf{H}^{(1)}$  and  $\mathbf{H}^{(2)}$  correspondingly.

We take the following parameters for the simulation of the time series:  $N = 71$ , the variance of all noise components is  $\sigma^2 = 25$ , the number of replications is 10000. We consider the following two versions of the signal  $(\mathbf{H}^{(1)}, \mathbf{H}^{(2)})$ .

**Example A** (the same periods; the difference between the phases is different from  $\pi/2$ ):

$$h_k^{(1)} = 30 \cos(2\pi k/12), \quad h_k^{(2)} = 20 \cos(2\pi k/12 + \pi/4), \quad k = 1, \dots, N.$$

**Example C** (different periods):

$$h_k^{(1)} = 30 \cos(2\pi k/12), \quad h_k^{(2)} = 20 \cos(2\pi k/8 + \pi/4), \quad k = 1, \dots, N.$$

The choice of these examples is determined by the observation that the dimensions of the signal trajectory spaces (i.e., ranks) are different for different extensions of the 1D-SSA method, see Table 4.1. For each example the rank in blue corresponds to the method with the best accuracy for this example. Cells in the row corresponding to 1D-SSA contain one number, since the ranks of the times series from the considered collections coincide. The estimation was performed by 10000 repetitions.

Table 4.1: Dimension of the signal trajectory space.

	Example A	Example C
MSSA	<b>2</b>	4
1D-SSA	2	<b>2</b>

The results of investigation for different window lengths  $L$  are summarized in Tables 4.2 and 4.3. The 24 term-ahead forecast was performed. For each example, the cells corresponding to the method with the reconstruction/forecast accuracy, which is closest to the best one, are shown in bold and the overall minimum is in blue color.

Table 4.2: MSE of signal reconstruction.

Example A	$L = 12$	$L = 24$	$L = 36$	$L = 48$	$L = 60$
MSSA	3.18	1.83	1.59	<b>1.47</b>	2.00
1D-SSA	3.25	<b>2.01</b>	<b>2.00</b>	<b>2.01</b>	3.25
Example C	$L = 12$	$L = 24$	$L = 36$	$L = 48$	$L = 60$
MSSA	6.91	3.77	3.07	<b>2.88</b>	3.84
1D-SSA	3.23	<b>2.01</b>	<b>2.00</b>	<b>2.01</b>	3.23

Comparison of Tables 4.2 and 4.3 with Table 4.1 clearly demonstrates the relation between the accuracy of the signal reconstruction (forecast) and the dimension of the signal trajectory space.

Note that the reconstructions by 1D-SSA are the same for window lengths  $L$  and  $N - L + 1$  (12 and 60, 24 and 48 for the considered examples). Reconstructions by MSSA are different for different  $L$ . Also note that the trajectory matrix for 1D-SSA has rank  $\min(L, N - L + 1)$  and the rank is maximal for  $L \approx N/2$ . The MSSA-trajectory matrix has rank equal to  $\min(L, (N - L + 1)s)$ , where  $s$  is the number of time series in the system. This rank is maximal for  $L \approx sN/(s + 1)$ . Although the maximality of the rank does not guarantee the minimality of errors, this consideration means that to achieve better separability the choice of the window length  $L$  larger than  $N/2$  can often be

Table 4.3: MSE of signal forecast.

Example A	$L = 12$	$L = 24$	$L = 36$	$L = 48$	$L = 60$
<b>Recurrent</b>					
MSSA-column	5.36	<b>3.67</b>	3.73	3.70	4.43
MSSA-row	6.02	4.25	3.83	<b>3.32</b>	3.98
1D-SSA	7.24	<b>5.59</b>	6.30	6.42	7.93
<b>Vector</b>					
MSSA-column	5.93	3.77	3.62	<b>3.11</b>	3.65
MSSA-row	4.00	<b>3.03</b>	3.39	3.17	4.24
1D-SSA	7.74	5.43	5.85	<b>5.14</b>	6.76
Example C	$L = 12$	$L = 24$	$L = 36$	$L = 48$	$L = 60$
<b>Recurrent</b>					
MSSA-column	25.76	<b>7.39</b>	7.55	7.43	9.00
MSSA-row	19.82	8.47	8.00	<b>6.66</b>	8.30
1D-SSA	7.36	<b>5.61</b>	6.28	6.44	8.00
<b>Vector</b>					
MSSA-column	25.34	7.56	7.57	<b>6.20</b>	7.67
MSSA-row	57.59	<b>6.04</b>	7.03	6.30	8.69
1D-SSA	7.84	5.47	5.84	<b>5.18</b>	6.88

recommended. Simulations confirm this: the minimum of the reconstruction error for MSSA is achieved at  $L = 48 = 72 \times 2/3$ .

The forecasting errors have much more complicated structure, see Section 3.4. In particular, these errors for forecasting depend on the reconstruction errors for the last time series points; therefore, the error may have a dependence on  $L$ , which is different from that for the average reconstruction errors. The considered examples show that the vector forecast is more accurate than the recurrent one and that the row MSSA forecast is slightly more accurate than the column MSSA forecast.

The considered examples confirm the following assertions:

- The accuracy of the SSA-based methods is closely related to the structure of the signal trajectory spaces generated by these methods. MSSA has an advantage if time series from the system have matched components. (Note that we considered equal levels of noise.)
- Optimal window lengths for analysis and forecasting can differ. Despite the accuracy of forecast is related to the accuracy of reconstruction, this relation is not straightforward.
- The vector forecast with the best window length is more accurate than the recurrent forecast. However, it is not always the case if we compare forecast accuracies for the same window length. This is probably valid for forecasting of well-separated signal of finite rank only, see Remark 36.
- In MSSA, the recommendations for the choice of the window length (e.g., ‘take  $L$  larger

(or smaller) than the half of the time series length') for recurrent forecasting are in a sense opposite to that for the vector forecasting.

- For the row and column forecasting (1D-SSA forecasting methods are particular cases of the column forecasting), the recommendations are also opposite. This is not surprising since  $L$  and  $K$  have swapped places in MSSA relative to 1D-SSA.

#### 4.4. Automation of grouping in MSSA

For MSSA, we propose a generalization of the low-frequency method from Section 2.5.1, the frequency method presented in Section 2.5.2 and the method by the regularity of angles described in Section 2.5.3 for identifying the oscillatory component [129].

Consider a multivariate time series  $\mathbf{X} = (\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(s)})$ ,  $\mathbf{X}^{(p)} = (x^{(p)}(1), \dots, x^{(p)}(N_p))$ ,  $p = 1, \dots, s$ ,  $x^{(p)}(i) \in \mathbb{R}$ .

The main feature of MSSA, unlike the real or complex case, is that all three kinds of objects: elementary reconstructed series, left singular vectors, and right singular vectors, have different structure.

It follows from the description of the MSSA algorithm that the reconstructed series have the same form as the original multivariate series, i.e. they are also multivariate series (or, the same, a system of time series).

It follows from Proposition 21 for MSSA that the left singular vectors are one-dimensional real-valued time series of length  $L$ . They describe the general structure of the series  $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(s)}$  (the structure of the column space of the trajectory matrix). They have the same form as singular vectors for a single time series.

The right singular vectors are also one-dimensional real-valued time series; they describe the structure of the row space of the trajectory matrix; each row consists of elements of all series  $\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(s)}$ , not just one. We will use the splitting of the right singular vectors  $V_i$  into parts  $V_i^{(m)}$ ,  $m = 1, \dots, s$ , introduced in (4.6).

Thereby, the algorithms for MSSA automation have different modifications for different variants of the input objects. Different modifications of the algorithms can produce different results.

##### Low-frequency method for trend identification

Let us generalize Algorithm 2.13 for trend identification algorithm to the case of multivariate series. Algorithm 2.13 can be applied in 1D-SSA either to left or right singular vectors or to elementary reconstructed series in the same manner. As already mentioned, in MSSA the left singular vectors are one-dimensional real-valued time series of length  $L$  and have the same form as singular vectors for one series. That is, the left singular vectors have the same form as in the case

of 1D-SSA, and Algorithm 2.13 of the low-frequency method can be applied to them in the same way. Algorithm 4.5 for this case is given below.

The right singular vectors are one-dimensional real-valued time series of length  $K$  and are represented as (4.6). We can apply the low-frequency method to each parts  $V_i^{(1)}, \dots, V_i^{(s)}$  of the right singular vector  $V_i$ . Consider the measure (2.20) for each part  $V_i^{(1)}, \dots, V_i^{(s)}$  of vector  $V_i$ , and take the maximum. The proposed Algorithm 4.6 is formally described below.

The elementary reconstructed series in the case of MSSA are multivariate time series. The idea of applying the low-frequency method to them is the same as for right singular vectors: in the first step, for each of  $s$  series, consider the measure by the formula (2.20) and take the maximum value of these measures to use in the second step. This variant of the method is also given by the algorithm 4.6, the only difference is that elementary reconstructed series are taken as inputs.

**ALGORITHM 4.5:** MSSA. Low-frequency method for trending: version with left singular vectors

*Input:* The following data and parameters are input.

1. **Data:** index group  $I \in \{1, \dots, d\}$ ; frequency  $0 \leq \omega \leq 0.5$ ; left singular vectors  $U_i, i \in I$ .
2. **Parameters:** threshold  $0 \leq T_0 \leq 1$ .

*Output:* A group of indices  $J \subset I$  of components related to the trend.

- 1: For each vector  $U_i, i \in I$ , we calculate the value  $T(U_i; \omega)$  using (2.20).
- 2:  $J$  is the group of indices  $i \in I$  such that  $T(U_i; \omega) \geq T_0$ .

**ALGORITHM 4.6:** MSSA. Low-frequency method for trending: version with right singular vectors or elementary reconstructed series

*Input:* The following data and parameters are input.

1. **Data:** index group  $I \in \{1, \dots, d\}$ ; frequency  $0 \leq \omega \leq 0.5$ ; series  $Y_i, i \in I$ : right singular vectors or elementary reconstructed series.
2. **Parameters:** threshold  $0 \leq T_0 \leq 1$ .

*Output:* A group of indices  $J \subset I$  of components related to the trend.

- 1: For each series  $Y_i = (Y_i^{(1)}, \dots, Y_i^{(s)})$ ,  $i \in I$ , we calculate the values  $T(Y_i^{(1)}; \omega), \dots, T(Y_i^{(s)}; \omega)$  using (2.20).  $T_m := \max\{T(Y_i^{(1)}; \omega), \dots, T(Y_i^{(s)}; \omega)\}$ .
- 2:  $J$  is the group of indices  $i \in I$  such that  $T_m \geq T_0$ .

### Frequency method of identification of the oscillatory component

In the multivariate case, we will also assume that the oscillatory component of each of the series is the sum of the e-m harmonics, i.e. the elements of the multivariate e.m. harmonic series  $S$

have the form (4.9).

In MSSA, the left singular vectors are one-dimensional real-valued time series of length  $L$  and have the same form as singular vectors for one series (see Section 4.2). Therefore, the frequency method described in Section 2.5.2 can be applied to them in the same way as in the case of 1D-SSA. Below is the algorithm 4.7 of the frequency method for MSSA as applied to left singular vectors.

**ALGORITHM 4.7:** MSSA. Frequency method for the oscillatory component: version with left singular vectors

*Input:* The following data and parameters are input.

1. **Data:** left singular vectors  $\{U_i\}_{i=1}^d$ .
2. **Parameters:** parameter  $s_0 \in \mathbb{Z}_+$ , threshold  $\rho_0 \in [0, 1]$ .

*Output:* A group of component indices  $J^{(P)}$  related to the oscillatory component.

- 1: Based on  $\{U_i\}_{i=1}^d$ , obtain the index group  $J_1^{(P)}$  using (2.21) with  $s_0$  and the index group  $J_2^{(P)}$  using (2.22) with  $s_0$ .
- 2: Based on  $\{U_j\}_{j \in J_1^{(P)} \cup J_2^{(P)}}$ , get the index group  $J^{(P)}$  using (2.25) with  $\rho_0$ .

As mentioned above, the right singular vectors  $V_i$  have the form (4.6). Since the vectors to which we apply the frequency method must have norm 1, together with vectors  $V_i^{(p)}$  we consider the normalized vectors  $\widehat{V}_i^{(p)} := \frac{V_i^{(p)}}{\|V_i^{(p)}\|}$ .

In the first step of the method, the sets of indices  $J_{1,p}^{(P)}$ ,  $p = 1, \dots, s$ , is calculated:

$$J_{1,p}^{(P)} = \{i : \theta_i^{(p)} \theta_{i+1}^{(p)} > 0, L|\theta_i^{(p)} - \theta_{i+1}^{(p)}| \leq s_0, 1 \leq i \leq d-1\}, \quad (4.17)$$

where  $\theta_i^{(p)} = \arg \max_{0 < k \leq K_p/2} \{\Pi_{\widehat{V}_i^{(p)}}^{K_p}(k/K_p)\}$ ,  $p = 1, \dots, s$ . Hereby, for each  $p = 1, \dots, s$ , the sets  $J_{1,p}^{(P)}$  are computed for each pair of sequential vectors  $\widehat{V}_i^{(p)}$  and  $\widehat{V}_{i+1}^{(p)}$  of length  $K_p$ .

Similarly, each singular vector is checked against a harmonic with frequency 0.5:

$$J_{2,p}^{(P)} = \{i : K_p |\theta_i^{(p)} - 0.5| \leq s_0, 1 \leq i \leq d\}. \quad (4.18)$$

In the second step, we combine the sets of indices:  $J_1^{(P)}$  is the union of the sets  $\{J_{1,p}^{(P)}\}_{p=1}^s$ ,  $J_2^{(P)}$  is the union of the sets  $\{J_{2,p}^{(P)}\}_{p=1}^s$ .

For the second step, we introduce the definition

$$\rho_{i,j} := \max_p \left\{ \max_{0 < k \leq K_p/2} \left( \rho_{\{\widehat{V}_i^{(p)}, \widehat{V}_j^{(p)}\}}(k/K_p) + \rho_{\{\{\widehat{V}_i^{(p)}, \widehat{V}_j^{(p)}\}\}}((k+1)/K_p) \right), p = 1, \dots, s \right\},$$

where  $\widehat{V}_i^{(p)}$  and  $\widehat{V}_j^{(p)}$  are the vectors with indices from the set  $J_1^{(P)}$  constructed in the first step.

Definition 9 of the measure  $\rho_A$  is given in Section 2.5.2.

To identify the components belonging to a harmonic with a frequency 0.5, the measure has the form

$$\rho_i := \max_p \left\{ \max_{0 \leq k \leq K_p/2} \left( \rho_{\{\widehat{V}_i^{(p)}\}}(\lfloor K_p/2 \rfloor / K_p) + \rho_{\{\{\widehat{V}_i^{(p)}\}\}}((\lfloor K_p/2 \rfloor + 1) / K_p) \right), p = 1, \dots, s \right\},$$

where  $i \in J_2^{(P)}$ .

The final result, see Algorithm 4.8, are the indices

$$J^{(P)} = \{(i, j) : \rho_{i,j} \geq \rho_0\} \cup \{i : \rho_i \geq \rho_0\}. \quad (4.19)$$

ALGORITHM 4.8: MSSA. Frequency method for the oscillatory component: version with right singular vectors

*Input:* The following data and parameters are input.

1. **Data:** right singular vectors  $\{V_i\}_{i=1}^d$ .
2. **Parameters:** parameter  $s_0 \in \mathbb{Z}_+$ , threshold  $\rho_0 \in [0, 1]$ .

*Output:* A group of indices  $J^{(P)}$  of component related to the oscillatory component.

- 1: Based on  $\{V_i\}_{i=1}^d$ , obtain the index group  $J_{1,p}^{(P)}$ ,  $p = 1, \dots, s$ , using (4.17) with  $s_0$  and the index group  $J_{2,p}^{(P)}$ ,  $p = 1, \dots, s$ , using (4.18) with  $s_0$ ;  $J_1^{(P)}$  is the union of the sets  $\{J_{1,p}^{(P)}\}_{p=1}^s$ ,  $J_2^{(P)}$  is union of the sets  $\{J_{2,p}^{(P)}\}_{p=1}^s$ .
- 2: The index group  $J^{(P)}$  by  $\{V_j\}_{j \in J_1^{(P)} \cup J_2^{(P)}}$  is obtained using (4.19) with  $\rho_0$ .

### Method for identifying the oscillatory component by the regularity of angles

Again, if we are considering left singular vectors, the algorithm can be applied in the same way as in the 1D-SSA case (Section 2.5.3). The algorithm 4.9 for this case is given below.

ALGORITHM 4.9: MSSA. The angle regularity identification method for the oscillatory component: the left singular vector variant

*Input:* The following data and parameters are input.

1. **Data:** number of components  $r$ ; left singular vectors  $\{U_j\}_{j=1}^r$ .
2. **Parameters:** number of e-m harmonics  $m \leq r/2$ , or threshold  $t_0 \geq 0$ , depending on the stopping criterion.

*Output:* A group of  $J$  singular vector indices related to the oscillatory component.

- 1: Based on  $\{U_i\}_{i=1}^r$ , compute the values  $\tilde{\tau}(U_j, U_{j+1})$ ,  $j = 1, \dots, r-1$  using (2.27) and order them in ascending order. While calculating for each  $j = 2, \dots, r-1$  we check if  $\tilde{\tau}(U_j, U_{j+1}) < \tilde{\tau}(U_{j-1}, U_j)$ , then we drop  $\tilde{\tau}(U_{j-1}, U_j)$  from consideration; otherwise, we drop  $\tilde{\tau}(U_j, U_{j+1})$ . The resulting values are  $\tau_1, \dots, \tau_{[r/2]}$ .
- 2: Two versions of the stopping criterion are as follows. Try the elements of the set  $i = 1, \dots, [r/2]$ 
  1. either until  $i < m$ ;
  2. or until  $\tau_i < t_0$ .

Denote  $i_0$  the moment of stopping.

- 3: The set  $J$  consists of the indices  $j$ ,  $j+1$  of singular vectors  $U_j$  and  $U_{j+1}$  involved in the computation of the values  $\tau_1, \dots, \tau_{i_0-1}$ .
- 

The method can be applied to the right singular vectors in the same way as it was for the frequency method. We calculate the values  $\tilde{\tau}(V_j^{(p)}, V_{j+1}^{(p)})$ ,  $p = 1, \dots, s$ , for a consecutive pair of right singular vectors  $V_j = (V_j^{(1)}, \dots, V_j^{(s)})$  and  $V_{j+1} = (V_{j+1}^{(1)}, \dots, V_{j+1}^{(s)})$ ,  $j = 1, \dots, r-1$ , whose form is defined by the formula (4.6). Then for each  $j$  we take the minimum value  $\tilde{\tau}^{(\min)}(V_j, V_{j+1}) = \min_p \{\tilde{\tau}(V_j^{(p)}, V_{j+1}^{(p)}), p = 1, \dots, s\}$  and to the already obtained values  $\tilde{\tau}^{(\min)}(V_j, V_{j+1})$  we apply the standard algorithm as in the one-dimensional case. The resulting algorithm 4.10 is shown below.

---

**ALGORITHM 4.10: MSSA.** Method of identification by the regularity of angles for the oscillatory component: version with right singular vectors

---

*Input:* The following data and parameters are input.

1. **Data:** number of components  $r$ ; right singular vectors  $\{V_j\}_{j=1}^r$ .
2. **Parameters:** number of e-m harmonics  $m \leq r/2$ ; or threshold  $t_0 \geq 0$ , depending on the stopping criterion.

*Output:* A group of indices of singular vectors  $J$  related to the oscillatory component.

- 1: For each pair of vectors  $V_j = (V_j^{(1)}, \dots, V_j^{(s)})$  and  $V_{j+1} = (V_{j+1}^{(1)}, \dots, V_{j+1}^{(s)})$  compute the values  $\tilde{\tau}(V_j^{(p)}, V_{j+1}^{(p)})$  using (2.27),  $j = 1, \dots, r-1$ ,  $p = 1, \dots, s$
- 2: Compute  $\tilde{\tau}^{(\min)}(V_j, V_{j+1}) = \min_p \{\tilde{\tau}(V_j^{(p)}, V_{j+1}^{(p)}), p = 1, \dots, s\}$ .
- 3: For each  $j = 2, \dots, r-1$ , if  $\tilde{\tau}^{(\min)}(V_j, V_{j+1}) < \tilde{\tau}^{(\min)}(V_{j-1}, V_j)$ , then we drop the  $\tilde{\tau}^{(\min)}(V_{j-1}, V_j)$  from consideration; otherwise, we drop  $\tilde{\tau}^{(\min)}(V_j, V_{j+1})$ . Arrange these values in ascending order and obtain  $\tau_1, \dots, \tau_{[r/2]}$ .
- 4: Two versions of the stopping criterion are as follows. Try the elements of the set  $i = 1, \dots, [r/2]$ 
  1. either until  $i < m$ ;
  2. or until  $\tau_i < t_0$ .

Denote  $i_0$  the moment of stopping.

- 5: The set  $J$  consists of the indices  $j, j+1$  of singular vectors  $U_j$  and  $U_{j+1}$  involved in the computation of the values  $\tau_1, \dots, \tau_{i_0-1}$ .
-

## Chapter 5

# Shaped and multidimensional SSA

This chapter is devoted to extensions of 1D-SSA (Chapters 2 and 3) and MSSA (Chapter 4) for the analysis of objects of dimension 2 and larger. The 2D case corresponds to digital image processing. Objects with larger dimensions are also widely used. For example, a color image can be considered as a system of 2D images and its analysis can be performed by multivariate 2D-SSA, which is an extension of MSSA designed for analysing a system of series. The third temporal dimension naturally arises if images are changing in time, which is typical data in climatology [171]. The data can be 3D, if the measurements are performed at points located in a 3D area, see e.g. Section 7.5 for quantitative analysis of gene expression data. A system of 3D data changing in time can also be considered. The list of extensions can be continued; see e.g. [75] for high-dimensional SSA algorithms applied to specific problems in seismology.

The scheme of SSA-family methods of Section 1.2 can be naturally applied for multidimensional objects of any dimension since the difference is in the embedding operator only. Moreover, the SSA analysis of an object of any dimension and shape is a particular case of the shaped version of SSA, where a window of arbitrary shape goes through an object of arbitrary shape and forms a trajectory matrix consisting of vectorized lagged windows.

The RSSA package implements the so-called Shaped nD-SSA for the analysis of objects of arbitrary dimensions, in rectangular and shaped versions. In particular, 2D or 3D images, which are not necessarily rectangular and are changing over time (that is, have an additional temporary dimension) can be analyzed.

This chapter is based on the long paper [12] and [41]. We start the chapter with a description of the general approach called Shaped SSA to the analysis of objects of arbitrary form, that is, shaped objects (Section 5.1).

In Section 5.2, a variety of SSA-family methods are considered as particular cases of Shaped SSA such as Shaped 1D-SSA, MSSA, 2D-SSA and nD-SSA.

Finally, we demonstrate an example of Shaped SSA in Section 5.3.

We do not formally write the algorithms for dimensions larger than two, since they can be obtained by a formal extension of the algorithms from Section 5.1.

### 5.1. Shaped 2D-SSA

In its general form, Shaped SSA can be used for analyzing an object given on a grid of dimension  $k \geq 1$ . The case  $k = 1$ , which is described in Section 5.2.1, is, to a certain extent, degenerate. For simplicity of notation, we consider the case  $k = 2$  as a representative for dimensions  $k > 1$ .

*Shaped 2D-SSA* (we will use the short name *ShSSA* if the dimension is clear from the context)

is a generalization of 2D-SSA, which allows arbitrary shapes of the input array and the window. This considerably extends the range of real-life applications of SSA, since it allows analysing of parts of the image with different structures separately, exclude areas with corrupted data, analyze images with gaps, decompose non-rectangular images, etc. In Shaped SSA, not all values of the rectangular image have to be specified, and the sliding window is not necessarily rectangular. Moreover, there is a circular version of Shaped SSA (Circular SSA), when the object is located on a circle, a cylinder or a torus [41].

### 5.1.1. Method

#### Shapes and actions with them

Formally, we call *shape*  $\mathfrak{B}$  a finite non-empty subset of  $\{1, \dots, T_x\} \times \{1, \dots, T_y\}$ , where  $T_x, T_y \in \mathbb{N} \cup \{\infty\}$ ,  $\mathbb{N}$  is the set of natural numbers. The values  $T_x$  and  $T_y$  characterize the topology of the set containing this shape. If  $T_x < \infty$  (or  $T_y < \infty$ ), then the topology is circular and the shapes are cyclic with respect to the  $x$ -coordinate (or the  $y$ -coordinate) with period  $T_x$  (respectively,  $T_y$ ).

A  $\mathfrak{B}$ -shaped array is a indexed set  $X = X_{\mathfrak{B}} = (x_{(i,j)})_{(i,j) \in \mathfrak{B}}$ . Let us denote the space of  $\mathfrak{B}$ -shaped arrays as  $\mathbb{R}^{\mathfrak{B}}$ . There is an isomorphism  $\mathbb{R}^{\mathfrak{B}} \sim \mathbb{R}^B$ , where  $B = |\mathfrak{B}|$  is the cardinality of the set  $\mathfrak{B}$ . This isomorphism is not unique, since the elements in the shape can be ordered in different ways. However, the result of the algorithm does not depend on the chosen order.

For convenience, we consider the lexicographical order, which fixes the isomorphism

$$\mathcal{J}_{\mathfrak{B}} : \mathbb{R}^{\mathfrak{B}} \mapsto \mathbb{R}^B. \quad (5.1)$$

We call  $\mathcal{J}_{\mathfrak{B}}$  *vectorization* and its inverse  $\mathcal{J}_{\mathfrak{B}}^{-1}$  *shaping*.

Next, we introduce the operation of addition in the considered topology for two pairs of indices  $\ell = (\ell_x, \ell_y)$ ,  $\kappa = (\kappa_x, \kappa_y)$ . For convenience, we omit the parameters  $(T_x, T_y)$  and write  $\oplus$  instead of  $\oplus^{(T_x, T_y)}$ :

$$\ell \oplus \kappa = ((\ell_x + \kappa_x - 2) \bmod T_x + 1, (\ell_y + \kappa_y - 2) \bmod T_y + 1),$$

where  $\bmod$  denotes the remainder in the integer division. Formally,  $a \bmod \infty = a$  for any  $a$ . Note that  $-2$  and  $+1$  in the definition of  $\oplus$  is the consequence of the indexing, which starts at 1 (so that  $\{1\} \oplus \{1\} = \{1\}$ ).

For two shapes  $\mathfrak{A}$  and  $\mathfrak{B}$ , we modify the definition of the Minkowski sum in the following way:

$$\mathfrak{A} \oplus \mathfrak{B} = \{\alpha \oplus \beta \mid \alpha \in \mathfrak{A}, \beta \in \mathfrak{B}\}. \quad (5.2)$$

#### Embedding step

The embedding operator for the Shaped 2D-SSA algorithm is defined as follows.

*Input data and parameters of the embedding.* The input data are the topology characteristics  $(T_x, T_y)$ , the shape  $\mathfrak{N}$ , where  $\mathfrak{N} \subset \{1, \dots, T_x\} \times \{1, \dots, T_y\}$ , and the  $\mathfrak{N}$ -shaped array  $\mathbf{X} \in \mathbb{R}^{\mathfrak{N}}$ . The parameter of the algorithm is a *window shape*  $\mathfrak{L} \subset \mathfrak{N}$ . It is convenient to consider the window shape in the form  $\mathfrak{L} = \{\ell_1, \dots, \ell_L\}$ , where  $L = |\mathfrak{L}|$  and  $\ell_i \in \mathbb{N}^2$  are ordered lexicographically.

For each  $\kappa \in \mathfrak{N}$ , we define a shifted  $\mathfrak{L}$ -shaped subarray as  $\mathbf{X}_{\mathfrak{L} \oplus \{\kappa\}} = (x_\alpha)_{\alpha \in \mathfrak{L} \oplus \{\kappa\}}$ . The index  $\kappa$  is a position of the origin for the window. Consider the set of all possible origin positions for the  $\mathfrak{L}$ -shaped windows:

$$\mathfrak{K} = \{\kappa \in \mathfrak{N} \mid \mathfrak{L} \oplus \{\kappa\} \subset \mathfrak{N}\}.$$

We assume that  $\mathfrak{K}$  is written as  $\mathfrak{K} = \{\kappa_1, \dots, \kappa_K\}$ , where  $K = |\mathfrak{K}| \neq 0$  and  $\kappa_j$  are ordered lexicographically.

If the shapes  $\mathfrak{N}$  and  $\mathfrak{L}$  are rectangles, then  $\mathfrak{K}$  is also rectangular and we call the version of Shaped 2D-SSA *rectangular*. Note that the rectangular version of Shaped 2D-SSA is exactly the ordinary 2D-SSA described in Section 5.2.3.

*Embedding operator.* The trajectory matrix  $\mathbf{X}$  is constructed by the embedding operator  $\mathcal{T}_{\text{ShSSA}} : \mathbb{R}^{\mathfrak{N}} \rightarrow \mathbb{R}^{L \times K}$

$$\mathcal{T}(\mathbf{X}) = \mathcal{T}_{\text{ShSSA}}(\mathbf{X}) := \mathbf{X} = [X_1, \dots, X_K], \quad (5.3)$$

where the columns

$$X_j = (x_{\ell_i \oplus \kappa_j})_{i=1}^L$$

are vectorizations of the shaped subarrays  $\mathbf{X}_{\mathfrak{L} \oplus \{\kappa_j\}}$ .

The embedding operator  $\mathcal{T}_{\text{ShSSA}}$  is linear. Denote its range  $\mathcal{M}_{L,K}^{(H)}$  as  $\mathsf{H}^{\mathfrak{L} \times \mathfrak{K}} \subset \mathbb{R}^{L \times K}$ . The subspace  $\mathsf{H}^{\mathfrak{L} \times \mathfrak{K}}$  consists of structured matrices, which we will call *quasi-Hankel*. (In fact, they are generalizations of quasi-Hankel matrices from [174].) If the operator  $\mathcal{T}_{\text{ShSSA}}$  is injective, then it sets the isomorphism between the spaces  $\mathbb{R}^{\mathfrak{N}}$  and  $\mathsf{H}^{\mathfrak{L} \times \mathfrak{K}}$ .

**Remark 40.** *The embedding operator  $\mathcal{T}_{\text{ShSSA}}$  is injective if and only if  $\mathfrak{L} \oplus \mathfrak{K} = \mathfrak{N}$ , i.e., if each point of the initial shaped array  $\mathbf{X}$  can be covered by a window of shape  $\mathfrak{L}$ .*

If there are uncovered points, we can remove them and consider only the decomposition of the restricted array  $\mathbf{X}' = (\mathbf{X})_{\mathfrak{N}'}$ , where  $\mathfrak{N}' = \mathfrak{L} \oplus \mathfrak{K}$ . Hereafter we will suppose that  $\mathfrak{N} = \mathfrak{L} \oplus \mathfrak{K}$ , i.e., all points are covered and the operator  $\mathcal{T}_{\text{ShSSA}}$  is injective.

**Projection to  $\mathsf{H}^{\mathfrak{L} \times \mathfrak{K}}$**  Projection  $\Pi_{\mathfrak{N} \rightarrow \mathfrak{K}}$  to the space of quasi-Hankel matrices is generated by the embedding operator  $\mathcal{T}_{\text{ShSSA}}$  and can be performed using the general form described in Section 1.2.2.

### 5.1.2. Rank of shaped arrays

#### $\mathfrak{L}$ -Rank

Following the definition of finite-rank time series [1, Chapter 5] and finite-rank 2D arrays [40] (see also [39]), let us generalize the notion of rank to the shaped and circular shaped cases. Note that generally the theory of infinite arrays of finite rank is closely related to multidimensional arrays satisfying LRRs.

**Definition 11.** *The  $\mathfrak{L}$ -trajectory space  $S^{\mathfrak{L}}$  of a shaped array  $S \in \mathbb{R}^{\mathfrak{N}}$  is defined as*

$$S^{\mathfrak{L}}(S) = \text{span}\{(S)_{\mathfrak{L} \oplus \{\kappa\}}\}_{\kappa | \mathfrak{L} \oplus \{\kappa\} \subset \mathfrak{N}}.$$

The introduced trajectory space corresponds to the column space of the trajectory matrix  $\mathbf{S} = \mathcal{T}_{\text{ShSSA}}(S)$ . In view of the isomorphism  $\mathcal{J}_{\mathfrak{L}}$ ,  $\{P_1, \dots, P_r\}$  is a basis of  $S^{\mathfrak{L}}(S)$  if and only if  $\{P_1, \dots, P_r\}$ , where  $P_k = \mathcal{J}_{\mathfrak{L}}(P_k)$ , is a basis of the column space of  $\mathbf{S}$ . Let  $\{Q_1, \dots, Q_r\}$  be a basis of the row space of  $\mathbf{S}$ , that is, of the row trajectory space. Then  $\{Q_1, \dots, Q_r\}$ , where  $Q_k = \mathcal{J}_{\mathfrak{R}}^{-1}(Q_k)$ , forms the basis of a space, which can be denoted as  $S^{\mathfrak{R}}(S)$ . We will use the terms *column/row trajectory spaces* and *column/row shaped trajectory spaces*, depending on the context.

**Definition 12.** *The  $\mathfrak{L}$ -rank of a shaped array  $S$  is defined as the dimension of its  $\mathfrak{L}$ -trajectory subspace:*

$$\text{rank}_{\mathfrak{L}} S = \dim S^{\mathfrak{L}}(S) = \text{rank } S.$$

#### Infinite arrays of finite rank

To describe the general form of arrays of finite rank, we consider infinite arrays (and their trajectory spaces) in both sides for both dimensions (i.e.,  $\mathfrak{N} = \mathbb{Z} \times \mathbb{Z}$ ). We will consider the planar case with  $T_x = T_y = \infty$ .

**Definition 13.** *Infinite (in both directions) array  $S_{\infty}$  is called the array of finite shaped rank if  $r = \max_{\mathfrak{L}} \text{rank}_{\mathfrak{L}} S_{\infty}$  (here the maximum is taken over  $\mathfrak{L} \subset \{1, \dots, \infty\} \times \{1, \dots, \infty\}$ ) is finite. In this case, we will write  $\text{rank } S_{\infty} = r$ .*

**Remark 41.** *The shaped rank for an infinite array is equal to its rectangular rank (when the maximum is taken over rectangular shapes), since a sequence of shapes that contain increasing rectangular shapes and are contained in the rectangular shapes can be easily constructed. Therefore, we will talk about arrays of finite rank omitting “shaped”.*

**Remark 42.** *If  $S_{\infty}$  is an infinite array of finite rank  $r$ ,  $S = (S_{\infty})_{\mathfrak{N}}$  is a sufficiently large finite subarray of  $S_{\infty}$  and  $\mathfrak{L}$  is a sufficiently large shape, then*

$$\text{rank}_{\mathfrak{L}} S = \text{rank}_{\mathfrak{L}} S_{\infty} = \text{rank } S_{\infty} = r. \quad (5.4)$$

Let  $\mathfrak{K} = \{\kappa \mid \mathfrak{L} \oplus \{\kappa\} \subset \mathfrak{N}\}$ . A sufficient condition for (5.4) is that both  $\mathfrak{L}$  and  $\mathfrak{K}$  shapes contain at least one  $r \times r$  square. More formally, it is sufficient that there exists a 2D index  $\alpha = (l, n)$  such that  $\{1, \dots, r\} \times \{1, \dots, r\} \oplus \{\alpha\} \subset \mathfrak{L}$ ; the same should be valid for  $\mathfrak{K}$ .

**Proposition 23.** Let  $T_x = T_y = \infty$ . Then an infinite array  $S_\infty$  of finite rank has the form

$$(S_\infty)_{m,n} = \sum_{k=1}^s P_k(m, n) \mu_k^m v_k^n, \quad (5.5)$$

where  $\mu_k, v_k \in \mathbb{C}$ , the pairs  $(\mu_k, v_k)$  are different, and  $P_k(m, n)$  are complex polynomials of  $m$  and  $n$ . This representation is unique up to the order of summation.

As was mentioned in Section 1.5, this important fact is well known starting from [47, §2.20]. Note that the rank of the array  $S_\infty$  given in (5.5) is not determined by the degrees of the polynomials  $P_k$  only and is hard to compute (see [40]).

An important special case is  $(S_\infty)_{ln} = \sum_{k=1}^r A_k \mu_k^l v_k^n$ ; it is widely used in signal processing. In this case,  $\text{rank } S_\infty$  is equal to the number of different  $(\mu_k, v_k)$ -pairs.

### 5.1.3. Algorithm

#### ALGORITHM 5.1: Shaped 2D-SSA (planar): decomposition

*Input:* Image  $X$  of size  $N_x \times N_y$  and mask of the same size consisting of TRUE and FALSE; window of size  $L_x \times L_y$  and mask of the same size consisting of TRUE and FALSE; if the objects contain ‘NA’, this means that the FALSE values are added to the masks at the places of ‘NA’.

*Output:* Shaped image  $\mathfrak{N}$ , which has been decomposed, consisting of  $N$  points covered by the shaped window  $\mathfrak{L}$  consisting of  $L$  points; shape  $\mathfrak{K}$  consisting of the  $K$  possible positions of the window. Decomposition of the trajectory matrix of size  $L \times K$ :  $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d$ , where  $\mathbf{X}_i = \sqrt{\lambda_i} U_i V_i^T$ .

- 1: Construct the shapes  $\mathfrak{L}$ ,  $\mathfrak{K}$ ,  $\mathfrak{N}$  and the trajectory matrix  $\mathbf{X} = \mathcal{T}_{\text{ShSSA}}(X|_{\mathfrak{N}})$ , where  $\mathcal{T}_{\text{ShSSA}}$  is defined by (5.3).
- 2: Compute the SVD  $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d$ ,  $\mathbf{X}_i = \sqrt{\lambda_i} U_i V_i^T$ .

The circular version of 2D-SSA can be considered in two versions, the cylindric and toroidal ones. Let us comment on the case when the image is given on a cylinder with the main axis parallel to the axis ‘y’. This case corresponds to  $T_x < \infty$  and  $T_y = \infty$ . It is convenient to take  $T_x = N_x$ , which means that the image is located on the circular segment of a cylinder. The only difference between the algorithm of the circular 2D-SSA and the rectangular one, see Algorithm 5.1, is in the number  $K$  of the windows, which cover the image, and therefore in the form of the trajectory matrix  $\mathbf{X}$ . For example, in the cylindric topology, the window, which starts on one edge of the image, can continue to the opposite edge.

Note that the presentation of  $U_i$  or  $V_i$  in plots is performed in the form of shapes  $\Psi_i = \mathcal{J}_{\mathfrak{L}}^{-1}(U_i)$  and  $\Phi_i = \mathcal{J}_{\mathfrak{K}}^{-1}(V_i)$ , which are called eigenshapes and factor shapes respectively. Here  $\mathcal{J}$  is the shaping operator, see (5.1). In the cylindric version, if the window is bounded, then eigenshapes are also bounded, while factors shapes are given on a cylinder.

Reconstruction stage is standard.

#### ALGORITHM 5.2: Shaped 2D-SSA: reconstruction

*Input:* Decomposition  $\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_d$ , where  $\mathbf{X}_i = \sigma_i U_i V_i^T$  and  $\|U_i\| = \|V_i\| = 1$ , grouping  $\{1, \dots, d\} = \bigsqcup_{j=1}^m I_j$ ,  $\mathfrak{N}$ .

*Output:* Decomposition of the object on identifiable components:  $\mathbf{X}|_{\mathfrak{N}} = \mathbf{X}_1 + \dots + \mathbf{X}_m$ .

- 1: Construct the grouped matrix decomposition  $\mathbf{X} = \mathbf{X}_{I_1} + \dots + \mathbf{X}_{I_m}$ , where  $\mathbf{X}_I = \sum_{i \in I} \mathbf{X}_i$ .
- 2:  $\mathbf{X}|_{\mathfrak{N}} = \mathbf{X}_1 + \dots + \mathbf{X}_m$ , where  $\mathbf{X}_i = \mathcal{T}_{\text{ShSSA}}^{-1} \circ \Pi_{\mathfrak{N}-\text{HbH}}(\mathbf{X}_{I_i})$ .

## 5.2. Particular cases of Shaped SSA

In this section, we show that all the considered variants of SSA are in fact special cases of Shaped SSA, for carefully chosen array and mask.

### 5.2.1. Shaped 1D-SSA

Basic SSA corresponds to the shapes

$$\mathfrak{N} = \{1, \dots, N\} \times \{1\}, \quad \mathfrak{L} = \{1, \dots, L\} \times \{1\}.$$

The shaped arrays  $\mathbf{X} \in \mathbb{R}^{\mathfrak{N}}$ , in this case, are time series, and the set  $\mathbf{H}^{\mathfrak{L} \times \mathfrak{K}}$  is the set of ordinary Hankel matrices.

1D-SSA for time series with missing values is obtained by removing the indices of missing elements from the shape  $\mathfrak{N}$ . This is the case when Shaped SSA is particularly useful when the time series contains gaps as in this case the versions of 1D-SSA considered above are not directly applicable. Let us consider it in detail.

The  $L$ -trajectory matrix  $\tilde{\mathbf{X}} = \mathcal{T}_{\text{shSSA}}(\mathbf{X})$  is constructed so that its columns are the complete  $L$ -lagged vectors. Any incomplete lagged vectors containing missing values are not included in  $\mathcal{T}_{\text{shSSA}}(\mathbf{X})$ .

Denote the set of series elements, which are presented in the trajectory matrix  $\tilde{\mathbf{X}}$ , as  $\mathfrak{N}$ ; that is,  $\mathfrak{N}$  is the set of non-missed elements of  $\mathbf{X}$ , which are covered by windows of length  $L$ . The operator  $\mathcal{T}_{\text{shSSA}}$  makes a one-to-one correspondence between a restriction of the series to  $\mathfrak{N}$  and the set of trajectory matrices, if the location of the missing data is fixed.

The SSA decomposition is performed by any technique (excluding Toeplitz SSA) described in

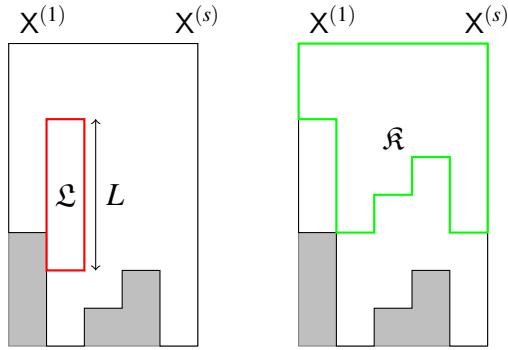


Figure 5.1: MSSA: 2D packing.

Sections 1.8, 2.1–2.3.1 including nested Iterative O-SSA and DerivSSA. All these SSA decompositions, except for Toeplitz SSA, are eligible tools since the construction of the trajectory matrix and a tool for SSA decomposition do not affect one another. Thus, we can naturally define shaped Basic SSA, shaped SSA with projection, and so on.

For Toeplitz SSA, the decomposition is performed in a very specific way, which is not directly based on the trajectory matrix; thereby the shaped version of Toeplitz SSA does not make much sense and hence it is not implemented in the current version of RSSA.

## Separability

The definition and conditions of separability for Shaped SSA are the same as for underlying modifications of SSA (see e.g. Section 1.8.3). However, the separability accuracy is naturally worse when there are gaps in the series. Moreover, there are extreme cases where ranks of series can be corrupted by gaps and where the conditions of separability can not be satisfied.

### 5.2.2. MSSA

Let us present two ways of representing MSSA, which was described in Chapter 4 as special cases of Shaped SSA.

#### MSSA: 2D packing

Consider a multivariate time series, that is, a collection  $\{\mathbf{X}^{(p)} = (x_j^{(p)})_{j=1}^{N_p}, p = 1, \dots, s\}$  of  $s$  time series of length  $N_p$ ,  $p = 1, \dots, s$ . We construct the shaped array  $\mathbf{X}_{\mathfrak{N}}$  such that

$$\mathfrak{N} = \{1, \dots, N_1\} \times \{1\} \cup \{1, \dots, N_s\} \times \{s\} \subset \{1, \dots, \max_p N_p\} \times \{1, \dots, s\}.$$

The window is taken to be  $\mathfrak{L} = \{1, \dots, L\} \times \{1\}$ . The construction of this array is shown in Figure 5.1, also with the window  $\mathfrak{L}$  and the shape  $\mathfrak{K}$ . It is easy to verify that  $\mathbf{X} = \mathcal{T}_{\text{ShSSA}}(\mathbf{X})$  coincides with  $\mathcal{T}_{\text{MSSA}}(\mathbf{X})$  defined in (4.1). Therefore, the rows of  $\mathbf{X}$  are vectorizations of the  $\mathfrak{K}$ -shaped subarrays (see Figure 5.1).

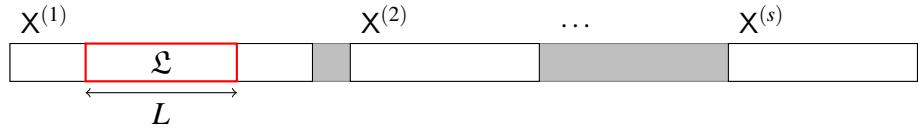


Figure 5.2: MSSA: 1D packing.

### MSSA: 1D packing

Now we consider an alternative packing of MSSA. From the same set of series we construct a  $N' \times 1$  (or  $1 \times N'$ ) array  $\mathbf{X}$ , where  $N' = N + (s - 1)$  (recall that  $N = \sum_{p=1}^s N_p$ ). The array consists of the time series plus “separators” between them that are not included in the array shape. The window is taken to be  $L \times 1$  (or  $1 \times L$ ), depending on the arrangement chosen. In Figure 5.2 we show the horizontal variant of packing.

### Mosaic Hankel matrices

A mosaic Hankel matrix [175] is a block matrix with Hankel blocks. It can be considered the most general one-dimensional (i.e., with one-dimensional displacement) generalization of Hankel matrices.

Let  $L_1, \dots, L_s$  and  $K_1, \dots, K_t$  be integer vectors, and  $X^{(i,j)} \in \mathbb{R}^{L_i+K_j-1}$  be time series. Then the mosaic Hankel matrix is constructed as follows:

$$\begin{pmatrix} H_{L_1, K_1}(X^{(1,1)}) & \dots & H_{L_1, K_t}(X^{(1,t)}) \\ \vdots & \dots & \vdots \\ H_{L_s, K_1}(X^{(s,1)}) & \dots & H_{L_s, K_t}(X^{(s,t)}) \end{pmatrix}.$$

Note that the sizes of the blocks may be different. The only requirement is that they should match as a “mosaic”. The case of mosaic Hankel matrices corresponds to several collections of multidimensional time series [138].

It is easy to construct mosaic Hankel matrices, based on the 2D embedding of MSSA. A  $j$ th block column is a transposed matrix  $\mathcal{T}_{\text{MSSA}}$  for the collection of time series  $(X^{(1,j)}, \dots, X^{(s,j)})$  and window length  $L = K_j$ . Therefore, the mosaic Hankel matrix can be constructed by stacking shapes (with separators) from Figure 5.1 and replacing  $\mathfrak{L}$  with  $\mathfrak{K}$  due to transposition. The resulting construction of the shaped array is shown in Figure 5.3.

### Subspace-based MSSA extensions

In view of the common structure of all SSA-family algorithms (see Section 1.2), many SSA-related techniques can be naturally extended from 1D objects (i.e., series) to other objects, and particularly to the systems of series.

In Section 4.3.1, we have considered methods of MSSA forecasting. Let us now describe some extensions. Since the algorithms are either exactly or almost identical to the 1D case, we are

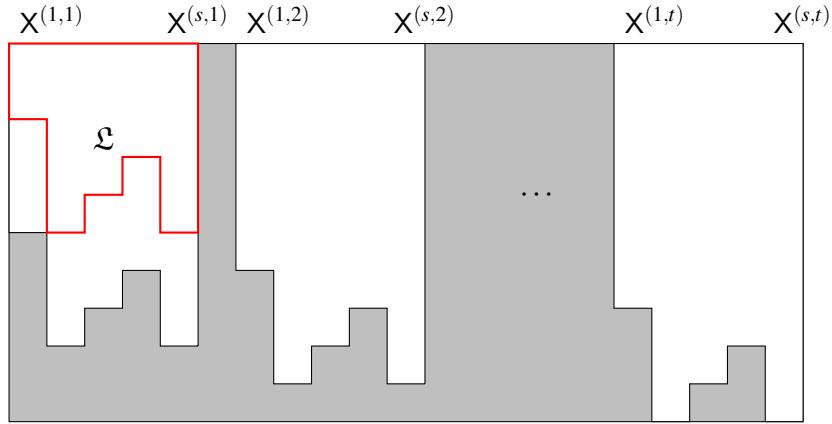


Figure 5.3: Shaped construction for mosaic Hankel matrices.

not writing them out. In particular, all methods and algorithms that are based on the use of the column subspaces are the same in MSSA and 1D-SSA. For example, parameter estimation based on the column subspace can be performed using the same function `parestimate` in the `RSSA` package.

The shaped version of MSSA is almost the same as Shaped 1D-SSA (Section 5.2.1) except for the following difference. If `NA` are placed at the ends of the time series, then the corresponding series is considered a series of smaller length. If there are no other missing data, then the resultant series of smaller length does not have any missing values and is decomposed by a non-shaped version of MSSA. It is important to mention that if there are `NA` at the right end of a series, then forecasts start from the last not-`NA` values.

Formally, forecasting can be applied to shaped SSA objects. However, it is generally recommended to fill gaps first (see Section 3.1) and then forecast the series.

Iterative gap-filling and low-rank approximation by Cadzow iterations (Section 1.9.1) are implemented in a general manner and therefore can be applied to systems of time series with gaps in the same manner as in the 1D case.

Subspace-based gap filling has more specificity but is based on the same approach as for the one-dimensional case.

### 5.2.3. 2D-SSA

In this section, we consider an extension of the 1D-SSA algorithm for the decomposition of two-dimensional data. This extension has the name *2D singular spectrum analysis* (or *2D-SSA* for short). For 2D-SSA, the data object  $\mathbf{X}$  is a *two-dimensional data array* of size  $N_x \times N_y$  (or simply an  $N_x \times N_y$  real-valued matrix), represented as  $\mathbf{X} = \mathbf{X}_{N_x, N_y} = (x_{ij})_{i,j=1}^{N_x, N_y}$ . A typical example of a 2D-array is a digital 2D monochrome image.

The 2D-SSA algorithm corresponds to Shaped 2D-SSA with

$$\begin{aligned}\mathfrak{N} &= \{1, \dots, N_x\} \times \{1, \dots, N_y\}, \\ \mathfrak{L} &= \{1, \dots, L_x\} \times \{1, \dots, L_y\}.\end{aligned}$$

The shaped arrays  $\mathbf{X} \in \mathbb{R}^{\mathfrak{N}}$  are rectangular images, the windows  $\mathfrak{L}$  are rectangular, and the set  $\mathcal{H}^{\mathfrak{L} \times \mathfrak{K}}$  is the set of *Hankel-block-Hankel* matrices.

2D-SSA was proposed as an extension of 1D-SSA in [66], and was further developed in [40, 176]. However, related decompositions were developed independently in texture analysis [177, 178], seismic data processing [179, 75], and parameter estimation methods for sums of two-dimensional complex exponentials, see e.g. [180].

Until recently, the major drawback of the methods based on 2D-SSA decomposition was its computational complexity. The RSSA package contains an efficient implementation of both the general Shaped SSA and the 2D-SSA decomposition and reconstruction, which to a great extent overcomes this deficiency.

## Method

For a matrix  $\mathbf{A} \in \mathbb{R}^{M \times N}$  (or  $\mathbb{C}^{M \times N}$ ), we denote by  $\text{vec}(\mathbf{A}) \in \mathbb{R}^{MN}$  (or  $\mathbb{C}^{MN}$ ) its column-major vectorization. For a vector  $\mathbf{A} \in \mathbb{R}^{MN}$  (or  $\mathbb{C}^{MN}$ ), we define its *M devectorization*, which we named ‘shaping’ in the general case, as the matrix  $\text{vec}_M^{-1}(\mathbf{A}) = \mathbf{B} \in \mathbb{R}^{M \times N}$  (or  $\mathbb{C}^{M \times N}$ ) that satisfies  $\text{vec}(\mathbf{B}) = \mathbf{A}$ .

**The embedding operator** The generic scheme of the SSA algorithm is described in Section 1.2. Hence, to formally present 2D-SSA, we only need to define the embedding operator  $\mathcal{T}_{\text{2D-SSA}}(\mathbf{X}) = \mathbf{X}$ .

The parameters of the method are the two-dimensional *window sizes*  $(L_x, L_y)$ , with restrictions  $1 \leq L_x \leq N_x$ ,  $1 \leq L_y \leq N_y$  and  $1 < L_x L_y < N_x N_y$ . For convenience, we also denote  $K_x = N_x - L_x + 1$ ,  $K_y = N_y - L_y + 1$ . As in the general scheme of the algorithms, we define  $L = L_x L_y$  (the number of rows of  $\mathbf{X}$ ) and  $K = K_x K_y$  (the number of columns of  $\mathbf{X}$ ).

Consider all possible  $L_x \times L_y$  submatrices of  $\mathbf{X}$  (2D sliding windows). For  $k = 1, \dots, K_x$  and  $l = 1, \dots, K_y$ , we define by  $\mathbf{X}_{k,l}^{(L_x,L_y)} = (x_{ij})_{i=k, j=l}^{L_x+k-1, L_y+l-1}$  the submatrix of size  $L_x \times L_y$  shown in Figure 5.4. Note that the  $x$  axis is oriented to the bottom, and the  $y$  axis is oriented to the right; the origin is in the upper left corner. We use this orientation as it is consistent with the standard mathematical indexing of matrices.

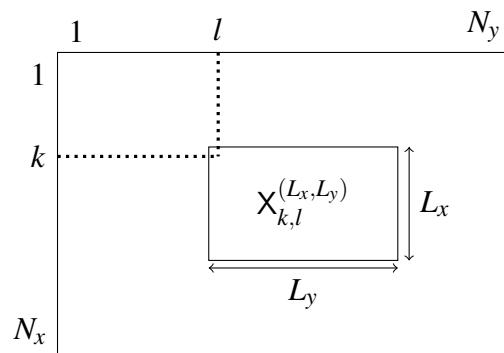


Figure 5.4: Moving 2D windows.

Then the trajectory matrix is defined as

$$\mathcal{T}_{\text{2D-SSA}}(\mathbf{X}) = \mathbf{X} = [X_1 : \dots : X_{K_x K_y}], \quad (5.6)$$

where the columns  $X_j$  are vectorizations of the  $L_x \times L_y$  submatrices:

$$X_{k+(l-1)K_x} = \text{vec}(\mathbf{X}_{k,l}^{(L_x, L_y)}).$$

**Hankel-block-Hankel structure** The trajectory matrix (5.6) has the following structure [40]:

$$\mathbf{X} = \mathcal{T}_{\text{2D-SSA}}(\mathbf{X}) = \begin{pmatrix} \mathbf{H}_1 & \mathbf{H}_2 & \mathbf{H}_3 & \dots & \mathbf{H}_{K_y} \\ \mathbf{H}_2 & \mathbf{H}_3 & \mathbf{H}_4 & \dots & \mathbf{H}_{K_y+1} \\ \mathbf{H}_3 & \mathbf{H}_4 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \mathbf{H}_{L_y} & \mathbf{H}_{L_y+1} & \dots & \dots & \mathbf{H}_{N_y} \end{pmatrix}, \quad (5.7)$$

where each  $\mathbf{H}_j$  is an  $L_x \times K_x$  Hankel matrix constructed from  $\mathbf{X}_{:,j}$  (the  $j$ th column of the 2D array  $\mathbf{X}$ ). More precisely,  $\mathbf{H}_j = \mathcal{T}_{\text{1D-SSA}}(\mathbf{X}_{:,j})$ , where  $\mathcal{T}_{\text{1D-SSA}}$  is defined in (1.1). The matrix (5.6) is called *Hankel-block-Hankel* (shortened to *HbH*), since it is block-Hankel with Hankel blocks. Thus,  $\mathcal{M}_{L,K}^{(\text{H})}$  in 2D-SSA is the set of HbH matrices.

Projection  $\Pi_{\text{HbH}}$  on the space of HbH matrices can be performed using the general form described in Section 1.2.2. In this particular case, the projection consists of hankelization of each matrix block at the places of each  $\mathbf{H}_j$  and then averaging of the blocks at the places of  $\mathbf{H}_j$  with the same index  $j$ .

**Trajectory space** From (5.6), we observe that the trajectory space is the linear space spanned by the  $L_x \times L_y$  submatrices of  $\mathbf{X}$ . Therefore, the eigenvectors  $U_i$  can also be viewed as vectorized  $L_x \times L_y$  arrays. Their devectorizations are denoted by  $\Psi_i = \text{vec}_{L_x}^{-1}(U_i)$ . Similarly, the rows of  $\mathbf{X}$  are vectorizations of the  $(K_x, K_y)$  submatrices

$$\mathbf{X} = [X^1 : \dots : X^{L_x L_y}]^T, \quad X^{k+(l-1)L_x} = \text{vec}(\mathbf{X}_{k,l}^{(K_x, K_y)}), \quad (5.8)$$

where  $X^j$  is the  $j$ th row of the matrix  $\mathbf{X}$ . The factor vectors  $V_i$  can also be viewed as  $K_x \times K_y$  arrays. Their devectorizations are denoted by  $\Phi_i = \text{vec}_{K_x}^{-1}(V_i)$ .

## Comments

1. The algorithm of 2D-SSA coincides with the algorithm of MSSA for time series of the same length when  $L_x = 1$  or  $L_y = 1$ .
2. The arrays  $\mathbf{X}$  of finite rank in 2D-SSA (i.e., the arrays such that  $\mathcal{T}_{\text{2D-SSA}}(\mathbf{X})$  is rank-deficient

and has a fixed rank) are sums of products of polynomials, exponentials and cosines, similar to the one-dimensional case. More details can be found in [40].

3. The generic scheme of SSA described in Section 1.2 includes a decomposition of the trajectory matrix into a sum of rank-one matrices. If this decomposition is performed using the SVD, then we call the method Basic 2D-SSA or simply 2D-SSA. Other versions, such as nested Iterative Oblique 2D-SSA or 2D-SSA with projection, can be also considered.

## Automation of 2D-SSA

We consider a field of size  $N_x \times N_y$ :  $\mathbf{X} = (x_{i,j})_{i,j=1}^{N_x, N_y}$ ,  $x_{i,j} \in \mathbb{R}$ .

For 2D-SSA, we generalize only the low-frequency method described by Algorithm 2.13 for 1D-SSA in Section 2.5.1 [129].

**Low-frequency method for pattern identification** Define a two-dimensional periodogram for the field  $\mathbf{Y} = (y_{n,m})_{n,m=1}^{M_x, M_y}$  as

$$\Pi_{\mathbf{Y}}^{M_x M_y} \left( \frac{k}{M_x}, \frac{l}{M_y} \right) = M_x M_y \|G_{kl}\|,$$

where  $1 \leq k \leq M_x$ ,  $1 \leq l \leq M_y$ ,  $G_{kl}$  are complex numbers, which are the coefficient of the two-dimensional Fourier expansion of the field  $\mathbf{Y}$ :

$$y_{n,m} = \sum_{k=1}^{M_x} \sum_{l=1}^{M_y} G_{kl} e^{2\pi i (nl/M_x + mk/M_y)},$$

$$G_{kl} = \frac{1}{M_x M_y} \sum_{n=q}^{M_x} \sum_{m=2}^{M_y} y_{n,m} e^{-2\pi i (nl/M_x + mk/M_y)}.$$

As well as in the 1D-SSA case, we define for the field  $\mathbf{Y}$  and  $-0.5 \leq \omega_1, \omega_2 \leq 0.5$  measure

$$T(\mathbf{Y}; \omega_1; \omega_2) = \sum_{k:0 \leq k/M_x \leq \omega_1} \sum_{l:0 \leq l/M_y \leq \omega_2} I_{\mathbf{Y}}^{M_x M_y}(k/M_x, l/M_y), \quad (5.9)$$

where  $I_{\mathbf{Y}}^{M_x M_y}(k/M_x, l/M_y) = \Pi_{\mathbf{Y}}^{M_x M_y} \left( \frac{k}{M_x}, \frac{l}{M_y} \right) / \|\mathbf{Y}\|^2$ .

Since  $\|\mathbf{Y}\|^2 = \sum_{k=1}^{\lfloor M_x/2 \rfloor} \sum_{l=1}^{\lfloor M_y/2 \rfloor} \Pi_{\mathbf{Y}}^{M_x M_y} \left( \frac{k}{M_x}, \frac{l}{M_y} \right)$ , then the measure  $T(\mathbf{Y}; \omega_1; \omega_2)$  can be considered as the contribution of frequencies contained in the frequency rectangle  $\{[0, \omega_1] \times [0, \omega_2]\}$ .

Algorithm 5.3 implements the low-frequency method for 2D-SSA.

ALGORITHM 5.3: 2D-SSA. Low frequency method for trend

*Input:* The following data and parameters are input.

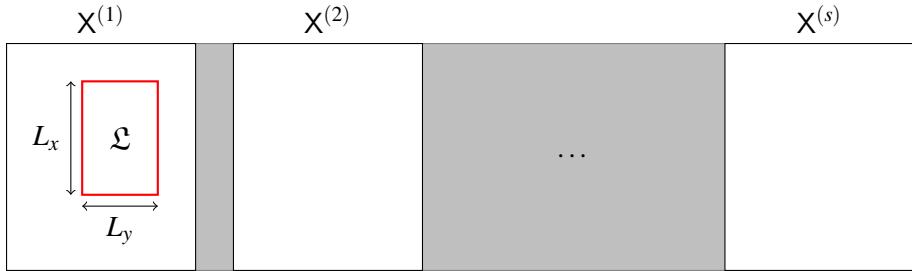


Figure 5.5: Shaped construction for M-2D-SSA.

1. **Data:** index group  $I \in \{1, \dots, d\}$ ; values  $-0.5 \leq \omega_1, \omega_2 \leq 0.5$ ; fields  $Y_i, i \in I$ , which can be elementary reconstructed arrays or devectorized left/right singular vectors.
2. **Parameters:** threshold  $0 \leq T_0 \leq 1$ .

*Output:* A group of indices  $J \subset I$  of components related to the trend.

- 1: For each field  $Y_i, i \in I$ , calculate the value  $T(Y_i; \omega_1; \omega_2)$  using (5.9).
  - 2: The set  $J$  is the group of indices  $i \in I$  such that  $T(Y_i; \omega_1; \omega_2) \geq T_0$ .
- 

#### 5.2.4. M-2D-SSA

Suppose that we have  $s$  arrays  $X^{(1)}, \dots, X^{(s)} \in \mathbb{R}^{N_x \times N_y}$  and we would like to consider a variant of 2D-SSA where the trajectory matrix is stacked from  $s$  trajectory matrices (as in MSSA):

$$\mathcal{T}_{M-2D-SSA}(X^{(1)}, \dots, X^{(s)}) = [\mathcal{T}_{2D-SSA}(X^{(1)}) : \dots : \mathcal{T}_{2D-SSA}(X^{(s)})]. \quad (5.10)$$

In this case, the 2D-SSA-like decomposition will have the common basis of eigenvectors, as in MSSA. The trajectory matrices of the form (5.10) are used in 2D-SSA-based for comparison of images [176]. These matrices are also used in recent methods of parallel magnetic resonance imaging [181].

The packing for the M-2D-SSA can be constructed in a similar way to the case of mosaic Hankel matrices. An array  $X'$  of size  $N_x \times (sN_y + s - 1)$  is constructed from the arrays with one-element separators. The resulting array is shown in Figure 5.5. In general, a similar construction can handle arrays of different sizes, shapes, and shaped windows. Also, note that in the extended array the original arrays (both for Figures 5.3 and 5.5) may be arranged arbitrarily (for example in a table-like planar arrangement). The only requirement is that they should be separated.

#### 5.2.5. Comments on nD extensions

We mentioned at the beginning of this chapter that the nD case is very similar to the 2D case. Moreover, many approaches presented in Chapters 2 and 3 for the 1D case can be applied in the nD case. Let us describe the commonalities and the specificity of the nD case with  $n > 2$ .

- The nD objects can be of non-rectangular shapes; also, non-rectangular windows can be used. That is, the shaped nD-SSA is implemented. Moreover, all other SSA versions can be considered as particular cases of Shaped nD-SSA.
- The method nD-ESPRIT of frequency and damped factor estimation, which is a direct extension of the 2D-ESPRIT and Shaped ESPRIT, is implemented for objects of any dimension and shape.
- For the nD case, Iterative O-SSA and Filter-adjusted O-SSA, which are described in Chapter 2, and also Iterative gap-filling and Cadzow iterations, which are described in Chapter 3, are implemented similarly.
- Visual inspection of nD objects is an obviously difficult problem. Therefore, visual control of results in the nD case is complicated.

### 5.3. Example of Shaped SSA

Let us consider the example of Shaped 2D-SSA for removing a texture in a shaped image.

As an example, we use the image of Mars by Pierre Thierry, from the tutorial of the free IRIS software. The image is of size  $258 \times 275$ , 8-bit grayscale, values from 0 to 255. The reconstruction results for decomposing the rectangular image are shown in Figure 5.6.

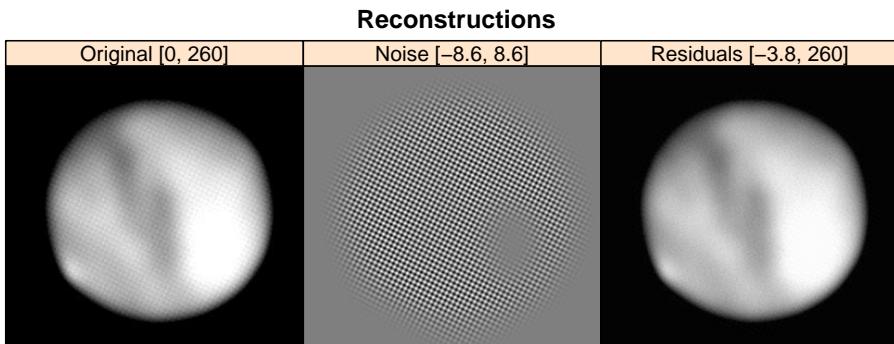


Figure 5.6: ‘Mars’: Decomposition by 2D-SSA,  $(L_x, L_y) = (25, 25)$ .

One can see that the image of Mars is circular-shaped and there is a small area of light-striking. Therefore, there is a effect of blurring sharp boundaries. Let us apply Shaped 2D SSA. To do it, we should set masks.

The array shape can be specified in two different ways:

- by passing the `NA` values in the input array (these elements are excluded), or,
- by specifying a logical mask — a logical  $N_x \times N_y$  array (the indicator of  $\mathfrak{N}$ ).

If both shape specifications are present, their intersection is considered.

In Figure 5.7 one can see both types of masks and the combined mask. The reconstruction

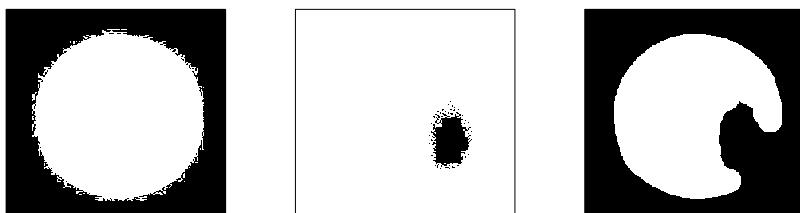


Figure 5.7: Mars masks specification. Left: specified by `NA`, center: logical mask, right: resulting mask. White squares — `TRUE`, black squares — `FALSE`.

results together with the original image are shown in Figure 5.8. The circle window of radius 15 was used. In Figure 5.8 we can see that the decomposition elements are only inside the resulting

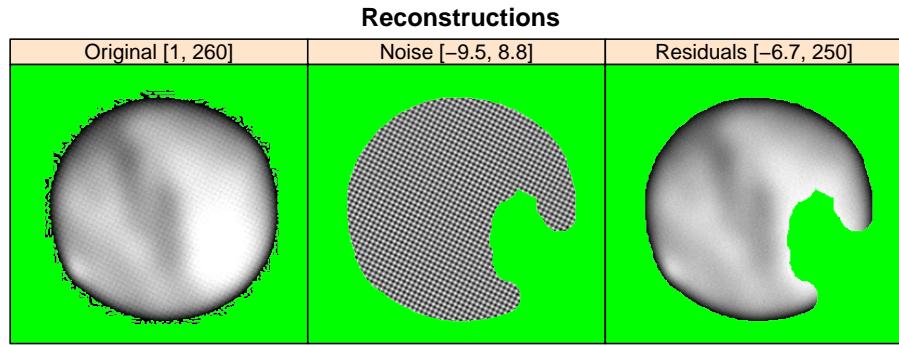


Figure 5.8: ‘Mars’: Decomposition by Shaped SSA, circular window of radius 15.

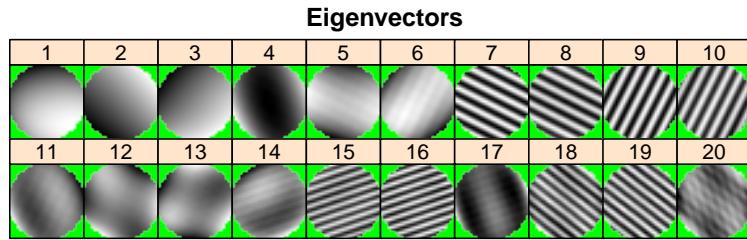


Figure 5.9: Mars: Eigenarrays, Shaped SSA.

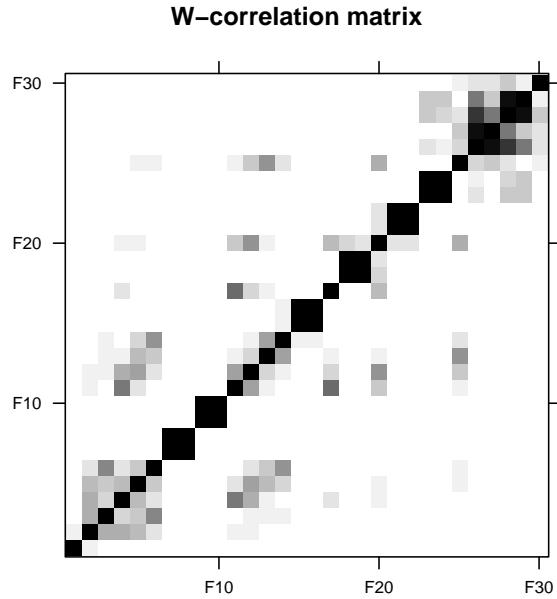


Figure 5.10: Mars: **w** correlations, Shaped SSA.

mask, however the original array is drawn for all available elements (except the NA values). The grouping for this decomposition was made based on the following information:

- eigenarrays (see Figure 5.9), and,
- the matrix of **w** correlations (see Figure 5.10).

The quality of the texture extraction and therefore of the image recovery by shaped SSA

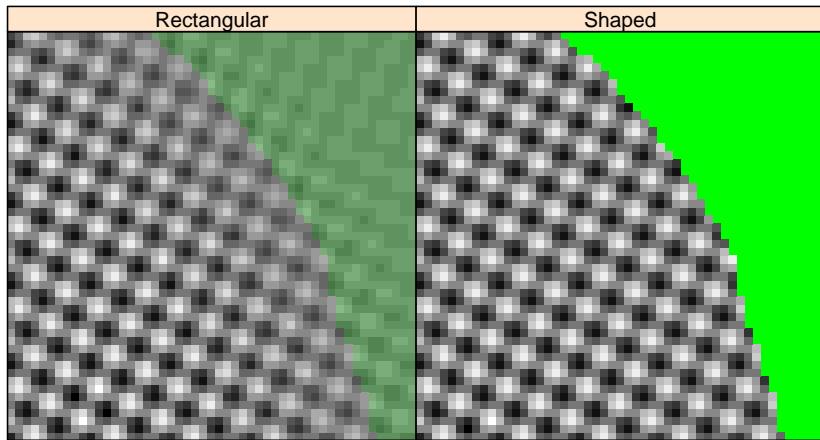


Figure 5.11: Mars: Comparison of texture reconstructions by 2D-SSA and Shaped 2D-SSA.

(Figure 5.8) is considerably better than that performed by 2D-SSA; the improvement of reconstruction accuracy is explained by an edge effect that is caused by a sharp drop of intensity near the boundary of Mars. In Figure 5.11, we compare magnified reconstructed images for 2D-SSA and Shaped 2D-SSA. In the left subfigure, a green shadow is shown for the background area in order to indicate the Mars boundary. In the right subfigure, light green color corresponds to NA.

## Chapter 6

# Package Rssa

There are many implementations of SSA. They differ by potential application areas, implemented methods, interactive or non-interactive form, free or commercial use, the computer system (Windows, Unix, Mac), level of reliability and support. The most known supported software packages implementing SSA are the following:

1. <http://gistatgroup.com>: general-purpose interactive ‘Caterpillar’-SSA software (Windows) following the methodology described in [1, 29];
2. <http://www.atmos.ucla.edu/tcd/ssa>: oriented mainly on climatic applications SSA-MTM Toolkit for spectral analysis [168] (Unix) and its commercial extension kSpectra Toolkit (Mac), interactive;
3. The commercial statistical software, SAS, has an econometric extension called SAS/ETS®, which includes SSA in its rather basic form; this version of SSA is based on the methodology of [1].
4. <http://cran.r-project.org/web/packages/Rssa>: R package RSSA [70], an implementation of the main SSA procedures for major platforms, extensively developed.

We consider the RSSA package as an efficient implementation of a large variety of the main SSA algorithms. This package also contains many visual tools which are useful for making the proper choice of SSA parameters and examination of results.

RSSA is the only SSA package available from CRAN and we believe it is the fastest implementation of SSA. Another important feature of the package is its very close relation to the SSA methodology thoroughly described in [1, 29, 30]. As a result of this, the use of the package is well theoretically and methodologically supported.

The contribution of the author to the creation of this package is the elaboration of the general structure, creating the function interfaces, writing the help to the package, working out the algorithms of the methods for their implementation together with a description of them in [8, 30, 12] devoted to RSSA.

**Tools for visual control and choice of parameters** SSA needs tools to help choose the parameters of the methods and control the results. To a great extent, SSA is an exploratory technique and hence visual tools are vital and they are extensively used in RSSA. For example, in order to help to choose the groups in (1.4), RSSA allows plotting measures of separability of series components in the obtained decompositions.

The tools for accuracy control are divided into two groups. First, the stability of results to parameter changes can be checked. Second, the bootstrap procedure could be used when the model (not necessarily parametric) of either the series or other object is built based on signal reconstruction and then the accuracy of this model is assessed by simulation according to the estimated model. Shortly, RSSA allows the user to enjoy a large variety of graphical tools and bootstrap procedures.

## 6.1. Brief introduction to Rssa

Let us enumerate the general principles for SSA implementation in RSSA.

- The interface of functions for different types of objects is unified.
- The effective and fast implementation of the SVD (in C) based on similar mathematic approaches is used.
- Many approaches that were first developed for 1D-SSA are implemented for multidimensional objects.

The RSSA package implements a lot of methods and tools developed in the framework of SSA. The main function is `ssa`, which initializes an *ssa object* and by default performs the decomposition by different methods. Together with `reconstruct`, they implement the SSA method. For nested versions, `iossa` and `fossa` serve for refined decompositions.

An `ssa-object` `s` contains, among others, elements of the decomposition (1.2), which can be accessed as `s$sigma`, `s$U` and `s$V`. Features of the decompositions differ for different versions of SSA (see Section 1.3.1). For Basic SSA, `s$sigma` are called singular values; squares of `s$sigma` are called eigenvalues; `s$U` are called eigenvectors. (We keep these names for other versions of SSA as well.) The relative contributions of components to the decomposition can be obtained as `contributions(s)`; see Section 1.3.1, where formulas for their calculation are given and explained.

A variety of functions `plot` help to visualize the results and additional information. The functionality of SSA-related methods is supplemented by the functions `forecast`, `parestimate` and some others.

All essential versions of SSA are implemented in RSSA but not all further actions like forecasting and gap filling are consistent with all implemented versions of SSA. The user can check the `ssa-object`, which is returned by the main function `ssa`, for consistency by the function `ssa.capabilities`. This function returns a list of capabilities with information `TRUE` or `FALSE` respectively.

A general scheme of the investigation by means of RSSA is as follows:

1. perform decomposition by `ssa`;
2. visualize the result by `plot`;

3. if necessary, refine decomposition by `iossa` or `fossa`;
4. again, visualize the result by `plot`;
5. perform grouping based on the obtained visual and numerical information; in particular, choose the group of signal components;
6. then perform one of the following actions: reconstruction of series components by `reconstruct`, forecasting by `forecast`, parameter estimation by `parestimate`;
7. visualize the result by `plot`.

## 6.2. Implementation efficiency

### 6.2.1. Efficiency of the R-package Rssa

The user does not need to know the specifics of the internal implementation of the Rssa functions. However, understanding the general principles of implementation can help to use the package more effectively.

The fast implementation of SSA-related methods, which was suggested in [113], extended in [12] and is used in the Rssa package [70], relies on the following techniques (see [41] for a more thorough discussion).

1. The truncated SVD calculated by the Lanczos methods [173, Ch. 9] is used. In most SSA applications, only several leading SVD components correspond to the signal and therefore are used at Grouping step of the SSA algorithm. Thus, a truncated SVD rather than the full SVD is usually required by SSA.
2. Lanczos methods do not use the explicit representation of the decomposed matrix  $\mathbf{A}$ . They need only the results of the multiplication of  $\mathbf{A}$  and  $\mathbf{A}^T$  on some vectors. In view of the special Hankel-type structure of  $\mathbf{A}$  in the SSA algorithms, multiplication by a vector can be implemented very efficiently with the help of the Fast Fourier Transform (FFT). Fast SVD algorithms are implemented in the R-package `svd` [182] in such a way that their input is the function of a vector which yields the result of fast multiplication of the vector by the trajectory matrix. Therefore, the use of `svd` in Rssa allows a fast and space-efficient implementation of the SSA algorithms.
3. At Reconstruction step, hankelization or quasi-hankelization of a matrix of rank 1, stored as  $\sigma \mathbf{U} \mathbf{V}^T$ , can be written through the convolution operator and therefore can also be effectively implemented; this is also done in Rssa.

In comparison with straightforward implementations, the overall complexity of the computations is dropped from  $O(N^3)$  down to  $O(kN \log N + k^2 N)$  and the memory consumption is reduced

from  $O(N^2)$  to  $O(N)$ , where  $N$  is the number of elements in a shaped object and  $k$  is the number of considered eigentriples, see details in [113] and in [12]. This makes the computations feasible for large data sets and large window sizes. For example, the case of an image of size  $299 \times 299$  and a window size  $100 \times 100$  can be handled rather easily, whereas the conventional algorithms (e.g., the full SVD [173]) are impractical because the matrix that needs to be decomposed has size  $10^4 \times 4 \cdot 10^4$ . Using larger window sizes is often advantageous, since, for example, the separability of signal from noise (in the “signal+noise” scenario) can be significantly improved.

### 6.2.2. Example of calculations in Rssa

Let us demonstrate how fast are the computations in RSSA. For the time series length  $N = 1000000$  and the window length  $L = 500000$ , the reconstruction of a sine wave signal based on two leading components is executed in a few seconds:

```
> library("Rssa")
> N <- 1000000
> signal <- sin((1:N) * 2 * pi / 10)
> ts <- signal + 10*rnorm(1:N)
> system.time(s <- ssa(ts, L = N/2, svd.method = "auto", neig = 2))
      user      system     elapsed
      1.19        0.16      1.34
> system.time(rec <- reconstruct(s, groups = list(sig = 1:2)))
      user      system     elapsed
      0.55        0.13      0.67
> max(abs(signal - rec$sig))
[1] 0.0515102
```

## 6.3. Unified approach for implementation of the SSA scheme

### 6.3.1. Basic analysis

Let us demonstrate that the decomposition and reconstruction are fulfilled in a similar manner. We start with the description of the scheme of application of Basic SSA for reconstructing a component from a one-dimensional time series. In this example, the trend is reconstructed using ET1,4,7. The code for times series with gaps is the same.

```
> F <- co2;
> # F[100:200] <- NA  #the code will be the same if to add missing values
> # Perform Basic or Shaped SSA
> s1 <- ssa(F, L = 72)
> # Plots for choosing the components for reconstruction
> plot(s1, type = "vectors", idx = 1:12)
```

```

> plot(s1, type = "series", groups = 1:6, layout = c(2, 3))
> plot(wcor(s1, groups = 1:20), scales = list(at = seq(1,20,2)))
> # Reconstruction of the trend
> r1 <- reconstruct(s1, groups = list(c(1, 4, 7)))
> # Plot the extracted trend
> plot(r1, add.residuals = FALSE,
+       plot.method = "xyplot", superpose = TRUE)

```

In multivariate and multidimensional extensions of SSA, decomposition and reconstruction are performed similarly. In the next code of MSSA application, for brevity, we omit the code for identifying the trend and seasonal components.

```

> wineFortDry <- wine[, c("Fortified", "Drywhite")]
> # Perform Multivariate SSA
> s.wineFortDry <- ssa(wineFortDry, L = 84, kind = "mssa")
> r.wineFortDry <- reconstruct(s.wineFortDry,
+                               groups = list(Trend = c(1, 6),
+                                             Seasonality = c(2:5, 7:12)))
> # Plot the extracted trend
> plot(r.wineFortDry, add.residuals = FALSE,
+       plot.method = "xyplot",
+       superpose = TRUE, auto.key = list(columns = 3))

```

Below, the code of Shaped 2D-SSA is applied to a rectangular image, where the shape is created by a bit mask. Since the form of the Mars image is close to a circle, the window form is also a circle.

```

> data("Mars", package = "Rssa")
> mask.Mars.0 <- (Mars != 0) #removes black
> mask.Mars.1 <- (Mars != 255)
> Mars[!mask.Mars.0] <- NA
> # Perform Shaped 2D SSA
> s.Mars.shaped <- ssa(Mars, kind = "2d-ssa",
+                         mask = mask.Mars.1, wmask = circle(15))
> # Plots for choosing the components for reconstruction
> plot(s.Mars.shaped, type = "vectors", idx = 1:30,
+       fill.color = "yellow", cuts = 255, layout = c(10, 3),
+       plot.contrib = FALSE)
> plot(wcor(s.Mars.shaped, groups = 1:30),
+       scales = list(at = c(10, 20, 30)))
> plot(s.Mars.shaped)
> r.Mars.shaped.groups <- list(Noise = c(7, 8, 9, 10))
> #reconstruct and plot the pattern

```

```
> r.Mars.shaped <- reconstruct(s.Mars.shaped,
+                               groups = r.Mars.shaped.groups)
> plot(r.Mars.shaped, cuts = 255, layout = c(3, 1),
+       fill.color = "yellow")
```

### 6.3.2. Different modifications of the decomposition step

The approach to the application of different modifications of the decomposition step is also unified.

The modifications related to different assumptions about the time series structure are implemented in the **ssa** function, which returns an **ssa**-object. For example, Toeplitz SSA is chosen by the parameter `kind = "toeplitz-ssa"`. The ProjSSA version is chosen if the additional parameters `column.projector` and `row.projector` are set.

The modifications for nested decompositions such as Iterative-OSSA or DerivSSA are performed after the signal components are detected, to improve separability. This can be done by **iossa** and **fossa** functions applied to **ssa**-object and signal components.

Moreover, this approach works for shaped and multidimensional objects. In particular, the nested decompositions are performed in the general case by the same **iossa** and **fossa** functions.

### 6.3.3. Subspace-based methods

By the use of the **ssa**-object and groups of eigentriples, such actions as gap filling, parameter estimation and forecasting can be performed. This is also implemented in a unified manner. However, a specificity of signal subspaces for different kinds of input objects imposes constraints on the applied subspace-based methods.

## Chapter 7

# Applications to real-life data

In this chapter, we introduce several real-life examples, where SSA helps to obtain results in the corresponding real-life area.

In Section 7.1, SSA is used for the prediction of parameters of Earth orientation. It is shown that the results are generally better than the known ones. Section 7.2 describes the application of SSA for smoothing cumulative distribution functions and estimation of densities.

Sections 7.3–7.5 are devoted to applications of SSA for analysis of gene expressions. In Sections 7.3, we show how SSA can help in the construction of the parametric model of data with the following analysis of the dynamics based on the estimated parameters. Sections 7.4 and 7.5 demonstrate the use of Shaped SSA for noise reduction and feature extraction in 2D and 3D cases correspondingly.

In addition to the results given in Sections 7.3–7.5, we point out the following. In [6], SSA is first applied to remove noise in 2D gene expression data. In the appendix of that paper, the technique of applying 2D-SSA to planar data for pattern extraction is described in detail. As a continuation of this work, in [26] 2D-SSA is applied to noise model analysis. In [27] and [18], SSA is applied for pattern extraction from one-dimensional gene expression profiles; special emphasis is placed on questions of the nature of noise, the biological one or that induced by preprocessing data.

In addition to the papers devoted to gene expression analysis, we also mention the papers [43, 44], where SSA is applied for filtering in 2D digital terrain models.

### 7.1. EOP time series prediction using singular spectrum analysis

Accurate forecasting of Earth orientation parameters (EOP) is important for improving the GPS location accuracy and navigation of Earth satellites. EOP time series include periodic components of complex structure. In this section, a unified approach to choosing parameters of the SSA forecasting algorithm for EOP time series prediction is proposed. EOP time series data published by IERS in Bulletin 14 C04 are used for 365-days prediction. The forecasts performed by the proposed techniques are compared with predictions taken from available public sources.

Earth orientation parameters are a collection of parameters that describe different aspects of Earth's rotation. They are changing in time and therefore can be considered as time series. In many applications, such as geolocation or high-precision satellite navigation, not only observed EOP values are needed, but also their predictions for several days in the future. Namely, there are five time series of interest: coordinates of the pole ( $x, y$ ), length of day  $LOD$ , and celestial pole offsets ( $dX, dY$ ). The coordinates ( $x, y$ ) of the celestial ephemeris pole determine the position of the celestial pole on the Earth's surface. The pole moves slowly because the axis of the Earth's instantaneous rotation does not stay still. Universal time ( $UT1$ ) is the time of the Earth clock,

which performs one revolution in about 24 hours. The excess revolution time is called the length of day (*LOD*). It is vulnerable to wind and world ocean movements; thus it has a complex structure and cannot be modeled precisely. The celestial pole position is described by IAU (International Astronomical Union) precession and nutation model IAU2000. The approximation accuracy is high; however, some nutations are still not predictable. The observed corrections to the modeled celestial pole coordinates in the celestial reference system are the offsets ( $dX, dY$ ).

The Earth orientation parameters prediction comparison campaign (EOP PCC, 2005–2008) was organized for comparing the existing prediction techniques in the same period of time for various lengths of forecasts: 10, 30, and 500 days. The results of the campaign [183] suggest that smaller prediction errors can be achieved by ensembling various models. A similar initiative was undertaken by the International Earth Rotation and Reference Systems Service (IERS) in 2010–2015; unfortunately, the project’s website and its archive are not available at the moment.

Various methods have been applied to the problem of EOP time series forecasting. Least squares interpolation using a harmonic model and autoregressive prediction (LS+AR) was applied to  $x, y$  and *LOD* time series in [184, 185]. A seasonal autoregressive model was proposed for the prediction of  $dX$  and  $dY$  in [186].

Singular spectrum analysis (SSA) was used in several recent works for the prediction of the polar motion ( $x, y$ ). The applicability of the method to the problem was demonstrated in [187] where the parameters were fixed to manually chosen values. The paper [188] proposes a more flexible approach with a combination of SSA and copula-based analysis.

Following [28], we introduce a technique for the (almost) fully automated procedure for the choice of parameters for each of the five EOP time series predictions, which are performed on the base of historical data by means of singular spectrum analysis. The set of parameters consists of both parameters of SSA and the length of the time series to forecast. The used approach to the parameter choice is commonly used in machine learning; therefore, it is particularly interesting to note that the forecasting accuracy appears to be comparable with other public forecasts, as it is demonstrated in this work.

### **Common consideration about the choice of the SSA parameters**

Certainly, the forecasting accuracy depends on the proper choice of parameters that was discussed in Section 3.4. SSA can be considered both a parametric and a non-parametric method. If the trajectory matrix of a signal  $S_N$  is rank-deficient, the signal is called a time series of finite rank. Time series of finite rank (under some unrestrictive limitations) can be exactly forecasted by SSA forecasting methods. For example, sine waves and their sums are of finite rank. For the SSA forecasting of periodic series, the period should not be known in advance, due to the non-parametric nature of SSA. The choice  $L \sim N/2$  is recommended for forecasting finite-rank signals in presence of noise.

If the signal is an amplitude-modulated sine wave, it is not necessarily exactly of finite rank.

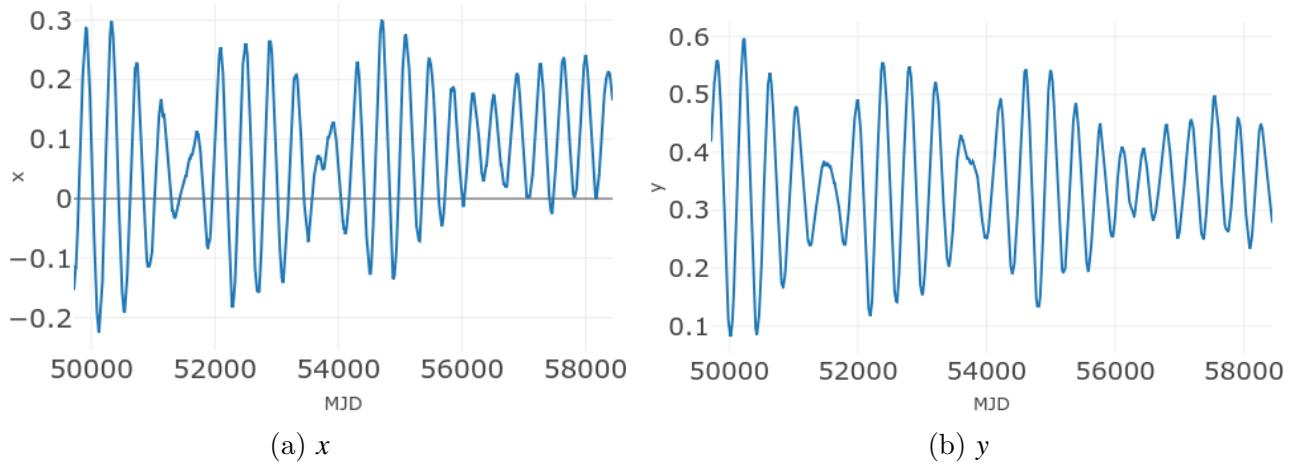


Figure 7.1: Examples of celestial pole time series from 1995 to present.

However, such a signal can be extracted with good accuracy by an increase of the number  $r$  of the chosen components and/or a decrease of the window length  $L$ .

In the interactive version of SSA, the identification of signal components is performed by the analysis of the decomposition eigentriples. However, if the time series is long enough, some automatic procedure for the choice of  $L$  and  $r$  can be applied to minimize forecasting errors on chosen historical data; see e.g. [30, Fragment 3.5.13–3.5.15]. Moreover, if the time series is long and has a changing structure, the question of the choice of the optimal time series length to use for forecasting arises. The problem of the choice of optimal time series length can be solved automatically for a very long time series only.

### 7.1.1. Data sources

As source data, we use IERS 14 C04 bulletin data published by IERS [189]. This source contains daily data for each of the EOP time series beginning from 1962, January 1, for  $x, y, LOD$  and from 1984, January 1, for  $dX, dY$  until the current date with approximately 30 days delay. The plots of the time series are shown in Figures 7.1–7.3.

IERS Bulletin A contains predictions of  $x, y, LOD$  time series for 365 days; these data are published weekly. For comparison of the forecasts, we will also use Pulkovo observatory daily predictions of all EOP time series for 365 days [190]. Unfortunately, as of June 2019, the online archive with Pulkovo predictions is not available. Further, in this work, we will use a backup copy of those files that we possess at the moment.

### 7.1.2. Automatic choice of parameters

We will perform forecasting of the EOP time series  $x, y, LOD, dX, dY$  for one year. For simplicity, we consider one year as consisting of 365 days.

As we mentioned, the SSA forecasting of a signal needs two parameters: the window length  $L$

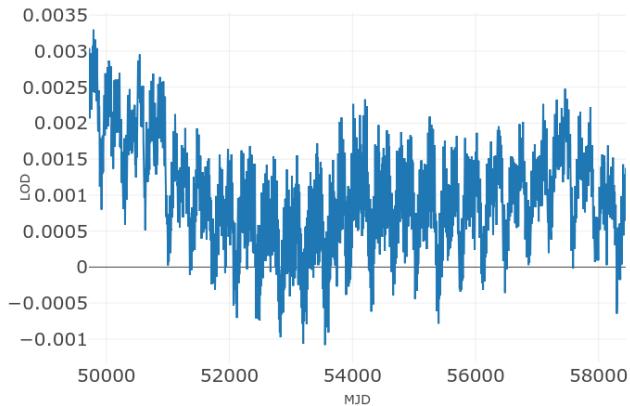


Figure 7.2: Example of  $LOD$  time series from 1995 to present.

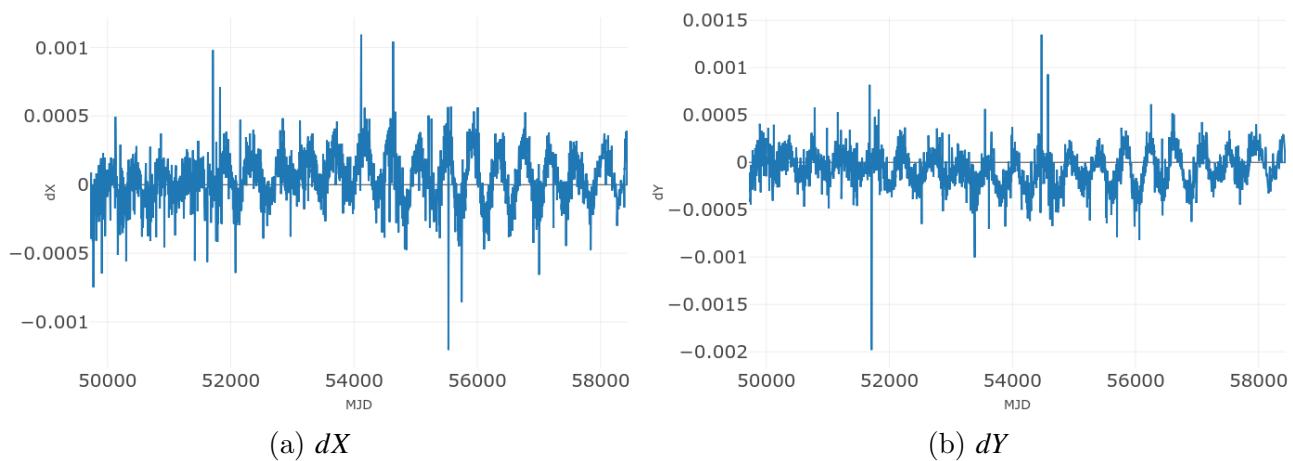


Figure 7.3: Examples of celestial pole offsets time series from 1995 to present.

and the number of leading components  $r$ . The latter is related to the signal structure, which is determined by the signal rank. The following assertions are related: the signal structure is more complex, a larger rank is used for its approximation, and a larger number  $r$  of components in the SSA decomposition should be chosen. We will search for an optimal  $r^*$  value that minimizes forecast mean squared error (MSE) for historical data. The considered boundaries are  $r^* \leq 30$  for the  $x, y, LOD$  time series, and  $r^* \leq 5$  for the  $dX, dY$  time series since they have a simpler structure. These boundaries were checked by the manual SSA analysis of several sample data. The boundaries for  $L$  were chosen similarly.

We use the grid-search for the optimal value  $L^*$  performing up to 10 steps within the range. The chosen grid for  $L$  and  $r$  values is shown in Table 7.1.

Table 7.1: Search grid for  $L$  and  $r$  parameter values.

EOP	$L$ values	$r$ values
$x, y$	300, 500, 700, 900, 1100, 1300, 1500, 1700, 1900, 2100	[1, 30]
$LOD$	300, 600, 900, 1200, 1500, 1800, 2100, 2400, 2700, 3000	[1, 30]
$dX, dY$	250, 300, 350, 400, 450, 500	[1, 5]

To choose the parameters  $L^*, r^*$  for SSA forecasting, we will use the time series cross-validation procedure as described in [30, Section 3.5.7].

For performing the cross-validation, we should fix the training period. Within the training period, we consider moving intervals of length  $Q + 1$  years, where the interval of  $Q$  years ( $365 \times Q$  days) is used for the signal estimation and one year (365 days) is taken for the signal forecasting and error calculation. The training period length is equal to  $Q$  years plus the length  $W$  of the validation period; the validation period consists of the time series values, which are used for the calculation of the forecasting errors that are involved in the cross-validation procedure. For decreasing the computational costs, we consider  $M$  moving intervals of length  $Q + 1$  years with equal lags; thereby, the lag size is approximately equal to  $(W - 1)/(M - 1)$ . For the SSA forecasting, the values  $L^*, r^*$  are chosen from the grid to minimize the forecast MSEs averaging by all the moving intervals.

In [185], the values  $Q = 5, 10, 15$  years were considered. We compare  $Q = 5, 10, 15, 20$  years (if the time series lengths allow this choice) for the validation period length equal to 5 years. The optimal parameters  $L^*, r^*$  are chosen automatically using the cross-validation procedure as described above with  $M = 10$  folds.

The implementation of the SSA algorithms in R language from the RSSA package [70] was used. The average mean-squared errors of the forecasts are shown in Table 7.2. Among the considered values, the value  $Q = 15$  years results in smaller forecast errors. We will use this setting later for performing forecasts on the test period.

We also need to choose the validation period length. To do this,  $Q$  is fixed to that chosen on the previous step (15 years) and then the forecast errors are compared using the cross-validation with  $M = 10$  folds for different lengths of the validation period: 3, 5, 7, and 10 years. The results of the experiments are presented in Table 7.3. We also checked (not shown) that the optimal  $r^*$

Table 7.2: Average MSE of 365 days EOP forecasts for different  $Q$  in the 2006–2010 years interval.

EOP	5 years	10 years	15 years	20 years
$x$	$2.6 \times 10^{-3}$	$9.5 \times 10^{-4}$	$8.5 \times 10^{-4}$	$1.1 \times 10^{-3}$
$y$	$1.9 \times 10^{-3}$	$1.2 \times 10^{-3}$	$9.0 \times 10^{-4}$	$1.1 \times 10^{-3}$
$LOD$	$2.4 \times 10^{-7}$	$1.9 \times 10^{-7}$	$1.0 \times 10^{-7}$	$1.0 \times 10^{-7}$
$dX$	$2.01 \times 10^{-8}$	$1.93 \times 10^{-8}$	$1.88 \times 10^{-8}$	—
$dY$	$1.63 \times 10^{-8}$	$1.53 \times 10^{-8}$	$1.42 \times 10^{-8}$	—

Table 7.3: Average MSE of 365 days EOP forecasts for different validation period lengths in 2006–2010 years interval.

EOP	3 years	5 years	7 years	10 years
$x$	$1.0 \times 10^{-3}$	$8.5 \times 10^{-4}$	$8.3 \times 10^{-4}$	$9.1 \times 10^{-4}$
$y$	$9.2 \times 10^{-4}$	$9.0 \times 10^{-4}$	$8.8 \times 10^{-4}$	$8.8 \times 10^{-4}$
$LOD$	$1.1 \times 10^{-7}$	$1.0 \times 10^{-7}$	$1.0 \times 10^{-7}$	$9.8 \times 10^{-8}$
$dX$	$1.91 \times 10^{-8}$	$1.88 \times 10^{-8}$	$1.88 \times 10^{-8}$	—
$dY$	$1.42 \times 10^{-8}$	$1.42 \times 10^{-8}$	$1.39 \times 10^{-8}$	—

Table 7.4: Average MSE of 365 days EOP weekly forecasts from different sources in 2011–2015 years interval.

EOP	SSA	Pulkovo AM	Bulletin A
$x$	$7.2 \times 10^{-4}$	$8.6 \times 10^{-4}$	$7.5 \times 10^{-4}$
$y$	$6.1 \times 10^{-4}$	$7.6 \times 10^{-4}$	$8.5 \times 10^{-4}$
$LOD$	$9.1 \times 10^{-8}$	$1.0 \times 10^{-7}$	—
$dX$	$1.3 \times 10^{-8}$	$1.1 \times 10^{-8}$	—
$dY$	$1.6 \times 10^{-8}$	$2.2 \times 10^{-8}$	—

and  $L^*$  values fall in the same intervals for different forecasts along the considered training period. This confirms that their choice is not random.

### 7.1.3. Forecasts on the test period

To perform forecasts for 365 days in the test period (2011–2015 years), we used the results of investigations described in Section 7.1.2. The lengths of the training and validation periods were chosen with the help of Tables 7.2 and 7.3, respectively. For forecasts, the parameters  $L^*, r^*$  were chosen automatically using the cross-validation procedure as described in Section 7.1.2 (see Figure 7.4 for examples of forecasts). The forecasting accuracy was compared with that of predictions from two sources available in the public domain: Pulkovo observatory [190] and IERS Bulletin A [191].

Since Bulletin A forecasts are published once a week, we generate forecasts each week starting from the dates of publications for each of the EOP time series. Then we calculate mean squared errors for all the forecasts and take the average MSE in the test period. The results are shown in Table 7.4. In most cases, the proposed method demonstrates a better average performance, except for the  $dX$  time series predictions published by Pulkovo observatory. However, the average MSEs of SSA and Pulkovo forecasts for  $dX$  are of the same order.

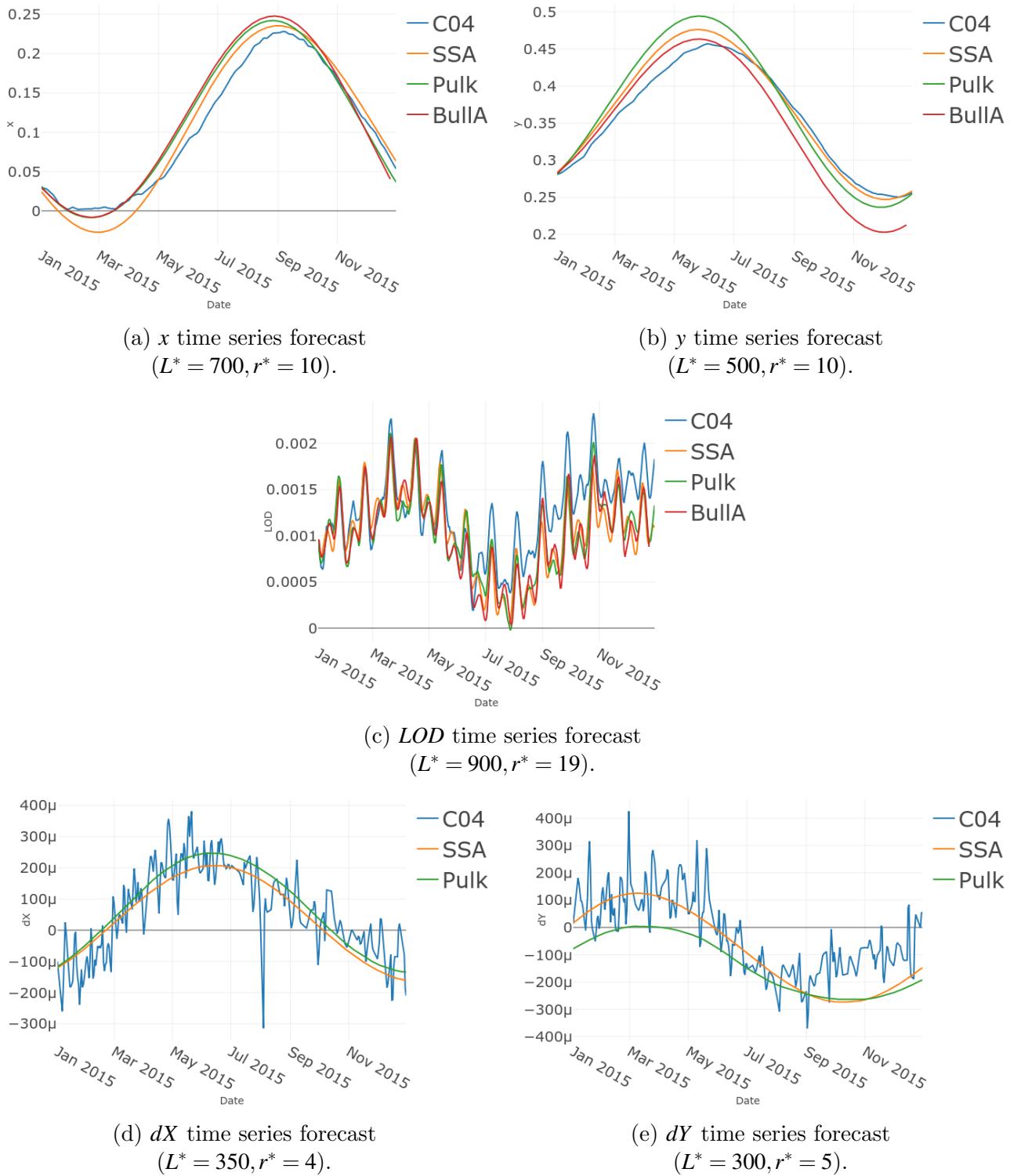


Figure 7.4: EOP time series forecasts for 365 days starting from January 1 2015 from different sources.

### 7.1.4. Conclusions

It appears that SSA is well-suitable for the prediction of the EOP time series. Since the computational cost of SSA is considerably diminished in the recent implementation [70, 30], the proposed approach is computationally feasible for daily EOP time series predictions. The numerical comparison in the test period from January 1, 2011, till December 31, 2015, shows that SSA together with the proposed method of automatization of the parameter choice provides forecasting accuracy, which is comparable to those that are published by international services; in many cases, the proposed method's errors are smaller on average.

The resulting forecasts and corresponding parameter values are published as a web application and can be downloaded via <http://eoppredict.ru>. The interface allows one to compare predictions from different sources on historical data.

## 7.2. Application of SSA to density estimation

For the fundamental problem of nonparametric density estimation various approaches have been proposed in the literature. The most widely used approach is kernel smoothing thoroughly discussed in [192, 193].

This example aims to show how SSA can be involved in density smoothing. Let us present a new approach to nonparametric estimation of distribution functions and densities following [7]. The basic idea is to construct an artificial time series by evaluating the empirical distribution function at equidistant points and consider it as a time series.

### 7.2.1. A new approach to density estimation

Let  $x_1, \dots, x_m$  be a sample from a distribution having a density  $p(x)$  and let  $F(x)$  be the corresponding cumulative distribution function. The empirical distribution function

$$F_m(x) = \frac{1}{m} \sum_{i=1}^m \mathbf{1}_{[x_i, \infty)}(x), \quad x \in \mathbb{R},$$

is the minimum-variance unbiased nonparametric estimate of  $F(x)$ . However, it cannot be used directly to estimate the density  $p(x)$  by taking the derivative since  $F_m(x)$  is a step function, and it is known that estimation can often be improved substantially by using smoothed but biased estimates of  $F(x)$ . To smooth the empirical distribution function, we propose to consider the values of  $F_m(x)$  on an equidistant dense grid as a series and then apply the SSA procedure to smooth this series.

## Outline of the approach

Let  $t_1, \dots, t_N$  be equidistant points such that  $t_j - t_{j-1} = \delta$ ,  $j = 2, \dots, N$ ,  $t_1 < \min_i x_i$  and  $t_N > \max_i x_i$ , where  $\delta$  is a small positive constant. We then define the series

$$\mathcal{F}_m = (f_1, \dots, f_N), \quad f_j = F_m(t_j), \quad j = 1, \dots, N,$$

which is nondecreasing and serves as a discrete version of  $F_m(x)$ , and the series

$$\mathcal{F}^* = (f_1^*, \dots, f_N^*), \quad f_j^* = F(t_j), \quad j = 1, \dots, N,$$

which has to be estimated from the sample.

We now choose a certain positive integer  $L$  satisfying  $1 \leq L \leq (N+1)/2$ , which will regulate the smoothness of the proposed estimate and is a parameter of the following procedure adapted from SSA theory, cf. [1]. If we consider the smoothing procedure from the viewpoint of filtration,  $L$  controls the filter bandwidth. Also, the parameter  $L$  has the sense of resolution, that is, the smaller  $L$ , the more refined and less stable smoothing is performed. The natural upper bound for the selection of the smoothing parameter  $L$  is

$$L_{\max} = \left\lfloor \frac{\max_i x_i - \min_i x_i}{2\delta} \right\rfloor.$$

Note that the series  $\mathcal{F}_m$  may have any number of zeroes to the left and any number of ones to the right. Extension of the series  $\mathcal{F}_m$  enables us to remove the boundary effects of the filters constructed in what follows.

The SSA algorithm for processing the series  $\mathcal{F}_m$  now works as follows. The trajectory matrix and its SVD are constructed in the same way as in Basic SSA. According to the SSA theory, a few leading terms of the SVD are of interest for the problem of smoothing [1, Section 1.3.2]. Therefore, we define the matrix

$$\mathbb{X}^{(r)} = (x_{i,j}^{(r)}) = \sum_{i=1}^r \sqrt{\lambda_i} U_i V_i^T. \quad (7.1)$$

The last step of the SSA algorithm is the Hankelization (averaging along anti-diagonals) of  $\mathbb{X}^{(r)}$  with a subsequent operation that is opposite to embedding. Taking into consideration the specifics of cumulative distribution functions, we define

$$\hat{f}_j = \hat{f}_j(L, r) = \begin{cases} 0 & 1 \leq j < L, \\ \frac{1}{L} \sum_{k=1}^L x_{k,j-k+1}^{(r)} & L \leq j \leq K, \\ 1 & K < j \leq N, \end{cases}$$

Thus, we obtain the series

$$\hat{\mathcal{F}}_m = \hat{\mathcal{F}}_m(L, r) = \left( \hat{f}_1(L, r), \dots, \hat{f}_N(L, r) \right),$$

which will be called the SSA estimate of  $\mathcal{F}^*$ . By construction, we have  $\hat{\mathcal{F}}_m(L, r) = \mathcal{F}_m$  if  $L = r$ .

The series  $\hat{\mathcal{F}}_m$  can be transformed to the SSA estimate of  $F(x)$  using the linear interpolation as follows

$$\hat{F}_m(x) = \sum_{j=2}^N \left( \hat{f}_{j-1} + (\hat{f}_j - \hat{f}_{j-1}) \frac{x - t_{j-1}}{t_j - t_{j-1}} \right) \mathbf{1}_{[t_{j-1}, t_j)}(x) + \mathbf{1}_{[t_N, \infty)}(x), \quad (7.2)$$

$x \in \mathbb{R}$ . Note that  $\hat{F}_m(x)$  is actually a linear spline on  $[t_1, t_N]$  with knots  $(t_1, \hat{f}_1), \dots, (t_N, \hat{f}_N)$ . Moreover, we can consider the derivative of  $\hat{F}_m(x)$  from the left as an estimate of the density. In particular,

$$\hat{p} = (\hat{p}_1, \dots, \hat{p}_N), \quad \hat{p}_j = (\hat{f}_j - \hat{f}_{j-1})/\delta,$$

$j = 2, \dots, N$ , gives an estimate of the density  $p(x)$  at the equidistant points.

Let us now describe how to select the points  $t_1, \dots, t_N$  used to transform the empirical distribution function to a series. To do this, we first notice that  $V_i = \mathbb{X}^T U_i / \sqrt{\lambda_i}$  and  $\mathbb{X}^{(r)} = (U_1 U_1^T + \dots + U_r U_r^T) \mathbb{X}$ , and, consequently,  $\hat{f}_j$  enables the representation

$$\hat{f}_j = \sum_{i=1}^L \sum_{l=1}^L (u_{1,i} u_{1,l} + \dots + u_{r,i} u_{r,l}) f_{j+i-l} / L \quad (7.3)$$

for  $j = L, \dots, K$ , where  $(u_{l,1}, \dots, u_{l,L})^T = U_l$ . Thus, formula (7.3) means that the SSA procedure creates a data-adaptive filter of size  $2L-1$ , where  $r$  can be interpreted as the complexity of the SSA filter and  $L$  controls the smoothness of the filtered series. Therefore, we should determine the points  $t_1, \dots, t_N$  such that the series  $\mathcal{F}_m$  contains  $2L$  zeroes at the beginning and  $2L$  ones at the end.

We suggest choosing  $\delta$  to satisfy the inequality

$$\mathbf{P}(\xi \in [x, x + \delta]) \approx p(x)\delta < 1/m,$$

where the density  $p(x)$  is assumed to be smooth enough. Then

$$\delta \leq \frac{1}{m \max_x p(x)} = \frac{\sigma}{m \max_x p_{\text{st}}(x)}, \quad (7.4)$$

where  $p_{\text{st}}(x)$  is the standardized density having zero mean and unit variance,  $p_{\text{st}}(x) = p((x - a)/\sigma)$ . This inequality provides the reasonable correspondence between the sample size and the interpolation step  $\delta$ . In the examples of the present work we consider  $\delta = 0.01\sigma$  that approximately satisfies inequality (7.4).

Summarizing above arguments, we define points  $t_1, \dots, t_N$  by

$$t_j = \delta(j - 2L) + \min_{i=1, \dots, m} x_i,$$

$$j = 1, \dots, N, N = 2L_{\max} + 4L.$$

Note that the specifics of the analyzed series (namely, monotonicity and slow variation) allow us to consider only a few leading components in (1.12) for smoothing. Contrary to the problems of signal extraction with fixed  $L$  and estimated  $r$ , we fix  $r$  and adjust  $L$  to control the smoothness.

For clarity, we now demonstrate the influence of the parameter  $L$  on the SSA estimate  $\hat{F}_m(L, r)$  with  $r = 1$  visually.

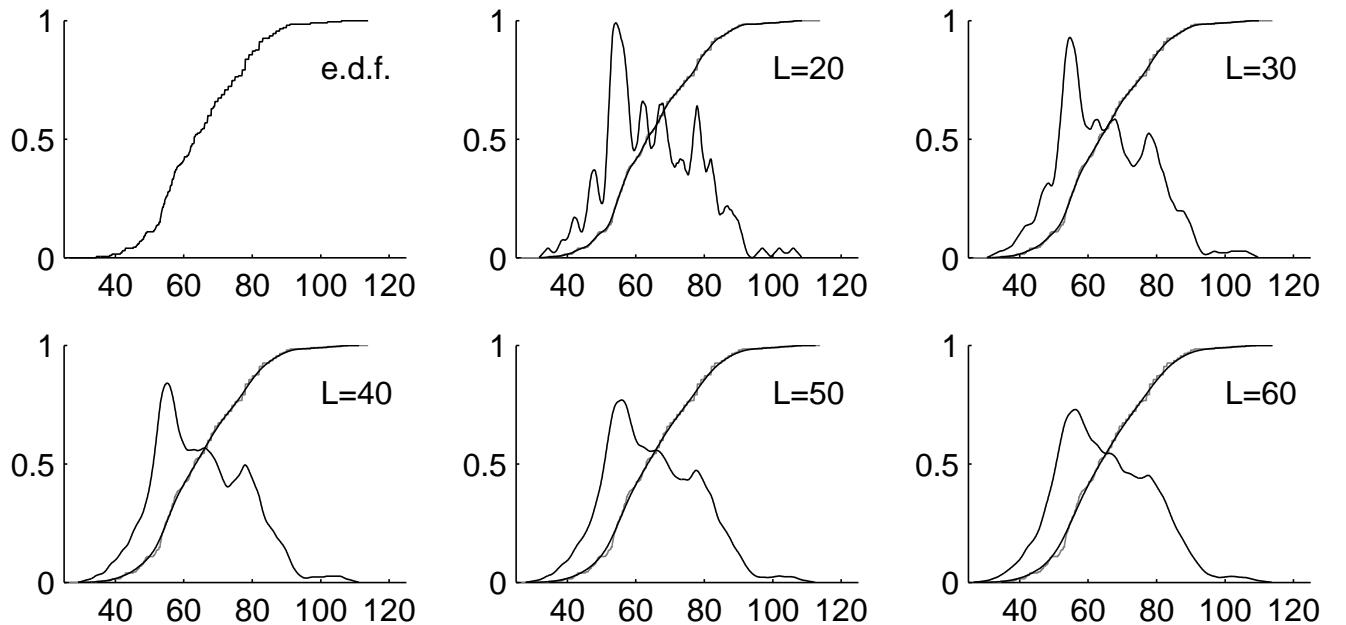


Figure 7.5: The empirical distribution function for measurements of Lean Body Mass from the Australian Institute of Sport data and the SSA estimates of the distribution function and density for  $L = 20, 30, \dots, 60$ .

**Example 4.** Let us consider 202 measurements of Lean Body Mass from the Australian Institute of Sport, which were studied by [194], amongst others. Figure 7.5 depicts the empirical distribution function for these measurements (which give  $\delta = 0.13$  and  $L_{\max} = 274$ ) and the SSA estimates for  $L = 20, 30, \dots, 60$ . The densities in the figure are re-scaled for convenience of visual impression. It can be seen that the SSA estimate  $\hat{F}_m(L, 1)$  becomes smoother and the number of modes of the corresponding density decreases as the parameter  $L$  increases. Compared to kernel smoothing, small values of  $L$  correspond to small bandwidths and large values of  $L$  to large bandwidths. Both methods lead to relatively similar results for this data set.

**Example 5.** As a second example, we consider 22 observations of silica in chondrite meteors [195]. In Figure 7.6, we depict the empirical distribution function for these observations (which give

$\delta = 0.043$  and  $L_{\max} = 163$ ) and the SSA estimates for  $L = 20, 40, \dots, 100$ . Notice that the density has three modes for a large range of values of  $L$ . In general, the SSA estimate may be used to investigate multimodality in a way as discussed in [193].

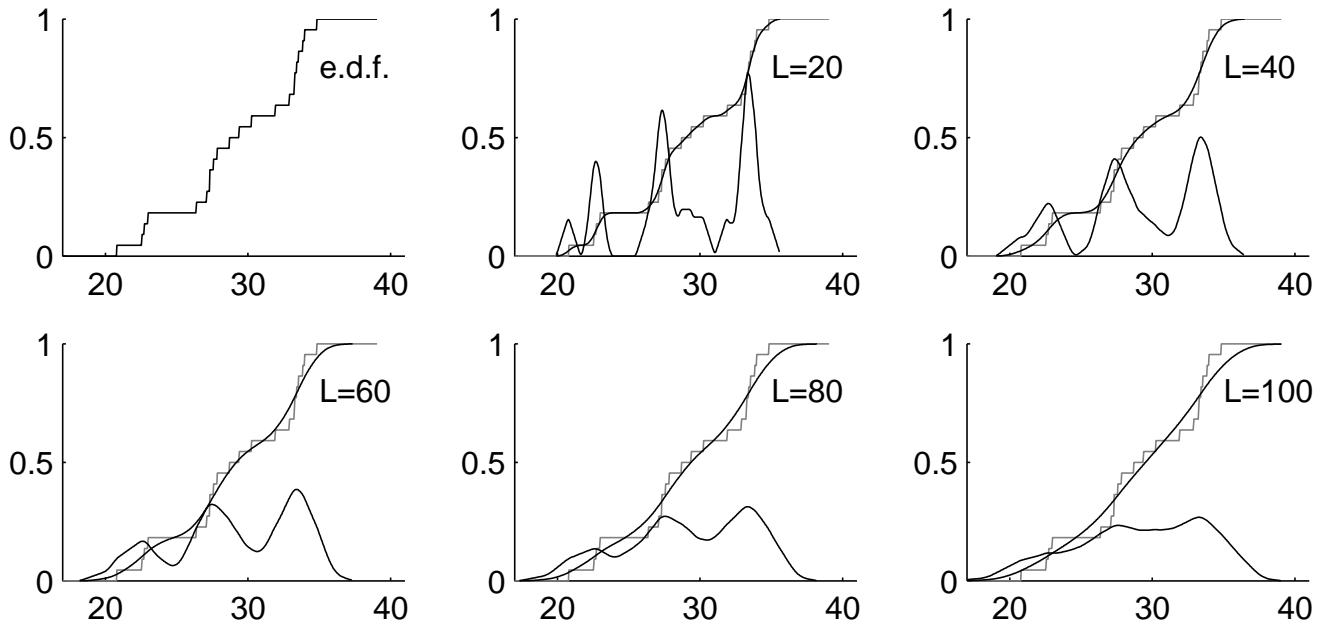


Figure 7.6: The empirical distribution function for observations of chondrite meteors and the SSA estimates of the distribution function and density for  $L = 20, 40, \dots, 100$ .

### Specific SSA estimates and bias correction

It is worth studying the SSA estimate for selected values of  $r$ , the number of leading components used in (7.1), in greater detail and to develop a procedure to ensure that the final estimate is monotone.

The SSA estimate, like any smoothing procedure, suffers from estimation bias. Therefore, we propose an effective approach for bias correction, where the bias is estimated by repeated applications of the SSA filter. A simulation study evaluates the performance of the SSA estimates with  $r = 1$ ,  $r = 2$ , and the bias-corrected version, in terms of three standard measures including the integrated mean squared error and the Kolmogorov-Smirnov distance.

**The SSA estimate with  $r = 1$**  The case of  $r = 1$  provides the simplest estimate, which will be denoted as the  $\text{SSA}^{1c}$  estimate, since one component of the SVD decomposition (1.12) is used. In the following lemma, we establish that the SSA algorithm with  $r = 1$  yields a valid result.

**Lemma 18.** *The  $\text{SSA}^{1c}$  estimate (7.2) is a valid distribution function.*

**Proof.** Since the elements of the series  $\mathcal{F}_m$  are positive, the elements of  $\mathbf{X}\mathbf{X}^T$  are also positive. This implies that the elements of the leading eigenvector  $U_1$  have the same sign, which may be assumed to be positive. Therefore, the monotonicity of the  $\text{SSA}^{1c}$  estimate follows from (7.3).

Applying Jensen's inequality, we obtain

$$\sum_{i=1}^L \sum_{l=1}^L u_{1,i} u_{1,l} / L = \left( \sum_{i=1}^L u_{1,i} \right)^2 / L \leq \sum_{i=1}^L u_{1,i}^2 = 1.$$

Therefore, using (7.3) we have  $0 \leq \hat{f}_j \leq 1$ , which completes the proof.  $\square$

**The SSA estimate with  $r = 2$**  We note that the estimate  $\hat{\mathcal{F}}_m(L, 2)$  (i.e., the SSA estimate with  $r = 2$ ) may be a series that is not necessarily a discrete version of a distribution function. In particular, the series  $\hat{\mathcal{F}}_m(L, 2)$  can be non-monotonic. Let us introduce an operator  $M(\mathcal{H})$  for a given non-decreasing series  $\mathcal{H} = (H_1, \dots, H_N)$ ,  $H_i \in \mathbb{R}$ ,  $1 \leq i \leq N$ , such that  $M(\mathcal{H})$  is a discrete version of a distribution function and close to  $\mathcal{H}$ . This can be achieved by replacing all negative values with zeroes and all values exceeding one by ones, and then taking the average of left and right monotonic functions. Thus, we define SSA<sup>2c</sup> estimate as  $\hat{\mathcal{F}}_m^{2c}(L) = M(\hat{\mathcal{F}}_m(L, 2))$ .

Our numerical results (see Table 7.6) indicate that the SSA estimate with  $r > 2$  is, on average, not really better. This can be explained as follows. In SSA theory, one of the aims of SSA is to extract a series of finite rank  $\tau$  from a noisy series, where a series of finite rank  $\tau$  is a series whose trajectory matrix  $\mathbb{X}$  has exactly  $\tau$  nonzero eigenvalues. Since distribution functions are generally not of finite rank, the problem of approximating the series  $\mathcal{F}^*$  by a series of finite rank arises. The accuracy of this approximation depends on  $L$  and  $r$ , which jointly give too much freedom. Therefore, the SSA estimate with  $r = 2$  is typically sufficient for practical purposes.

**The SSA estimate with  $r = 1$  and bias correction** Recall that the SSA<sup>1c</sup> estimate is a linear filter of size  $2L - 1$ . Moreover, the proof of Lemma 1 shows that this filter is a weighted moving average with positive coefficients. Therefore, the SSA<sup>1c</sup> estimate generally has a bias, which is desirable to remove.

Therefore we introduce the operator  $S$  defined as the SSA<sup>1c</sup> estimate, that is,

$$S(\mathcal{F}_m|L) = \hat{\mathcal{F}}_m(L, 1).$$

We also consider the operator  $S^k$  which means repeated applications of the operator  $S$ , that is,  $S^k = S(S^{k-1})$ . Note that  $S^k(\mathcal{F}|L)$  means that the SSA<sup>1c</sup> filter is consequently applied  $k$  times. To compute  $S^k(\mathcal{F}|L)$ , a series  $\mathcal{F}$  should have at least  $(1+k)L$  zeroes at the beginning and  $(1+k)L$  ones at the end. For further consideration, we suppose that  $m$  is large enough such that  $S(\mathcal{F}^*|L) \approx S(\mathcal{F}_m|L)$  and  $L$  is relatively small.

Let us define the operator

$$S^b(\mathcal{F}_m|L) = 3S(\mathcal{F}_m|L) - 3S^2(\mathcal{F}_m|L) + S^3(\mathcal{F}_m|L),$$

which has a smaller bias than the operator  $S$  as proved in Section 7.2.2. We notice that the

series  $S^b(\mathcal{F}|L)$  is not necessarily monotonic but it changes from 0 to 1. Therefore, the series  $\hat{F}_m^b(L) = M(S^b(\mathcal{F}_m|L))$  gives an estimate which will be called the SSA<sup>b</sup> estimate of  $\mathcal{F}^*$ .

### 7.2.2. Performance of the SSA estimates

To gain some insight into the finite sample properties, we investigate the SSA estimates by simulation and compare them with the kernel density estimate with the least-squares cross-validation (LSCV) bandwidth.

To do this, we consider the integrated squared error (ISE)

$$D_{\text{ISE}}(\hat{p}) = \int (\hat{p}(x) - p(x))^2 dx,$$

the Kolmogorov-Smirnov distance

$$D_{\text{KS}}(\hat{F}) = \|\hat{F} - F\|_\infty = \max_x |\hat{F}(x) - F(x)|$$

and the Hellinger distance

$$D_{\text{H}}(\hat{p}) = \int (\sqrt{\hat{p}(x)} - \sqrt{p(x)})^2 dx,$$

where  $p(x)$  is the density and  $F(x)$  is the distribution function for the assumed model.

We consider two models: the normal distribution  $N(0,1)$  and the mixture  $0.4N(0,1) + 0.6N(5,2^2)$ , and compute the average values of the above distances using 10000 simulated samples of size 100. These settings give  $\delta \approx 0.01$  and  $L_{\max} \approx 240$  for the first model and  $\delta \approx 0.03$  and  $L_{\max} \approx 200$  for the second model.

In Figure 7.7, we present the kernel density estimates and the SSA estimates with  $L = 100$  and  $L = 130$  for five samples of size 100 from  $N(0,1)$ , while these estimates with  $L = 60$  and  $L = 70$  for the mixture  $0.4N(0,1) + 0.6N(5,2^2)$  are depicted in Figure 7.8. We can observe that the SSA estimates are slightly smoother than the kernel density estimates with the LSCV bandwidth.

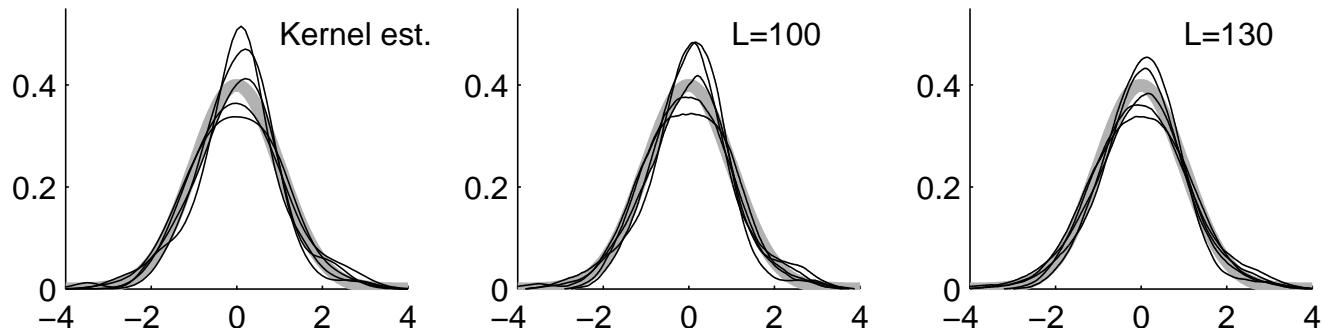


Figure 7.7: The kernel density estimates and the SSA<sup>1c</sup> estimates with  $L = 100$  and  $L = 130$  for samples of size 100 from  $N(0,1)$ , whose density is given in grey.

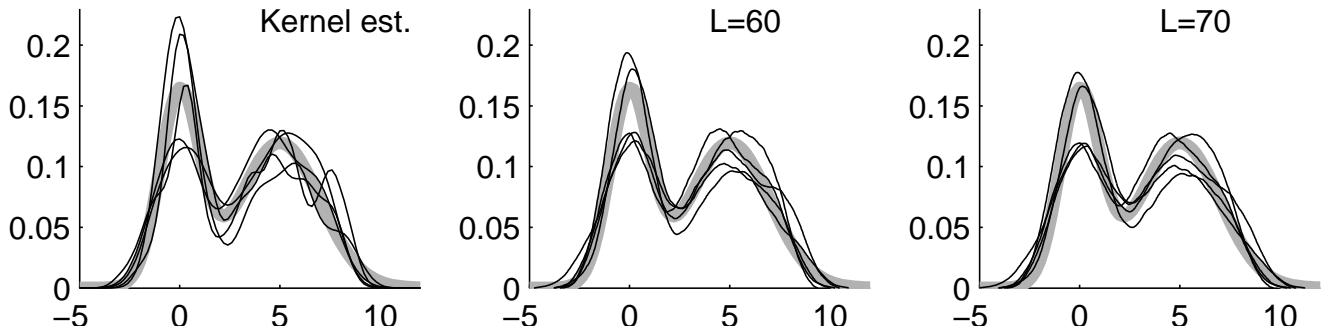


Figure 7.8: The kernel density estimates and the  $\text{SSA}^{1c}$  estimates with  $L = 60$  and  $L = 70$  for samples of size 100 from  $0.4N(0, 1) + 0.6N(5, 2^2)$ , whose density is given in grey.

Table 7.5 shows the means of the ISE, the Kolmogorov-Smirnov and Hellinger distances of the estimates to the true distribution functions. We can see that the  $\text{SSA}^{1c}$ ,  $\text{SSA}^{2c}$  and  $\text{SSA}^b$  estimates with  $L$  from a wide range perform better in terms of all three criteria than the kernel density estimates.

Table 7.5: Means of the ISE, the Kolmogorov-Smirnov and Hellinger distances for the kernel,  $\text{SSA}^{1c}$ ,  $\text{SSA}^{2c}$  and  $\text{SSA}^b$  density estimates for samples of size 100.

	$ED_{\text{ISE}}$	$ED_{\text{KS}}$	$ED_{\text{H}}$
model $N(0, 1)$			
Kernel est. with $h_{\text{LSCV}}$	0.0071	0.0551	0.0143
$\text{SSA}^{1c}$ est. with $L=80$	0.0065	0.0521	0.0122
$\text{SSA}^{1c}$ est. with $L=100$	0.0055	0.0509	0.0113
$\text{SSA}^{1c}$ est. with $L=130$	0.0061	0.0550	0.0139
$\text{SSA}^{2c}$ est. with $L=160$	0.0054	0.0496	0.0135
$\text{SSA}^{2c}$ est. with $L=190$	0.0048	0.0486	0.0122
$\text{SSA}^{2c}$ est. with $L=220$	0.0048	0.0493	0.0117
$\text{SSA}^b$ est. with $L=130$	0.0054	0.0497	0.0144
$\text{SSA}^b$ est. with $L=160$	0.0046	0.0479	0.0135
$\text{SSA}^b$ est. with $L=190$	0.0048	0.0492	0.0137
model $0.4N(0, 1) + 0.6N(5, 2^2)$			
Kernel est. with $h_{\text{LSCV}}$	0.0058	0.0617	0.0231
$\text{SSA}^{1c}$ est. with $L=50$	0.0044	0.0595	0.0172
$\text{SSA}^{1c}$ est. with $L=60$	0.0044	0.0593	0.0179
$\text{SSA}^{1c}$ est. with $L=70$	0.0048	0.0603	0.0204
$\text{SSA}^{2c}$ est. with $L=80$	0.0045	0.0598	0.0190
$\text{SSA}^{2c}$ est. with $L=90$	0.0042	0.0591	0.0179
$\text{SSA}^{2c}$ est. with $L=100$	0.0042	0.0588	0.0174
$\text{SSA}^b$ est. with $L=70$	0.0044	0.0608	0.0193
$\text{SSA}^b$ est. with $L=80$	0.0042	0.0602	0.0182
$\text{SSA}^b$ est. with $L=90$	0.0044	0.0607	0.0185

We notice that larger values of  $L$  are used for the model given by the normal distribution and

smaller values of  $L$  are used for the model with two modes. This is similar to the behavior of the LSCV bandwidth, which decreases as a distribution becomes less normal.

We can observe that each SSA estimate has its range of favorable values of  $L$ . The SSA<sup>b</sup> estimate should be used with larger values of  $L$  since this gives greater smoothness while the SSA<sup>1c</sup> estimate for such  $L$  has a noticeable bias. Meanwhile, the ranges for the SSA<sup>2c</sup> and SSA<sup>b</sup> estimates are relatively similar.

An automatic procedure to select a smoothing parameter is supposed by the coauthor Andrey Pepelyshev in [7]; therefore, we refer to this paper and do not put this section here.

### Justification of the bias approximation

Consider the convolution operator  $C(F) = \mu_\epsilon * F$ , where  $\mu_\epsilon$  is a measure with finite moments,

$$C(F)(x) = \int F(x-y)\mu_\epsilon(dy),$$

and we interpret  $\mu_\epsilon$  as a kernel with width  $\epsilon$ . If  $\mu_\epsilon$  is a discrete measure, then  $C(F)$  is a weighted moving-average of  $F$ .

Let our problem be estimating  $F$  by means of linear combinations of  $C^j(F)$  for  $j = 1, 2, \dots$ , where  $C^j$  is the  $j$ -fold composition of the operator  $C$ ,  $C^0(F) = F$ . Therefore, we consider an estimate of  $F$  in the form  $\widehat{F}_P(\mu_\epsilon) = P(\mu_\epsilon) * F$ , where  $P$  is a polynomial with no intercept. We note that this estimate has the required form since  $P(0) = 0$ . Besides, the error of the estimate is equal to  $P(\mu_\epsilon) * F - F = P^{\text{err}}(\mu_\epsilon) * F$ , where  $P^{\text{err}} = P - 1$ .

Let the estimate  $\widehat{F}_P(\mu_\epsilon)$  be generated by  $P_k(z) = 1 - (1-z)^k$ . Then we have  $P_k^{\text{err}}(z) = -(1-z)^k$ . To study this estimate, we define  $w_{k,\epsilon} = \int y^k \mu_\epsilon(dy)$  for  $k \in \mathbb{N}$ ,  $w_{0,\epsilon} = \int \mu_\epsilon(dy) - 1$  and consider the formal Taylor expansion of a smooth function  $F$  at the point  $x$ :

$$F(x-y) = \sum_{i=0}^{\infty} (-1)^i \frac{F^{(i)}(x)}{i!} y^i$$

and its derivatives

$$F^{(j)}(x-y) = \sum_{i=0}^{\infty} (-1)^i \frac{F^{(i+j)}(x)}{i!} y^i.$$

Then we have the following formal expansion of the error term:

$$\widehat{F}_{P_k}(\mu_\epsilon) - F = P_k^{\text{err}}(\mu_\epsilon) * F = \sum_{i_1, \dots, i_k=0}^{\infty} (-1)^{k+1+\sum i_j} \frac{F^{(\sum_{j=1}^k i_j)}(x)}{\prod_{j=1}^k i_j!} \prod_{j=1}^k w_{i_j, \epsilon}. \quad (7.5)$$

Note that the SSA<sup>1c</sup> estimate corresponds to the case  $k = 1$ , where  $P_1(z) = 1 - (1-z) = z$  implies  $\widehat{F}_{P_1}(\mu_\epsilon) = \mu_\epsilon * F = C(F)$ . Meanwhile, the SSA<sup>b</sup> estimate corresponds to the case  $k = 3$ , where  $P_3(z) = 1 - (1-z)^3 = z^3 - 3z^2 + 3z$  entails  $\widehat{F}_{P_3}(\mu_\epsilon) = C^3(F) - 3C^2(F) + C(F)$ .

The formula (7.5) shows that if  $w_{j,\epsilon} \rightarrow 0$  as  $\epsilon \rightarrow 0$  with sufficient rates of convergence (for example,  $w_{j,\epsilon} = O(\epsilon^j)$ ), then the error  $\widehat{F}_{P_k}(\mu_\epsilon) - F$  decreases as  $k$  increases for small  $\epsilon$ .

Recall that the operator  $S$  is an operator  $C$  with the measure  $\mu_\epsilon$  concentrated at the points  $\pm i\delta$ ,  $i = 0, \dots, L$ . Therefore,  $\mu_\epsilon$  is close to the Dirac delta function concentrated at 0 if  $L$  is small. Thus, we can expect that the SSA<sup>b</sup> estimate of a smooth enough distribution function  $F$  has a smaller bias than the SSA<sup>1c</sup> estimate which is confirmed by extensive numerical studies.

## Simulation study

In Table 7.6 we present the performance of the automatic procedure for the SSA and kernel density estimates. We see that the Sheather-Jones plug-in (SJPI) bandwidth is better than the LSCV bandwidth and the Indirect cross-validation (ICV) bandwidth is not quite good for the model of normal distribution. However, the SJPI bandwidth is worse than the LSCV bandwidth if the model has two modes of significantly distinct width. It is worth noticing that the density with the LSCV bandwidth gives a small ISE for the model  $0.3N(0, 1) + 0.7N(15, 4^2)$  but it yields a lot of modes while the density with other bandwidths and the SSA estimates have typically 2 or 3 modes.

We can observe that the SSA<sup>1c</sup>, SSA<sup>b</sup> and SSA<sup>2c</sup> estimates with an automatic choice of the smoothing parameter provide a small ISE compared to the kernel density estimate, while results for the Kolmogorov-Smirnov and Hellinger distances are similar or slightly better. The SSA<sup>3c</sup> estimate is useful only for ‘hard’ distributions, which can be identified by the distinctness of values  $L_a$  for  $r = 1, 2, 3$ . Overall, the SSA<sup>2c</sup> estimate has better performance.

### 7.2.3. Application to c.d.f. estimation in market research

In this section, we analyze a real-data set from a market research study, where the coarsening of the data makes the estimation of quantiles a challenging issue.

## Market research

In market research, observed values are typically coarsening. For example, in questionnaires, age should be filled in as integers while asked prices are a mixture of values rounded to integers and multiples of ten and reported earnings are heaped to hundreds or thousands [196]. In these cases, the computation of quantiles via order statistics does not provide good results. Thus, it is vital to obtain a smoothed distribution and then compute quantiles using it.

As an example, let us consider one variable from a product pricing survey. Specifically, let  $x_i$  be a ticket price such that the  $i$ -th responder will buy a ticket for this price,  $i = 1, \dots, 475$ .

In Figure 7.9 we depict a frequency plot of the sample, the empirical distribution function and the SSA<sup>2c</sup> estimate using the automatic procedure for heaped data. We can see that the majority of responders have rounded prices to multiples of ten. Estimates of the median and quantiles, which are of interest to marketing managers, are  $\tilde{q}_{1/5} = 60$ ,  $\tilde{q}_{1/2} = 70$  and  $\tilde{q}_{4/5} = 80$  that

Table 7.6: Means of the ISE, the Kolmogorov-Smirnov and Hellinger distances for the kernel density estimates with different bandwidths and the SSA<sup>1c</sup>, SSA<sup>b</sup> and SSA<sup>2c</sup> estimates using the automatic procedure for samples of size 100. The last column shows the mean of the parameter  $L_a$ , which is found by the automatic procedure.

	<b>ED<sub>ISE</sub></b>	<b>ED<sub>KS</sub></b>	<b>ED<sub>H</sub></b>	<b>EL<sub>a</sub></b>
model $N(0, 1)$				
Kernel est. with $h_{\text{LSCV}}$	0.0071	0.0551	0.0143	
Kernel est. with $h_{\text{SJPI}}$	0.0066	0.0536	0.0131	
Kernel est. with $h_{\text{ICV}}$	0.0075	0.0546	0.0146	
SSA <sup>1c</sup> est.	0.0061	0.0537	0.0128	107.6
SSA <sup>2c</sup> est.	0.0060	0.0503	0.0142	145.7
SSA <sup>3c</sup> est.	0.0090	0.0556	0.0189	143.4
SSA <sup>b</sup> est.	0.0052	0.0488	0.0141	141.7
model $0.4N(0, 1) + 0.6N(5, 2^2)$				
Kernel est. with $h_{\text{LSCV}}$	0.0058	0.0617	0.0231	
Kernel est. with $h_{\text{SJPI}}$	0.0052	0.0609	0.0210	
Kernel est. with $h_{\text{ICV}}$	0.0055	0.0614	0.0229	
SSA <sup>1c</sup> est.	0.0051	0.0607	0.0215	67.6
SSA <sup>2c</sup> est.	0.0047	0.0610	0.0195	86.4
SSA <sup>3c</sup> est.	0.0054	0.0623	0.0219	90.3
SSA <sup>b</sup> est.	0.0052	0.0617	0.0206	89.7
model $0.3N(0, 1) + 0.7N(15, 4^2)$				
Kernel est. with $h_{\text{LSCV}}$	0.0048	0.0672	0.0394	
Kernel est. with $h_{\text{SJPI}}$	0.0069	0.0733	0.0602	
Kernel est. with $h_{\text{ICV}}$	0.0049	0.0670	0.0396	
SSA <sup>1c</sup> est.	0.0053	0.0679	0.0451	40.9
SSA <sup>2c</sup> est.	0.0046	0.0660	0.0380	60.0
SSA <sup>3c</sup> est.	0.0043	0.0643	0.0349	73.8
SSA <sup>b</sup> est.	0.0047	0.0670	0.0391	48.9

are computed as ordered statistics. However, using the SSA<sup>2c</sup> estimate of the distribution function we obtain  $\hat{q}_{1/5} = 57.27$ ,  $\hat{q}_{1/2} = 71.99$  and  $\hat{q}_{4/5} = 82.14$ , which are more accurate values.

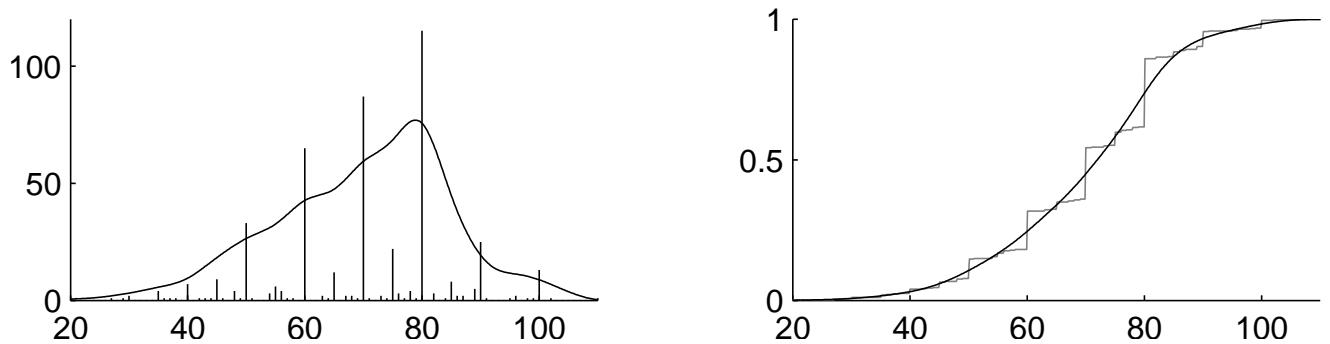


Figure 7.9: Left: The frequency plot of ticket prices and the  $\text{SSA}^{2c}$  density estimate using the automatic procedure for heaped data. Right: The empirical and smoothed distribution functions for a ticket price.

### 7.3. Two-exponential models of gene expression patterns for noisy experimental data

Spatial pattern formation of the primary anterior-posterior morphogenetic gradient of the transcription factor Bicoid (*Bcd*) has been studied experimentally and computationally for many years [197]. *Bcd* specifies positional information for the downstream segmentation genes, affecting the fly body plan. More recently, a number of researchers have focused on the patterning dynamics of the underlying *Bcd* mRNA gradient, which is translated into *Bcd* protein. New, more accurate techniques for visualizing *Bcd* mRNA need to be combined with quantitative signal extraction techniques to reconstruct the *Bcd* mRNA distribution.

Here, we present a robust technique for quantifying gradients with a two-exponential model. This approach: 1) has natural, biologically relevant parameters; and 2) is invariant to linear transformations of the data which can arise due to variation in experimental conditions (e.g. microscope settings, non-specific background signal). This allows us to quantify *Bcd* mRNA gradient variability from embryo to embryo (important for studying the robustness of developmental regulatory networks); sort out atypical gradients; and classify embryos to the developmental stage by quantitative gradient parameters. Results of this section follow [17] and [21].

**Approach.** In whole embryo imaging, variability can arise during tissue fixation and staining with fluorophores, as well as from differences in microscope settings (gain and offset) between measurements of different batches of embryos on different days. Here, we discuss features of the data extraction which are insensitive to such experimental variation.

Our approach aims to create a model for basal and apical profiles (see Figure 7.10B) for *Bcd* gradients, estimate the model parameters and show that they can help to obtain biological results; in particular, to compare different ages in embryo development. We show an example of how data extracted and modelled by this technique can provide new biological insights into *Bcd* RNA gradient formation.

The novelty of the approach consists in consideration of the model parameters, which do not

depend on the linear transformation of the data and thereby on the non-specific background signal and the microscope settings. It is very important, since otherwise the comparison results can be caused by the experiment's conditions, not by the biology reasons.

**Model.** A two-exponential fit of a *Bcd* protein profile can be well approximated by a single exponential plus a nearly-constant background [198, 45]. In contrast, while some *Bcd* RNA profiles show such characteristics, many others, especially at early stages, show a much sharper exponential drop in the anterior, plus a constant or even posteriorly-rising component through the rest of the embryo (Figure 7.11). The transition between components can be readily visible in RNA patterns (and not in protein), as a ‘kink’ around the 20–30% egg length (%EL) position. These different components suggest multiple scales (or mechanisms) in the posterior-ward transport of *Bcd* RNA.

**Technique.** We previously applied a signal extraction technique based on Singular Spectrum Analysis (SSA) to quantify *Bcd* protein gradients [45]. This demonstrated that SSA could reliably and automatically extract AP *Bcd* protein gradients. These were the sum of two exponentials, one with a significant decay constant (strong curvature) and one of nearly linear form, capturing the non-specific background signal. Here, we adapt the SSA technique to the more complex cases of *Bcd* RNA gradients, validating the reliability and effectiveness of the approach. SSA itself is used for signal extraction, and the SSA-related method ESPRIT [108, 29] is used for the estimation of signal parameters. SSA techniques have proven to be robust to signal extraction from data with substantial experimental variability and intrinsic noise [1, 45, 199, 29].

### 7.3.1. Methods

#### Two-exponential modeling

**Description of the model** We fit the following two-exponential function (of AP distance,  $x$ ) to *Bcd* mRNA data, to capture the distinct two-component pattern of most *Bcd* RNA gradients (with the ‘kink’, commonly observed at 20–30 %EL):

$$s(x) = C_1 \exp(\alpha_1 x) + C_2 \exp(\alpha_2 x), \quad (7.6)$$

or

$$s(x) = C_1 \lambda_1^x + C_2 \lambda_2^x,$$

for  $\lambda_i = e^{\alpha_i}$ . The two components, where the first we call anterior (for the sharp, quickly decaying pattern in the anterior) and the second we call shallow (for the more constant component in the mid- and posterior embryo), each have two parameters — an amplitude  $C$ , and a rate  $\alpha$  (or an exponential base  $\lambda$ ). In biological terms,  $C$  in  $C\lambda^x$  for  $\lambda < 1$  represents the maximum concentration of the exponential component, and  $\alpha$  (and therefore  $\lambda$ ) represents the rate at which the component decreases (or increases) along the AP coordinate. The anterior exponential is always decreasing and therefore  $\lambda_1 < 1$ ,  $\alpha_1 < 0$ ; while the shallow exponential can be decreasing or increasing (Figure 7.11).

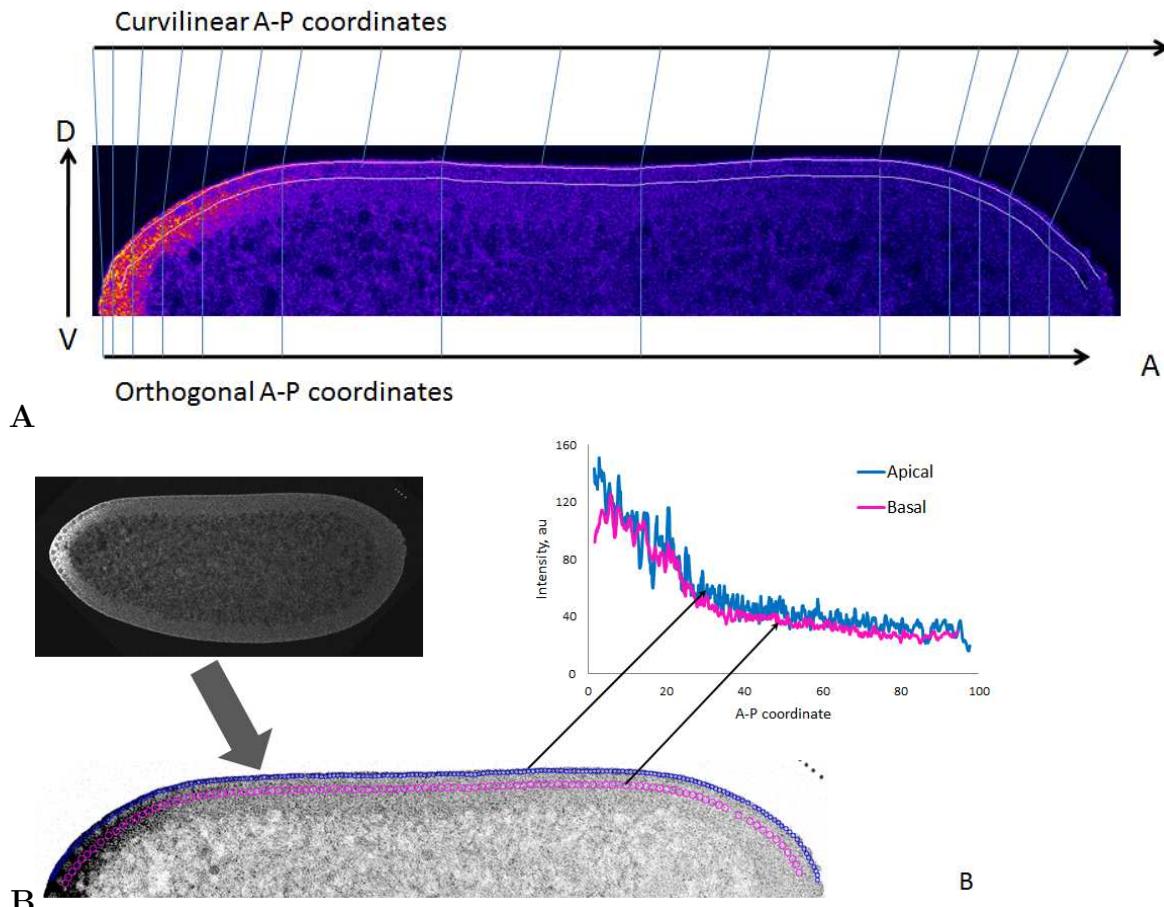


Figure 7.10: Preparation of data for quantitative analysis of sagittal images by 1D Singular Spectrum Analysis (SSA). **A.** Fluorescence intensity is proportional to the concentration of *Bcd* RNA. The gradient in *Bcd* mRNA is chiefly in the head-to-tail, AP, direction (left to right), but DV variation (top-to-bottom coordinate) can be seen, as well as variation by depth in the embryo (BA direction). For transport and patterning along the surface of the embryo, the natural coordinates are curvilinear. For extraction of the head-to-tail gradient patterning, the curvilinear coordinates are well approximated by a projection onto the AP axis (see text). **B.** For quantification of the AP gradient and BA differences, we sample data from an apical layer above the cortical nuclei and a basal layer below the cortical nuclei, using chains of overlapping regions of interest (ROIs). Data from each layer is analyzed independently with 1D SSA. Each layer can then be plotted as intensity vs. AP position (right insert).

One-exponential plus constant background is a special case of equation (7.6), with the rate  $\alpha_2 = 0$  (the base  $\lambda_2 = 1$ ). Use of (7.6) does not require two strong (nonzero  $\alpha$ ) exponentials in the signal (pattern). In the case of the model commonly applied to the *Bcd* protein gradient, the first exponential describes the signal and the second exponential describes the non-specific background signal and the offset of the microscope.

Note that raw image data is likely of the form  $s(x) + \epsilon(x)$ , where  $\epsilon(x)$  represents “noise”, i.e. non-regular oscillations with zero mean.

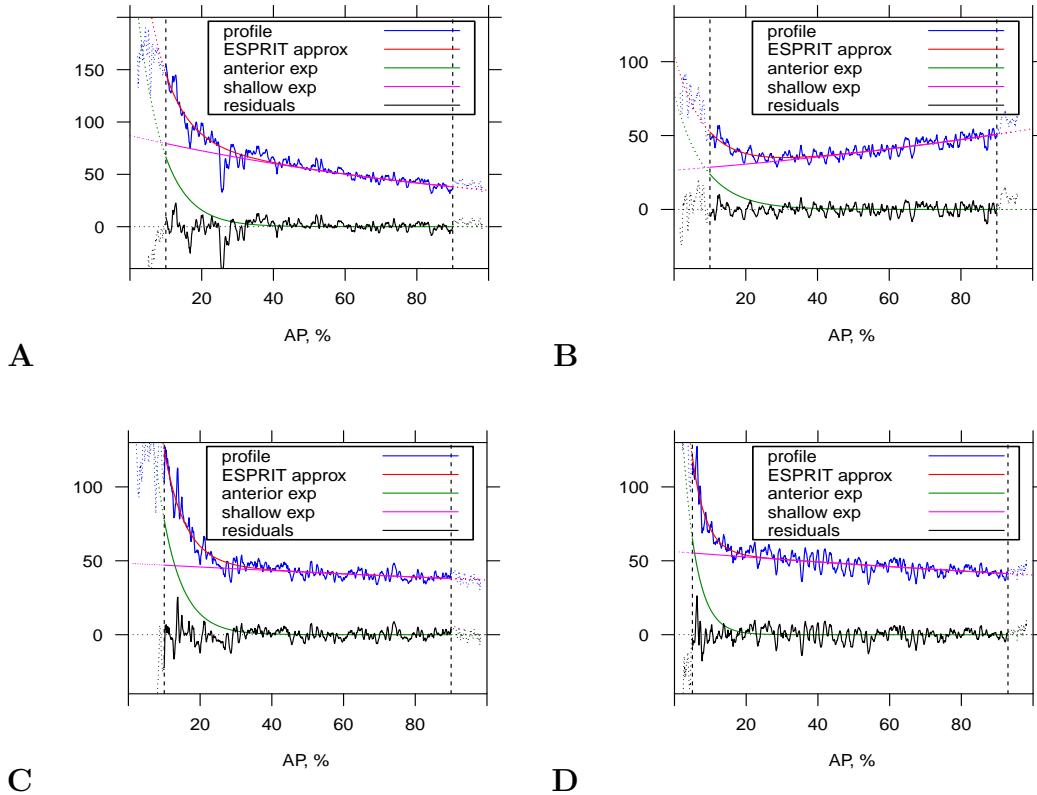


Figure 7.11: Representative examples of AP profiles of *Bcd* mRNA, illustrating the variety of cases and efficacy of the modeling approach. The raw data are noisy; the smooth pattern is the ESPRIT fit, which is the sum of two exponentials; they are depicted individually (decreasing to zero anterior and trunk shallow ones; the latter is similar to a straight line). The residual noise is oscillating around zero. **A.** An early nuclear cleavage cycle 14A (nc14) embryo with a typical broad anterior exponential and shallow 2nd component extending throughout the embryo (Cf [200]). **B.** A *Bcd* mRNA profile in which the 2nd, trunk, component rises towards the posterior (i.e. has a positive exponential rate). **C.** A case with a nearly flat 2nd component (representing the mRNA signal posterior of 25%EL). **D.** An embryo with a very sharp anterior (1st component) exponential, dropping to low values by 10%EL.

**Model characteristics independent of the microscopy gain/offset and background** To remove effects from variability in microscope settings (gain and offset) and the unknown form of non-specific background [198, 201, 202], gradient characteristics can be used which do not change under a linear transformation of the gradient.

That is if each profile (gradient) can be represented by the linear transformation:

$$f(x) = A(s(x) + \varepsilon(x)) + B, \quad (7.7)$$

where  $A > 0$  and  $B$  represent an unknown scaling and an unknown offset, respectively.  $A$  and  $B$  are likely to differ between embryos (with different staining conditions, microscope settings, etc.), but when we take basal and apical traces within each embryo image, we assume that the  $A$  and  $B$  are

constant within a single embryo. To compare data between embryos, we take advantage of the independence of the profile characteristics from linear transformations, i.e. independence from  $A$  and  $B$  values.

The signal (7.6) has characteristics which approximately satisfy independence of linear transformations if the second (shallow-gradient) exponential rate is small enough ( $\alpha_2 \approx 0$ ,  $\lambda_2 = e^{\alpha_2} \approx 1$ ) and can therefore be approximated by a linear function. This is a reasonable assumption for the *Bcd* mRNA data, giving

$$\begin{aligned} A(C_1 \exp(\alpha_1 x) + C_2 \exp(\alpha_2 x)) + B &\approx \\ AC_1 \exp(\alpha_1 x) + AC_2(1 + \alpha_2 x) + B &\approx \tilde{C}_1 \exp(\tilde{\alpha}_1 x) + \tilde{C}_2 \exp(\tilde{\alpha}_2 x), \end{aligned}$$

where

$$\tilde{C}_1 = AC_1, \quad \tilde{\alpha}_1 = \alpha_1, \quad \tilde{C}_2 = AC_2 + B, \quad \tilde{\alpha}_2 = AC_2 \alpha_2 / (AC_2 + B).$$

One can see that some parameters (e.g. the pre-exponential coefficients  $C_1$  and  $C_2$ ) depend on  $A$  and/or  $B$ , while the rate of the first exponential  $\alpha_1$  is not affected by linear transformations. Additionally, we assume that the coefficients before the first and second exponentials are positive (true, generally, for the *Bcd* RNA data) for  $A > 0$  and  $B$ , which appear in the experiments. Then, the sign of  $\alpha_2$  is also stable to the transformation and determines if the second (shallow) exponential is either increasing or decreasing. The negative sign of  $C_1$  or  $C_2$  can be used for the detection of profiles, which do not fit the model.

Note that similar considerations about parameters independent of linear transformations can be applied to any number of active exponentials (e.g., we can consider both anterior and tail ones and study their rates), while the shallow exponential should be alone. It is important that the noise level directly depends on the scaling  $A$  and therefore we do not use it as an embryo characteristic.

**Model characteristics of apical and basal profiles** We apply the model to both apical and basal profiles, adding corresponding upper indices to parameter notation. Several characteristics, which reflect the relation between two profiles of the same embryo (therefore,  $A$  and  $B$  are the same for both profiles), can be added. For clarity, we will write ‘anterior’ or ‘shallow’ in the lower indices instead of 1 and 2 correspondingly.

Thus, the following combinations of the model parameters of the profiles can be considered as almost independent of a linear transformation of the intensities, i.e., of  $A$  and  $B$ :

$$\alpha_{\text{anterior}}^{(\text{apical})}, \quad \text{sign}(\alpha_{\text{shallow}}^{(\text{apical})}), \quad C_{\text{shallow}}^{(\text{apical})} \alpha_{\text{shallow}}^{(\text{apical})} / C_{\text{anterior}}^{(\text{apical})}, \quad (7.8)$$

$$\alpha_{\text{anterior}}^{(\text{basal})}, \quad \text{sign}(\alpha_{\text{shallow}}^{(\text{basal})}), \quad C_{\text{shallow}}^{(\text{basal})} \alpha_{\text{shallow}}^{(\text{basal})} / C_{\text{anterior}}^{(\text{basal})}, \quad (7.9)$$

$$C_{\text{anterior}}^{(\text{apical})} / C_{\text{anterior}}^{(\text{basal})}, \quad C_{\text{shallow}}^{(\text{apical})} \alpha_{\text{shallow}}^{(\text{apical})} / (C_{\text{shallow}}^{(\text{basal})} \alpha_{\text{shallow}}^{(\text{basal})}), \quad (7.10)$$

where (7.8) are characteristics of apical profiles, (7.9) are characteristics of basal profiles, and characteristics (7.10) show relations between apical and basal profiles.

Hereinafter we use the following model characteristics based on these combinations:

- the anterior gradient rates  $\lambda_{\text{anterior}}^{(\text{apical})} = \exp(\alpha_{\text{anterior}}^{(\text{apical})})$  and  $\lambda_{\text{anterior}}^{(\text{basal})} = \exp(\alpha_{\text{anterior}}^{(\text{basal})})$ ;
- the logarithmic ratio between the anterior gradient pre-exponential coefficients for the apical and basal profiles  $C^{\text{ab}} = \ln(C_{\text{anterior}}^{(\text{apical})}/C_{\text{anterior}}^{(\text{basal})})$ ;
- indicators of non-increase in the shallow components  $\lambda_{\text{shallow}}^{(\text{apical})} \leq 1$  and  $\lambda_{\text{shallow}}^{(\text{basal})} \leq 1$ .

Note that these characteristics have sense if  $C_{\text{anterior}}^{(\text{apical})}$ ,  $C_{\text{anterior}}^{(\text{basal})}$ ,  $C_{\text{shallow}}^{(\text{apical})}$ , and  $C_{\text{shallow}}^{(\text{basal})}$  are positive. Constructing the model, we assumed that the anterior exponential vanishes within the embryo length. To check it, let us consider additional characteristics: the AP positions  $AP0_{\text{anterior}}^{(\text{apical})}$  and  $AP0_{\text{anterior}}^{(\text{basal})}$  at which the anterior exponentials become almost zero (more precisely, such percent  $x$  %EL that  $C\lambda^x < 1$ ; recall that the intensity is measured in the range 0–255). These characteristics can be affected by a scaling and therefore should be used carefully.

These relations underlie the quantitative conclusions in this work. We also use these relations to screen for atypical embryos and outliers, which aids in following the development of the *Bcd* RNA gradient over time and for studying apical-basal differences.

**Estimation of the two-exponential model parameters** We use the subspace-based method ESPRIT, motivated by the success of SSA (also a subspace-based method) in smoothing one-dimensional gene profiles from *Drosophila* embryos, see [199] and Section 7.2, as examples of smoothing by SSA. On profiles from different genes, the method proved to be robust to high noise and to variations in embryo characteristics.

The mathematical details of ESPRIT can be found in Section 1.9.2. We use the method to estimate the exponential decays in (7.6):  $\lambda_{\text{anterior}}^{(\text{apical})}$ ,  $\lambda_{\text{anterior}}^{(\text{basal})}$ ,  $\lambda_{\text{shallow}}^{(\text{apical})}$  and  $\lambda_{\text{shallow}}^{(\text{basal})}$ . The estimation of the coefficients  $C_{\text{anterior}}^{(\text{apical})}$ ,  $C_{\text{anterior}}^{(\text{basal})}$ ,  $C_{\text{shallow}}^{(\text{apical})}$  and  $C_{\text{shallow}}^{(\text{basal})}$  is then performed by the conventional least-squares method, since the model (7.6) is linear in pre-exponential coefficients.

Since the first exponential is expected to be rapidly decreasing ( $\lambda_{\text{anterior}}^{(\text{apical/basal})} < 1$ ) and the second exponential is expected to be close to constant ( $\lambda_{\text{shallow}}^{(\text{apical/basal})}$  near 1), we reorder the ESPRIT estimates of two exponential rates accordingly.

## Data

**FISH and data acquisition** For approbation of the suggested model, we consider the dataset of the confocal images of early *Drosophila* embryos stained by Fluorescent for *bcd* mRNA (the database FISH). The data were characterized in [200], and it is the largest such dataset up to date. Our dataset consists of images of about 160 embryos, ranging in stage from unfertilized eggs (not analyzed) to early nuclear cleavage cycle 14A (nc14, same dataset as in [200]). In the

current study, we analyzed 124 embryos. These were divided into three developmental stages, based on preliminary analysis and biological considerations: Cleavage, or pre-blastoderm (nc1–nc9); Syncytial Blastoderm (nc10–nc13); and Cellularizing Blastoderm (nc14A). The Cleavage stage is long, lasting about 80 min (at room temperature), and has highly variable *bcd* mRNA gradients. For more detailed analysis, we subdivided Cleavage into two sub-groups: Early (nc1–nc8) and Late (nc9). The Syncytial Blastoderm stage spans about 45 min, and this could be subdivided into two sub-groups: nc10–nc12 and nc13. The last stage, early nc14A, is short (15–20 min.), but highly variable and dynamic. Careful visual inspection allowed us to divide the nc14A embryos into three sub-groups: early, mid and late.

**Construction of 1D profiles** Raw data from the confocal microscope consists of mRNA intensities per a small circular area with 2D spatial coordinates. After selecting the regions of interest (ROI chains), two techniques were tested for converting the data into 1D AP profiles. The first (and simplest) technique projects intensities onto an AP axis orthogonal to the DV axis by discarding the DV component of the coordinate (Figure 7.10A). This has been used by many groups, see for example [203, 198]. The second technique preserves the natural curvilinear coordinates of the embryo, with the distance between ROIs calculated by  $d^2(i) = (\text{AP}(i+1) - \text{AP}(i))^2 + (\text{DV}(i+1) - \text{DV}(i))^2$ . Cumulative distances are then normalized by dividing by the sum of  $d(i)$ .

Regardless of the technique (AP or curvilinear coordinates), the 1D coordinates obtained are not equidistant. Linear interpolation was used to create equidistant points of a given spatial step. Step 0.08–0.1%EL was chosen to generate approximately equal numbers of points for the two techniques. These results obtained employing AP coordinates appear to be more precise than that obtained by curvilinear coordinates. Therefore, here we can consider only AP coordinates.

### 7.3.2. Results and discussion

#### Model application

Figure 7.11 demonstrates a set of examples to illustrate the variety of profiles which can be fit by the two-exponential model. These include the typical profiles considered in [200], with a rapidly decreasing anterior gradient and slowly decreasing gradient to the posterior (Figure 7.11A), and profiles with increasing (Figure 7.11B) or flat (Figure 7.11C) posterior gradients.

Data is generally too biased and noisy from the terminal regions of the embryo: 0–10 %EL and 90–100 %EL (Cf [198], see Figure 7.11). Processing and analyzing data from 10–90 %EL is sufficient for extracting *Bcd* RNA profiles from nearly all embryos older than nc6. For very early embryos (CleavageEarly stage), gradients have just begun to form from initial terminal locales; in these cases, it may be more appropriate to process the data from 5 %EL (Figure 7.11D). For uniformity, we will process all the data on 10–90 %EL. Figure 7.11 shows that the two-exponential model suits different types of data very well.

Typically, embryos have a decreasing anterior exponential component and decreasing or close-to-constant shallow posterior gradients (both for the apical and basal profiles). We call these Type 1 (typical) embryos. Some embryos, however, show a posteriorly-increasing shallow gradient for either apical or basal profiles. We call these Type 2 (atypical) embryos. Type 2 profiles are common early in development (Cleavage) and uncommon in later stages. Here, we focus on Type 1 embryos, which represent the majority of the dataset.

Detection of Type 1 can be performed by means of exponential rates of the shallow exponents: (A)  $\lambda_{\text{shallow}}^{(\text{apical})} < 1.002$ ,  $\lambda_{\text{shallow}}^{(\text{basal})} < 1.002$ . This condition screens for shallow profiles (both basal and apical) which do not increase towards the posterior (1.002 is used for 1, to account for estimation errors).

## Model validation

Even within one developmental stage, the shape of mRNA profiles from embryo to embryo is highly variable. This makes the construction of a prototype profile challenging and complicates the understanding of the underlying biological mechanisms. Fortunately, the variability is mostly due to a minority of embryos, and these can be detected using the two-exponential model. Removal of such embryos reduces the variability significantly.

We remove embryos, which were not described by the model with reasonable parameters. First, the pre-exponential coefficients for anterior and shallow exponentials should be positive. Therefore, we assume that (B)  $C_{\text{anterior}}^{(\text{apical})} > 0$ ,  $C_{\text{anterior}}^{(\text{basal})} > 0$ ,  $C_{\text{shallow}}^{(\text{apical})} > 0$  and  $C_{\text{shallow}}^{(\text{basal})} > 0$ . Then, for regular embryos, the anterior exponentials should vanish within the range 0–100 %EL. Therefore, we impose the condition (C)  $AP0_{\text{anterior}}^{(\text{apical})} \leq 100$ ,  $AP0_{\text{anterior}}^{(\text{basal})} \leq 100$ . Note that (A), (B), (C) are robust to small changes in constraint thresholds (results not shown).

92 embryos satisfy conditions (A)–(C), from the complete dataset of 124 embryos; the analysis in the rest of this study is on these 92 embryos. For these 92 embryos, it was checked that the systematic errors in the model are negligible relative to the residual noise or to the profile itself. Thus, we conclude that the profiles suit the considered model with sufficient accuracy.

## Model efficacy for finding trends in developmental biology

The parameters from the two-exponential fits are quite variable, both within and between developmental stages (Figure 7.12A), as expected from the observed variability in profiles.

Though the large variability and small sample size do not allow for statistically significant conclusions for all comparisons, several observations can be made. In particular, CleavageEarly has the largest average anterior exponential decay constant of any developmental stage (i.e. the steepest profile). This difference is statistically significant (t-test), but could be rendered insignificant by moderate changes in just one of the six embryos. We therefore combine groups to obtain 3 age groups (from 7) with larger sample sizes: (1) Cleavage,  $n = 20$ ; (2) nc10–nc13,  $n = 25$ ; (3) nc14,  $n = 47$ . Figure 7.12B shows that these larger groups have more distinct clustering, with distinct

means.

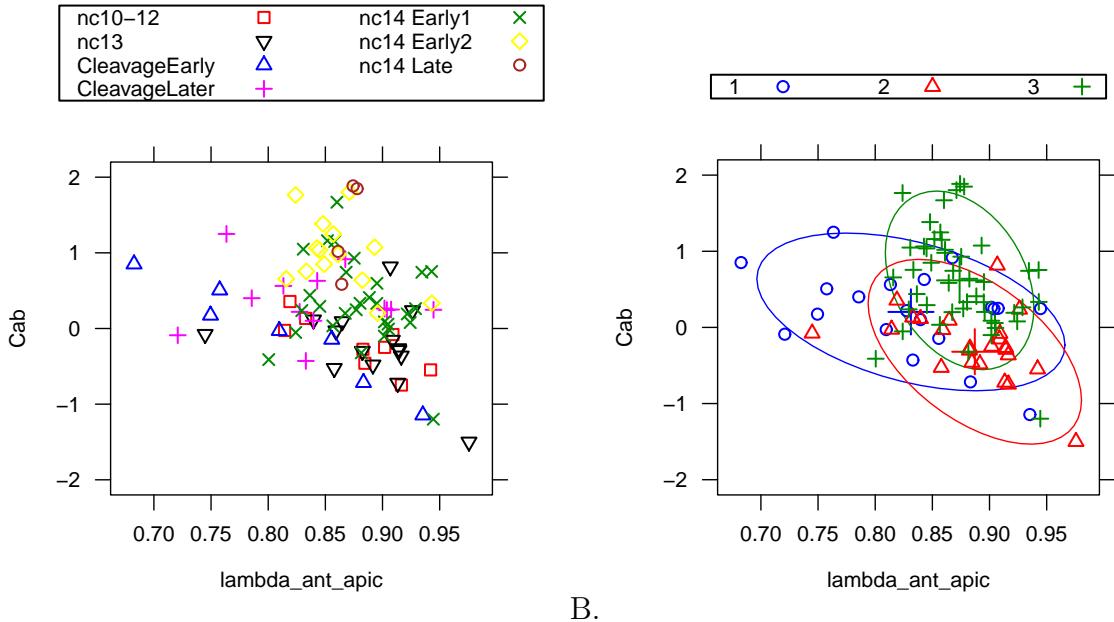


Figure 7.12: Pre-exponential factor  $C^{ab}$  vs. anterior gradient  $\lambda_{\text{anterior}}^{(\text{apical})}$ . **A.** 7 (marked) developmental stages (Section 7.3.1): Two-exponential parameters show large variability between and within the developmental stages. **B.** 3 combined groups (see key): difference in parameter values, with 80% dispersion ellipsoid.

Table 7.7 shows the average values for  $\lambda_{\text{anterior}}^{(\text{apical})}$  and  $C^{ab}$  with their 90% confidence intervals.

Table 7.7: Combined groups: means and 90% confidence intervals for main characteristics of apical and basal profiles for 3 groups.

	$\lambda_{\text{anterior}}^{(\text{apical})}$	lower bound	upper bound
Cleavage	0.831	0.803	0.859
nc10-13	0.887	0.870	0.904
nc14	0.874	0.865	0.883
	$C^{ab}$	lower bound	upper bound
Cleavage	0.203	-0.010	0.417
nc10-13	-0.320	0.543	-0.096
nc14	0.618	0.461	0.775

One-way ANOVA (both parametric and non-parametric (Kruskal-Wallis)) confirms that both  $\lambda_{\text{anterior}}^{(\text{apical})}$  and  $C^{ab}$  significantly differ between the groups at the 5% level. Post-hoc comparisons show that  $C^{ab}$  (the logarithm of the ratio between the apical and basal anterior gradients at 10 %EL) is significantly different between all three groups; while the exponential decay rate of the anterior gradient is significantly larger ( $\lambda$  is smaller) only for the Cleavage group.

**Potentials of the approach** In Section 7.3.2, we screened embryos into Type 1 using condition (A), i.e. non posteriorly-increasing profiles. We can apply the suggested approach to embryos of Type 2 with posteriorly-increasing profiles (see Figure 7.11B). Moreover, the model can be extended to three exponentials (see Figure 7.13). With the extension of SSA to fit a 3-exponential model, these sorts of patterns can be readily analyzed by the current approach, broadening the use of the technique to allow for the comparison of patterns from different genes (e.g., consider Stau protein [200], which has a sharp rise in the vicinity of the posterior pole).

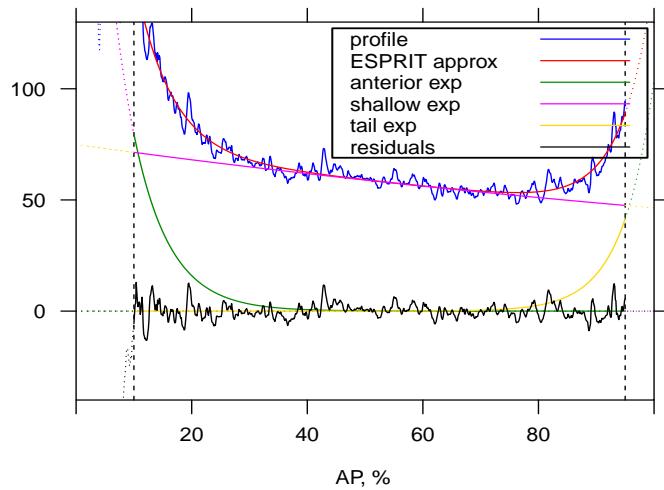


Figure 7.13: AP profile of the Stau protein (cf. [200, Fig.6]). The raw data are noisy; the pattern is obtained by the 3-exponential model: anterior exponential 1, decreasing to zero from the left; shallow exponential 2, which is similar to a straight line; posterior exponential 3, increasing from zero to the right; The residual noise is oscillating around zero.

The approach presented here is likely to be an effective tool for quantifying other spatial gradients in developmental biology, which could aid in revealing new features in the patterning dynamics and regulation of critical developmental events, especially where there are large dynamic changes and high variability — i.e. in cases where it is difficult to construct a reference or prototype profile.

## 7.4. Shaped Singular Spectrum Analysis for Quantifying Gene Expression, with Application to the Early *Drosophila* Embryo

In recent years, with the development of automated microscopy technologies, the volume and complexity of image data on gene expression have increased tremendously. The only way to analyze quantitatively and comprehensively such biological data is by developing and applying new sophisticated mathematical approaches. Here, we present extensions of 2D singular spectrum analysis (2D-SSA) for application to 2D and 3D datasets of embryo images. These extensions,

circular and shaped 2D-SSA, are applied to gene expression in the nuclear layer just under the surface of the *Drosophila* (fruit fly) embryo. We consider the commonly used cylindrical projection of the ellipsoidal *Drosophila* embryo. We demonstrate how circular and shaped versions of 2D-SSA help to decompose expression data into identifiable components (such as trend and noise), as well as to separate signals from different genes. Detection and improvement of under- and over-correction in multi-channel imaging are addressed, as well as the extraction and analysis of 3D features in 3D gene expression patterns.

In this publication, we introduce new computational tools to analyze gene patterning for three spatial dimension datasets, applied to early *Drosophila* embryos. These tools are an extension of two-dimensional singular spectrum analysis (2D-SSA).

Singular spectrum analysis is chosen, since it does not need a noise model to be given a priori. We decompose the data series into a set of elementary series; analyze them; choose appropriate components; and finally, sum the identifiable components together in classes. As an example, the selection of smooth components can produce adaptive smoothing. SSA is very useful for exploratory analysis since the method can deal with modulated noise, i.e. noise that can depend on trend values (e.g. has a multiplicative nature).

The decomposition of images is more complicated compared to time series analysis due to the variability of 2D patterns. But methods which are easily controlled and adaptive, such as 2D-SSA, can have broad applicability. 2D-SSA and related subspace-based methods are applied in texture analysis [178], seismology [179], spatial gene expression data [6], and medical imaging [204]. The paper [43] applied 2D-SSA to the analysis of digital terrains in geology and demonstrated that 2D-SSA is a useful tool for analyzing different levels of details in surface data. Later, 2D-SSA was applied to gene expression data to separate nuclear noise from expression trend [6].

In this section, we demonstrate how Shaped 2D-SSA can be applied to analyzing gene expression data [10]. The new approaches described here, circular and shaped 2D-SSA, are particularly applicable to cylindrical surfaces (as used for *Drosophila* embryos), to avoid edge effects, and patterns of non-rectangular shape. For example, the area of good-quality data in an image (e.g., without over-saturation) can be non-rectangular and even have gaps. Also, since the planar projection of a *Drosophila* embryo is nearly elliptical, the ability to analyze non-rectangular shapes can be useful.

#### 7.4.1. Materials

Data are taken from the Berkeley Drosophila Transcription Network Project (BDTNP) [205], which contains three-dimensional (3D) measurements of relative mRNA concentration for 95 genes in early development (including *snail* (*sna*)), and the protein expression patterns for four genes (bicoid, giant, hunchback (*hb*), and Krüppel (*Kr*)) during nuclear cleavage cycles 13 (C13) and 14 (C14A). BDTNP Release 2 contains individual datasets (PointCloud files) for 2830 embryos (<http://bdtnp.lbl.gov/Fly-Net/bioimaging.jsp>). These data were registered

to the coordinates of 6078 nuclei on the embryo cortex and presented as an integrated dataset (VirtualEmbryo file, with tools for visualization and analysis). Embryos were fixed and fluorescently stained to label the mRNA expression patterns of two genes plus nuclear DNA. One of the genes stained was either *even skipped* (*eve*) or *fushi tarazu* (*ftz*), which were used as fiduciary markers for subsequent spatial registration.

#### 7.4.2. Choice of parameters, separability and component identification

The approach to the choice of window size for one-dimensional time series is thoroughly described in [1], see also Section 3.4. Recommendations for 2D-objects are more complicated. For extraction of so-called objects of finite rank (sums of products of polynomials, exponentials and sinusoids), which satisfy linear recurrence relations (LRRs), windows should be large, up to half of the object size. However, real-world patterns usually have complex forms and satisfy LRRs only approximately and locally. The window needs to agree with this local character. In particular, sine waves are exactly governed by an LRR. However, if a 2D-sine wave has a slowly-changing location, then only its local parts satisfy an LRR. The window sizes need to be in accordance with the scale of this locality. The choice of window size is always a balance between the local and the global scales of the data.

Generally, SSA can separate smooth patterns from noise for a wide variety of patterns. For regular patterns, 2D-SSA can be applied whether the pattern varies smoothly or sharply. However, if the pattern is not regular, variation needs to be smooth to use 2D-SSA for signal separation. Irregular pattern with sharp variation is poorly separated by 2D-SSA. If, however, the sharp change occurs in a narrow area, this can be cut out, and the remaining data analyzed by Shaped SSA, which is a version of 2D-SSA with a non-rectangular shape of the image or the window.

The elementary components are grouped based on their similarity to the data components being extracted. For regular components like sine waves, the number of elementary components can be calculated from theory. Also, patterns usually have a limited frequency range (usually lacking high frequencies). In general, therefore, the leading elementary components with the appropriate frequency characteristics are ascribed to a pattern.

In this study we show how 2D-SSA can be used to remove noise, to separate regular oscillations from slowly-varying patterns (for correcting erroneous unmixing procedures), and to extract stripes for their further analysis. Shaped SSA allows for the analysis of complex patterns by splitting images into several parts.

*Drosophila* early gene expression (before the mid-blastula transition) produces smooth and simple patterns suitable for 2D-SSA processing. A number of web resources have such datasets (BDTNP BID, Fly-FISH, FlyEx among others). Shaped SSA can also be useful for a common subset of this data, in which patterns fall sharply to zero. In these cases, sub-regions can be excised or analyzed separately from the whole image. The gene *sna* is a typical *Drosophila* example seen in the BDTNP BID; such compact patterns are also seen in other experimental organisms, such

as the nine Zebrafish genes [206]. We expect 2D-SSA and Shaped SSA to therefore have broad applicability to image processing in developmental biology.

The problem of unmixing expression patterns from two different genes in one image [207] requires additional conditions. Specifically, information is needed on the unmixed expression of each gene (i.e. data from one gene in the absence of the other gene). If the two genes have slowly varying patterns, they cannot readily be separated by SSA. In such cases, SSA cannot be used to detect or correct errors in mixed images. However, SSA is an effective unmixing method for cases in which one gene has an approximately regular structure, and this differs from the structure of the other gene. In this work, we apply SSA to signal unmixing and image correction for such cases from *Drosophila* data.

## Data preprocessing

Initially, the data for 2D-SSA analysis should be measured on a regular grid. Data for gene expression are measured at nuclei, which are not regularly located on a 3D surface of the embryo (which is roughly ellipsoidal). The first step of preprocessing is a cylindrical projection of the data (centred on the major axis of the ellipsoid; the major axis of the embryo is found by Principal Component Analysis). We then interpolate the data to a regular grid on this cylinder. We analyze a central region of the cylinder, in order to avoid corruptions near the poles from the ellipsoid to cylinder transformation. After 2D-SSA decomposition, we interpolated the data back onto the nuclear centers. This interpolation is performed for smooth components; residuals are calculated as the difference between the initial data and interpolated smooth components.

Interpolation involves Delaunay triangulation followed by linear interpolation of nuclear centers to the triangulation.

## Implementation

The algorithms are implemented in the RSSA and BIOSSA packages in R. The R-package BIOSSA is an addition to RSSA for application to fly embryo gene expressions data and is briefly described at <http://biossa.github.io/>.

### 7.4.3. Periodic patterns produced by unmixing algorithms

Different emission spectra for fluorescent probes allow for the simultaneous staining of 3–4 gene products in embryonic tissues. Quantitative imaging projects [203, 205] use the same gene in one of these channels in all embryos, for reliable quantitative comparisons, registration, etc. The gene used for this marking in *Drosophila* embryos is commonly one of the pair-rule genes (such as *eve* or *ftz*), which have a characteristic periodic 7-stripe expression pattern.

Multi-channel imaging suffers from an inherent problem of overlapping emission spectra (when the fluorescent markers are simultaneously excited), where light from more than one fluorescent

dye is collected by a given acquisition channel. To computationally reduce this ‘cross-talk’, an automated channel unmixing method was developed and applied to the BDTNP data.

The problem with this approach in large-scale projects with automatic data processing is that the unmixing parameters can end up being too high or too low. If the parameters are overestimated, unmixing produces an overcorrection, which manifests as a partial subtraction of the common, reference pattern from the pattern of the second gene (the gene under study for the embryo). With periodic reference patterns (*eve*, *ftz*), this produces periodic grooves in the ‘unmixed’ pattern. Fig. 7.14 shows the effects of such overcorrection in one of the BDTNP embryos.

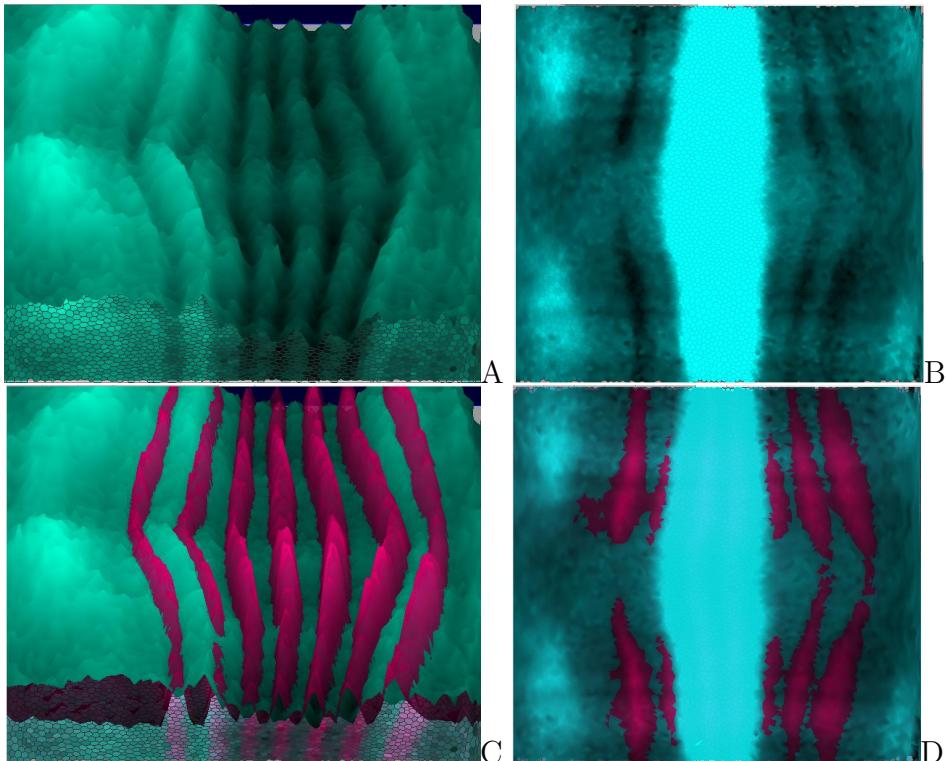


Figure 7.14: An example of over-correction in gene expression data causing the subtraction of the reference gene pattern (the seven-striped *ftz* and *eve* patterns; dark magenta) from the pattern under study (*hb* and *Kr* gene products (transcription factors); light blue). Visualization by PointCloudXplore tools, BDTNP embryos *hb* ‘v5-s11512-2oc06-25’ (A and C), *Kr* ‘v5-s12169-24oc07-22’ (B and D); (C) is the same as (A) with added *ftz*; (D) is the same as (B) with added *eve*

On the other hand, if the unmixing parameters are underestimated, unmixing produces an undercorrection, which can be seen as an addition of the common, reference pattern to the pattern of the second gene (that one being studied in the given embryo). Fig. 7.15 shows an example of undercorrection on a BDTNP embryo.

Mis-estimation of the unmixing parameters can be seen to introduce periodicity in a number of BDTNP embryos from the 7-stripe *eve* or *ftz* reference patterns. The effect is strong enough to be seen in some images integrated from multiple embryos (such as Fig. 7.15).

We now show how decomposition by circular 2D-SSA can be used to estimate and eliminate

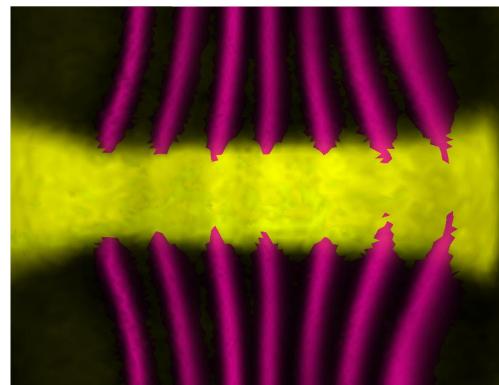


Figure 7.15: An example of under-correction, in which the periodic reference gene pattern (*eve*; dark magenta) adds periodicity to the non-periodic pattern under study (*sna* gene product; yellow). Visualization by PointCloudXplore. Embryo ‘v5-s10531-28fe05-07’.

the periodic components caused by under- or over-correction, using the examples of the BDTNP images in Fig. 7.14 and 7.15.

### Circular 2D-SSA, *hb* corrupted by *ftz*, strong over-correction

Fig. 7.16 shows the original images for *hb* and *ftz* expressions from a BDTNP embryo (ID ‘v5-s11512-2oc06-25’). The natural *hb* trend is low frequency; the natural pattern of *ftz* is high frequency; cross-talk, with over-correction in the unmixing algorithm, ‘bleeds’ the high frequency *ftz* pattern into the *hb* pattern. These images are ‘unrolled’ from the cylindrical projection of the data; therefore, the top and bottom edges connect (periodic boundary conditions).

We pre-process the images by interpolating them to a regular grid (step 0.5%) and removing 20% from the left and 5% from the right (to focus on the stripe region). The use of circular 2D-SSA allows us to analyze the cylindrical dataset. We use a rectangular window of  $25 \times 10$ . In consideration of the regular oscillations along the anteroposterior (AP, horizontal) coordinate, the first window dimension, 25, is larger than the second dimension, 10.

Fig. 7.17 presents 2D-SSA decomposition into elementary image components for *hb*; Fig. 7.18 shows this for *ftz* (we depict the 26 largest components; the smaller components were not found to be significant in image reconstruction). Fig. 7.17 contains several components with vertical stripes caused by or influenced by the *ftz* channel. If one compares elementary components of the *ftz* decomposition (Fig. 7.18, striped components 2–5, 9–11, 15–17) with the *hb* decomposition (Fig. 7.17), it appears that *hb* components 1–4 are likely due to expression pattern, while components 5–9,11 and probably 10,12 are due to *ftz*-correction.

Fig. 7.19 shows reconstructions from the leading high-frequency components for each image, components 5 and 6 from Fig. 7.17, components 2 and 3 from Fig. 7.18. The reconstructions are very similar, but have opposite phases, indicating that the *hb* data was overcorrected. Fig. 7.20 is reconstructed from all striped components for each image; again, the patterns are very similar but of opposite phases.

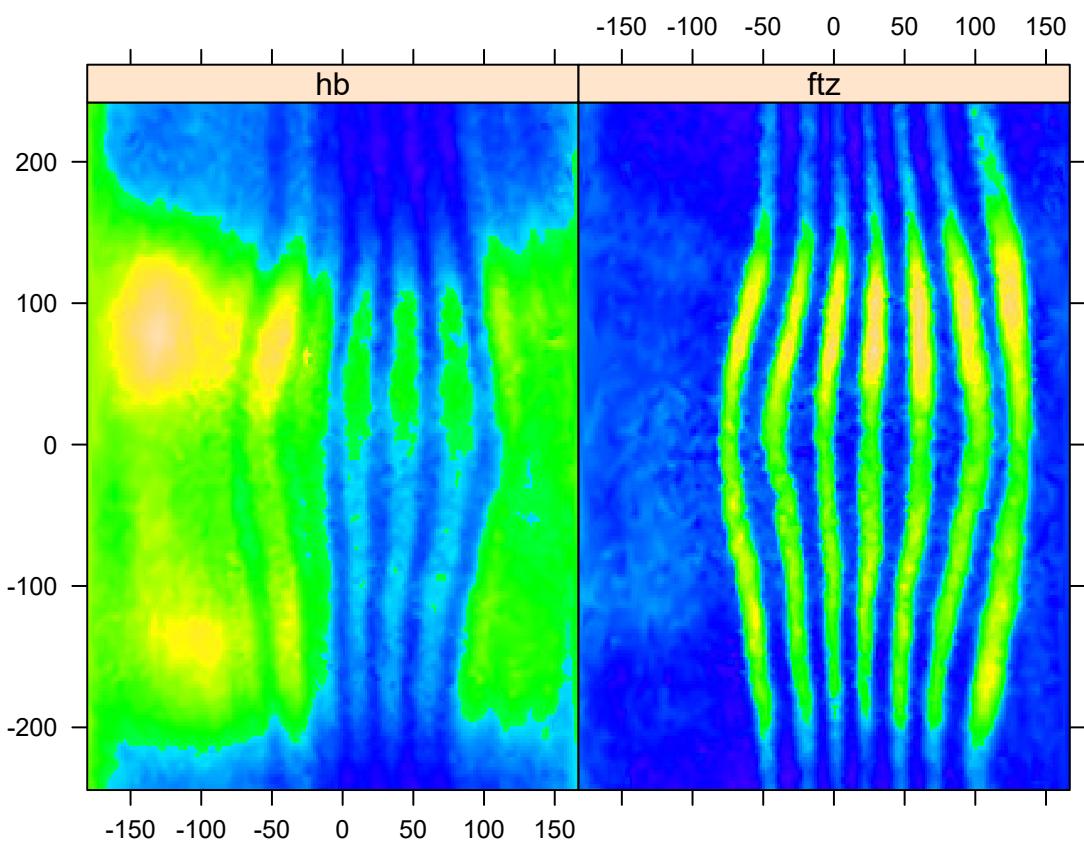


Figure 7.16: *hb* and *ftz*: original images of the 'unrolled' cylindrical surface; the top values are a direct continuation of the bottom values.

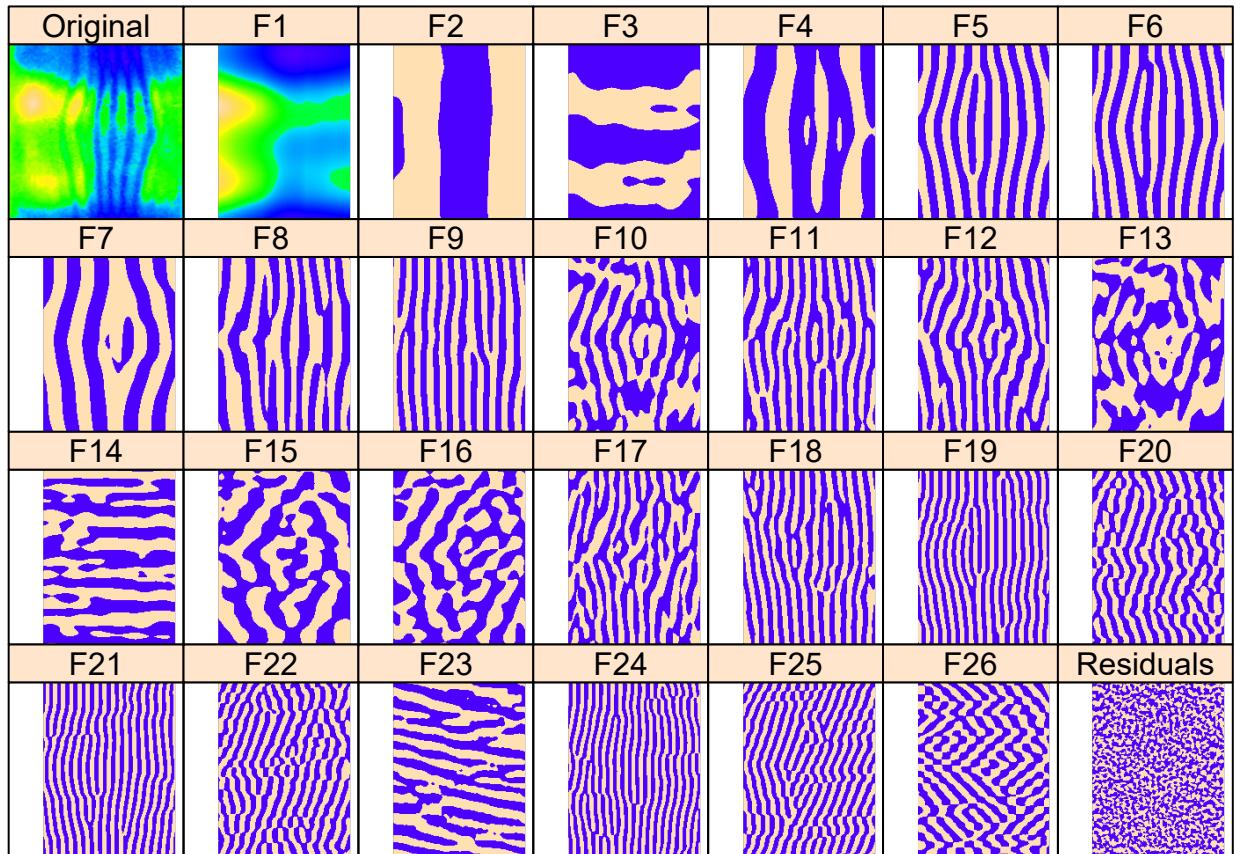


Figure 7.17: *hb*: the original image and the elementary components extracted by circular 2D-SSA. The original image and the leading component (F1) are colour-mapped according to the min and max expression levels. For more contrast, the remaining components are depicted in a binary format, with positive values in beige and negative values in purple.

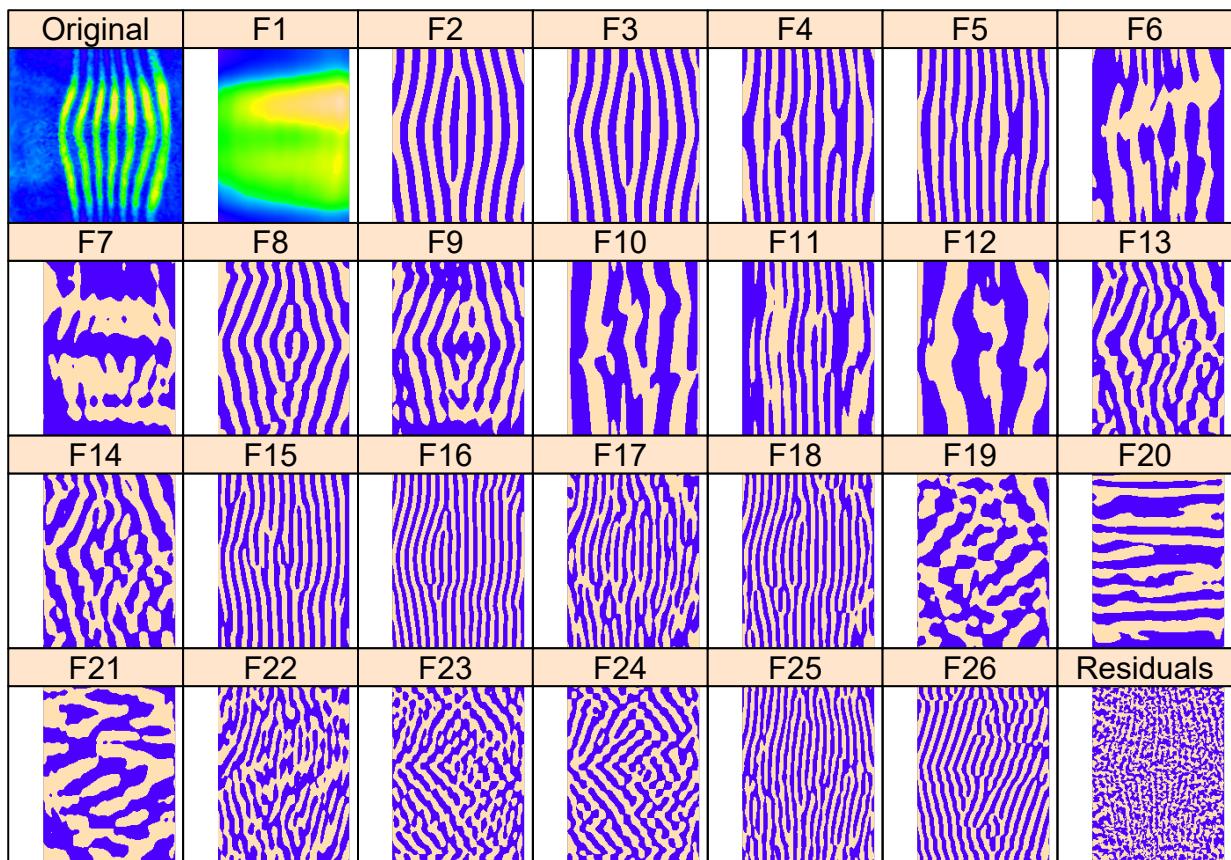


Figure 7.18:  $ftz$ : the original image and F1 with the background; the remaining elementary components are depicted in a binary format.

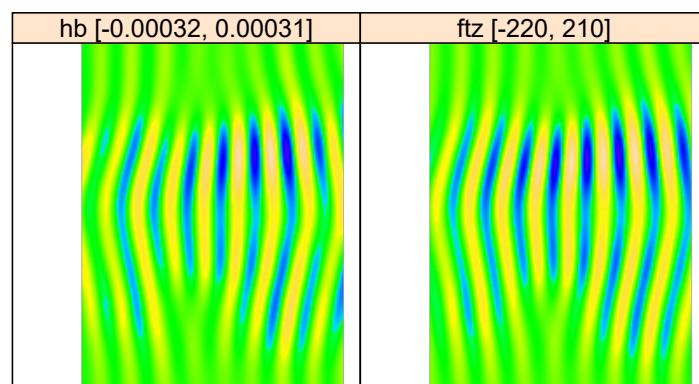


Figure 7.19:  $hb$  (left) and  $ftz$  (right): reconstruction from the main striped components 5 and 6 for the  $hb$  analysis, 2 and 3 for the  $ftz$  analysis. The stripes are out of phase for  $hb$  and  $ftz$ .

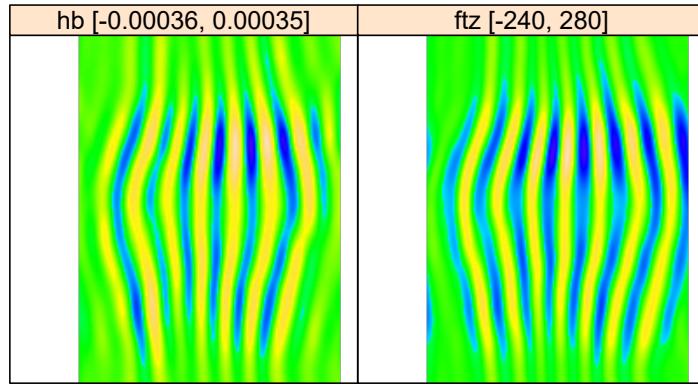


Figure 7.20:  $hb$  (left) and  $ftz$  (right): reconstruction from all striped components.

Simultaneous with removing stripes, this process also decomposes an image into pattern and noise (residuals): Fig. 7.21 shows the reconstruction of  $hb$  expression from the ‘un-striped’ components 1–4, alongside the striped components (strongly affected by  $ftz$ ) 5–12, and the residuals. Circular 2D-SSA provides a method for removing under- or over-correction in the unmixing algorithm, and therefore for clearing gene patterns from cross-talk effects. For an image without stripes, 2D-SSA produces a direct decomposition into pattern and noise. We showed here that SSA decomposition is robust for data with cross-talk stripes.

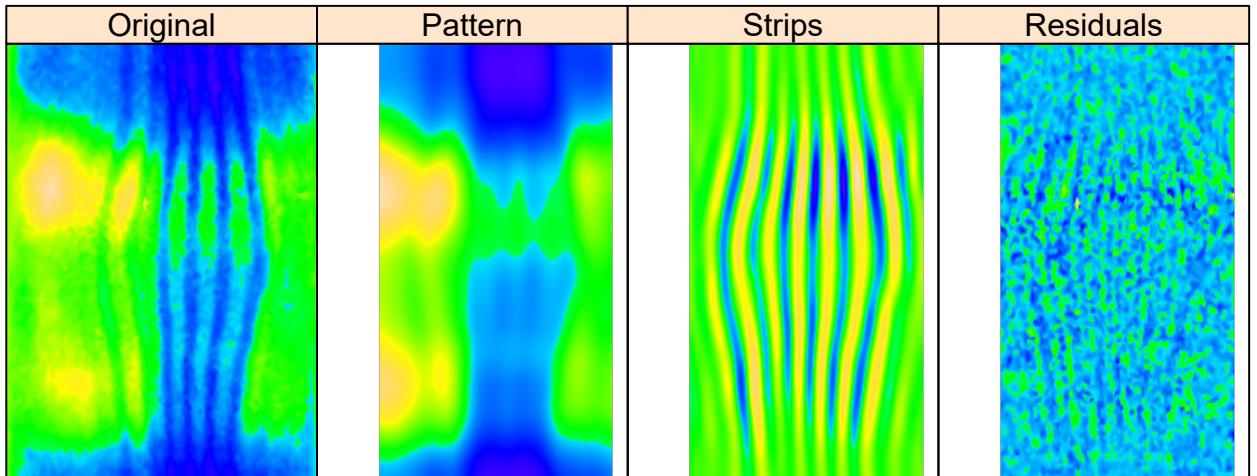


Figure 7.21:  $hb$ , left to right: original image, un-striped pattern, stripes, and residual noise.

### Shaped 2D-SSA, $sna$ corrupted by $eve$ , under-correction

A number of genes express in patterns which are more complex than the general AP variation seen with gap genes such as  $hb$  and  $Kr$ . To analyze cross-talk for such data, we introduce the shaped version of 2D-SSA. As an example, *snail* ( $sna$ ) is expressed in a broad band along the

ventral midline of the embryo (Fig. 7.24, v5-s10531-28fe05-07, cy3\_apical)). Since *sna* shows a very sharp transition from expressing to non-expressing regions, we analyzed these separately (Fig. 7.22, expressing; Fig. 7.23, non-expressing). Analysis was conducted on a regular grid (step 0.5%), clipped 15% from left and right. For the central expressing zone (Fig. 7.22), we used a window of  $40 \times 10$ ; for the lateral non-expressing zone (Fig. 7.23), we used a window of  $30 \times 10$ .

The decomposition shows that the elementary components  $\{3,4\}$  (Fig. 7.22) and  $\{4,5,16,17\}$  (Fig. 7.23) correspond to stripes, which come from the *eve* reference marker. Figs. 7.22 and 7.23 show these stripe components, and the effect of removing these stripes to reveal the *sna* signal. Fig. 7.24 shows this for the complete *sna* image (combination of the expressing and non-expressing zones). In this case, the stripe components from the *sna* image and from the *eve* marker image are in phase, indicating that this is a case of under-correction in the unmixing algorithm (see Fig. 7.25, where the original images and the stripe reconstructions are put together).

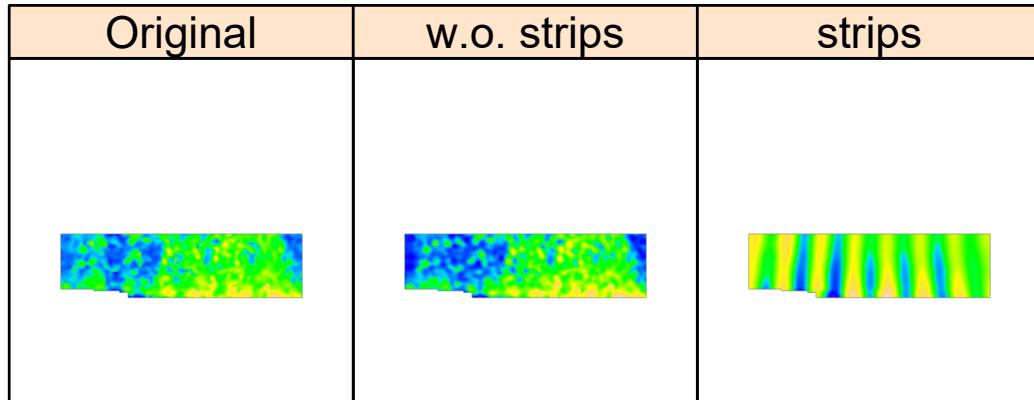


Figure 7.22: *sna* image, area 1, strong expression zone. Left to right: original image, reconstruction without stripes, and stripe components from the *eve* marker.

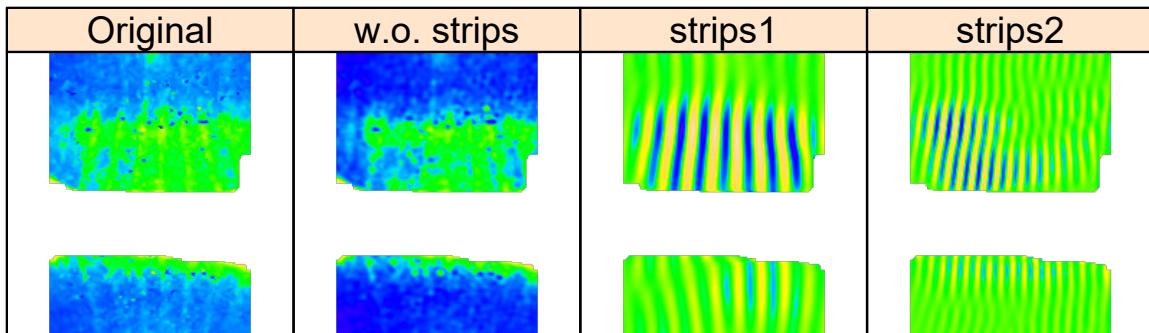


Figure 7.23: *sna* image, area 2, weak expression zone. Left to right: original image, reconstruction without stripes, and stripe components.

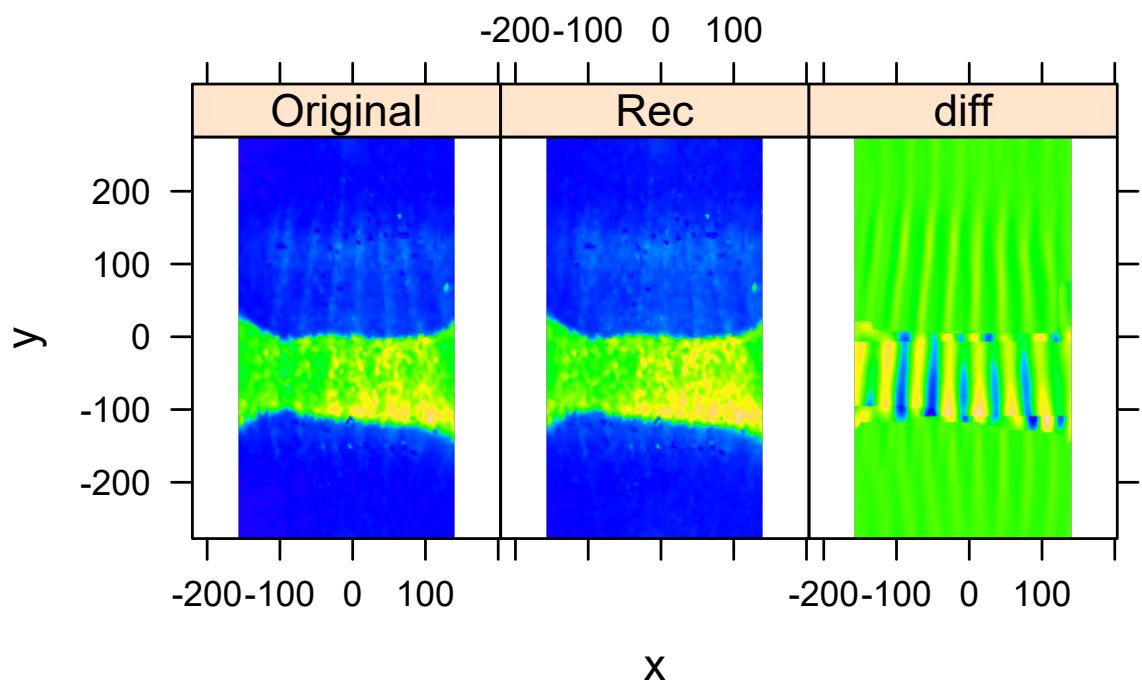


Figure 7.24: *sna*, combined image (both zones from Figs. 7.22 and 7.23). Left to right: original image, reconstruction without stripes, and the difference. BDTNP embryo v5-s10531-28fe05-07.pce

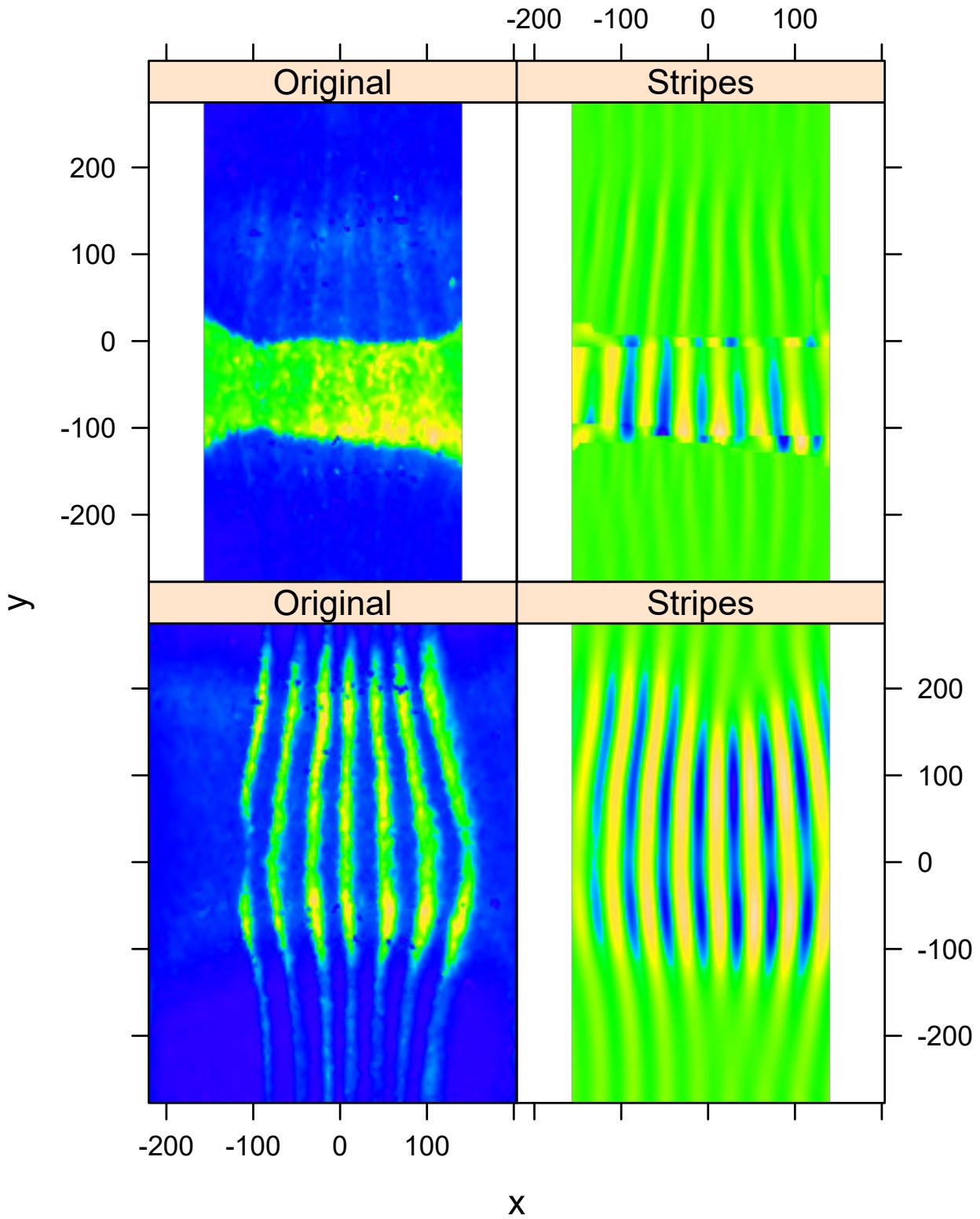


Figure 7.25: *sna* and *eve*: the original images (left) and the stripes (right), *sna* at the top and *eve* at the bottom.

Thus, we have constructed a procedure for removing under- or over-corrections. Note that if an image does not contain stripes, images of elementary components also will not contain stripes and therefore can see if correction is necessary.

#### 7.4.4. Conclusions

This study has shown the applicability of Shaped and Circular extensions of 2D-SSA to analyzing embryo images from a quantitative high-throughput project in developmental biology. We have shown that 2D-SSA can decompose images and classify components according to the gene of interest. This is an effective means for reducing the ‘cross-talk’ between gene channels which arises in the imaging technique but can be amplified by the automated post-processing unmixing algorithm.

Circular 2D-SSA is a critical extension for analyzing cylindrical data projections (accounting for periodic boundaries in ‘rectangular’ images). Shaped 2D-SSA allows for the analysis of sub-regions of the image, important for analyzing complex expression patterns and geometries, and avoiding edge effects.

The procedure is performed under user control and can be adapted to an image’s unique structure with a flexible choice of window shapes and sizes.

We have demonstrated that 2D-SSA can be used to extract signal and noise from images with both strong and weak over- or under-correction of cross-talk. This is a significant tool for separating gene expression in multi-channel images, and for extracting residual noise for studying the stochastic aspects of gene expression. In particular, we have used SSA to separate low-frequency genes of interest (the gap genes *hb* and *Kr*, and *sna*) from ‘bleed-through’ cross-talk of the high-frequency pair-rule fiduciary markers (*eve* and *ftz*). In addition, we have shown how SSA components can be used to quantify *eve* stripes (in particular stripe 4) that reveal new types of variability in expression, leading to new insights into developmental mechanisms. These are all examples of how 2D-SSA can be applied — we expect them to be broadly generalizable to other cases of multi-channel 3D data from *Drosophila* and other organisms.

### 7.5. Shaped 3D Singular Spectrum Analysis for Quantifying Gene Expression, with Application to the Early Zebrafish Embryo

In this section, we will focus on the analysis of three-dimensional data. In addition to the data being noisy, the analysis is complicated by the fact that the cells/nuclei may be unevenly spaced in three-dimensional space. This makes the processing, extraction and study of expression signals and intrinsic biological noise a serious challenge for 3D data, requiring new computational approaches. Here, we present a new approach for studying gene expression in nuclei located in a thick layer around a spherical surface. The method includes depth equalization on the sphere, flattening, interpolation to a regular grid, pattern extraction by Shaped 3D-SSA, and interpolation

back to the original nuclear positions. The approach is demonstrated in several examples of gene expression in the zebrafish egg (a model system in vertebrate development). The method is tested on several different data geometries (e.g. nuclear positions) and different forms of gene expression patterns. Fully 3D datasets for developmental gene expression are becoming increasingly available; we discuss the prospects of applying 3D-SSA to data processing and analysis in this growing field.

For reconstructing the spatio-temporal dynamics of gene expression, the raw data sets are stacks of confocal microscope scans of early embryos (fixed or live). Data is usually the intensity of fluorescent markers for either the mRNA or proteins encoded by the genes of interest. Extracting this data requires image segmentation to identify the signal for each cell (or nucleus). This produces text files with the spatial coordinates, gene expression levels and lineage history of each cell.

A major goal of this processing is to collect reliable quantitative data for the fitting and verification of modern computer dynamic and stochastic models of developmental gene regulation at single cell resolution.

The data in cellular resolution 3D gene expression atlases typically has very high noise, with contributions from aspects such as the intrinsic gene expression noise observed in prokaryotes and eukaryotes and the disorder in cellular/nuclear positions. New quantitative approaches are needed to separate the raw expression data into signal and noise components. While some animals' embryos have simpler geometries, being relatively flat with spherical or ellipsoidal cell layers (like the early *Drosophila* fly embryo), many types of embryos have inherently spatially three-dimensional cell order, adding methodological difficulties concerning specimen thickness and optical non-transparency. Such 3D challenges require new experimental and computational approaches.

Different embryo geometries can produce different spatial characteristics on the gene expression data. For instance, in the early *Drosophila* embryo the data can be considered as patterns on an ellipsoidal surface. With an appropriate two-dimensional unfolding (e.g. cylindrical projection) the data can be studied with 2D image processing techniques, see e.g. Section 7.4 and [208].

There are, however, datasets of recent experimental data which are truly 3D and cannot be properly transformed and analyzed in 2D. A prime example is expression data from early zebrafish embryos, where nuclei (cells) are in several irregular layers on the fish egg. Such data requires techniques and algorithms for directly processing 3D data. The irregular distribution of nuclei in layers presents an added challenge, since most quantitative methods operate on spatially regular data points.

To this end, this section introduces a new method for processing irregular data points scattered in layers in the vicinity of an ellipsoidal (spheroidal) surface [11]. We present a non-parametric method, which can address the arbitrary spatial distribution and unknown noise character of the expression data. In Section 7.4, we presented extensions of 2D-SSA to analyze 2D and surface 3D datasets from *Drosophila* confocal scans. These extensions, circular and shaped 2D-SSA, were applied to gene expression patterns in the thin nuclear layer just under the surface of the embryo. We demonstrated how circular and shaped 2D-SSA can decompose the expression data into identifiable components (trend and noise), as well as separate signals from different genes.

In this work, we extend SSA to irregular three-dimensional expression data (3D-SSA). For an initial application of the approach, we focus on dealing with spatial irregularity — using real zebrafish data for the spatial coordinates of nuclei and the spatial gene expression patterns from the ‘MatchIT’ and ‘AtlasIT’ packages [206], but using artificial functions (simple math functions and a smooth approximation of an intensity indicator) for the expression (fluorescence) intensities.

Figures 7.26 and 7.27 illustrate the main steps of the 3D-SSA approach (see details in Section 7.5.2). Figure 7.26a shows simulated intensity data (a two-exponential pattern, with noise) on experimental data for nuclear positions in the spherical-cap geometry of the gene expression region. Figure 7.27 shows the algorithm steps: depth equalization on the sphere, flattening, interpolation, and reconstruction. Figure 7.26b shows the resulting pattern reconstruction after the application of the 3D-SSA algorithm. The coloring of nuclei corresponds to gene expression intensities. Mixing of nuclei of different colors reflects noise. Regular (smooth) color patterns reflect noise removal.

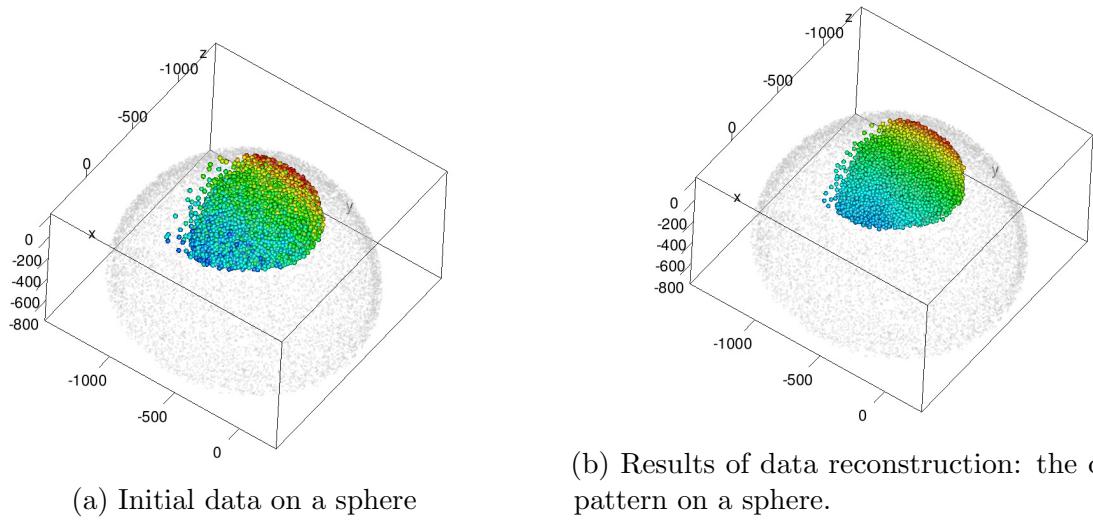


Figure 7.26: Zebrafish data and the 3D-SSA approach: original nucleus position data for the spherical-cap gene expression region, with simulated intensity values (a schematic two-exponential pattern, with noise) represented by the color map; colors are given in the topographic scale, where the blue color corresponds to small values when brown/red colors correspond to large values of intensity.

This section is structured as follows. Section 7.5.1 describes the semi-artificial datasets which were analyzed. The approach is described in Section 7.5.2. In Section 7.5.3, the reliability of the approach is demonstrated on semi-artificial data similar to real observations.

In [11], several examples are considered. The first example considers all nuclei detected in the specimen at the shield stage, in which nuclei are distributed in a “spherical cap”, and expression (with noise) is generated for two patterns: a) the sum of two exponentials, and b) bell-shaped. The second example is for expression patterns similar to those for the nine regulatory genes characterized in [206]. Specifically, the test pattern is limited to nuclei where the *ntla* gene is found experimentally (the ‘MatchIT’ package [206]). This set of nuclei is distributed in an

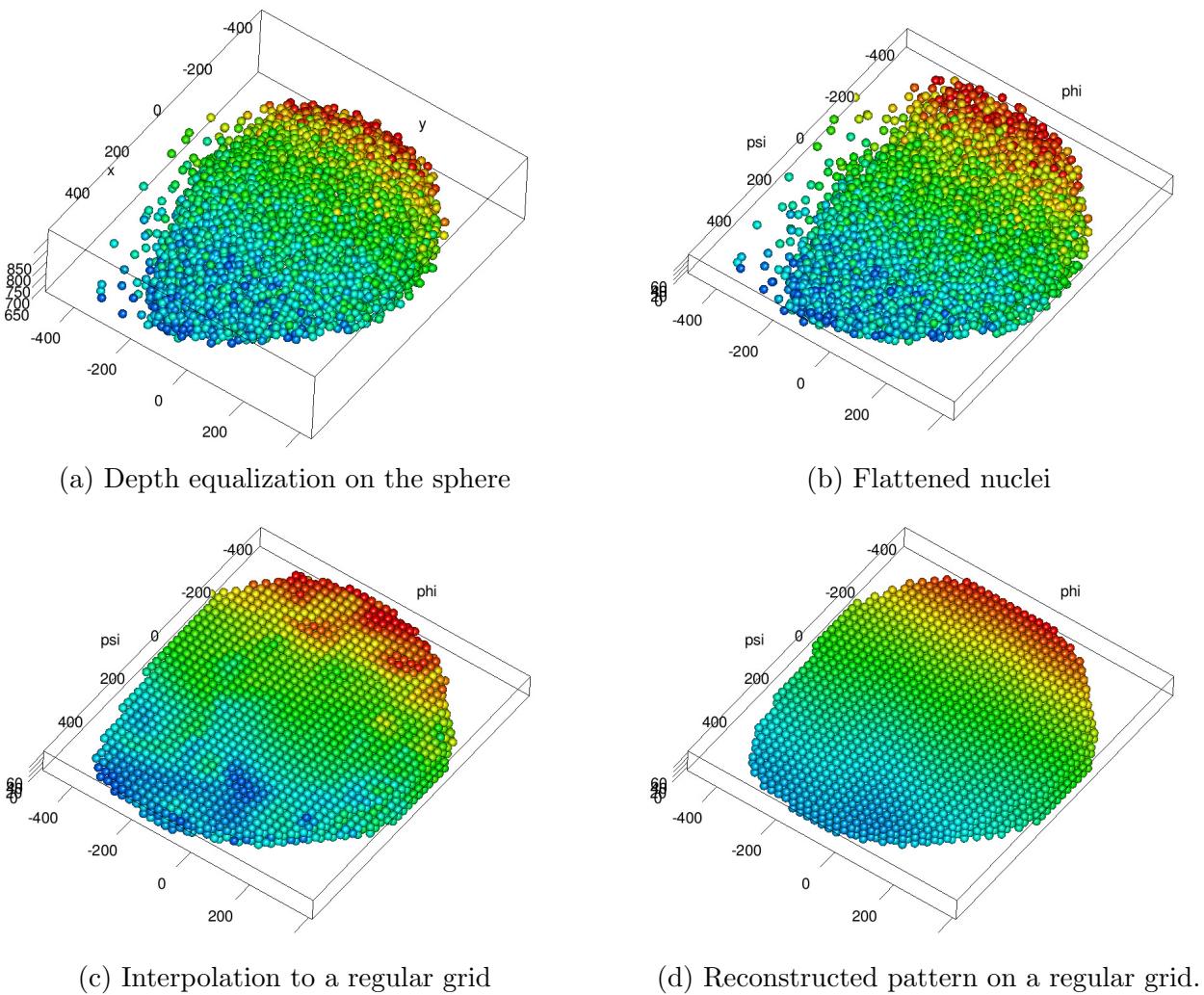


Figure 7.27: Processing steps in the 3D-SSA approach.

equatorial strip. For its extraction, we build a hull (envelope) of expressed nuclei (generally not convex since the expressing area has a complex shape). Shaped 3D-SSA is applied in all of the test cases. In the dissertation, we describe the general methodology and the first example from [11] with the two-exponential pattern, which was considered in [45] for 1D profiles, in detail.

### 7.5.1. Data

In a 3D dataset from a zebrafish embryo, each datapoint corresponds to a nucleus, each represented by an array of numbers: three spatial coordinates for the nucleus centroid, and the fluorescence intensities (in arbitrary units) of the labelled genes (usually two genes are labelled per embryo). Geometrically, the datapoints are distributed around a 3D ellipsoid in several irregular layers (see Figures 7.26, 7.29a, 7.30a). If we approximate the fish egg as an ellipsoid (or spheroid), then the early fish embryo can be geometrically described as a thick (multilayer) spherical cap overlaying the egg, which can be flattened to a disc without substantial distortions at the margins (biologists refer to the geometry of these embryonic stages as ‘dome’ or ‘disc’). The key genes

studied at these early stages tend to form expression patterns in compact subareas of the spherical cap, such as open-ended rings, etc. Preprocessing and SSA procedures can be focused or confined to these subareas. In other words, we assume that there is a transformation of a given expression area to a parallelepiped and that the transformation does not distort the data drastically. If the expression area is too large for such a transformation as a whole, then the transformation can be done as several independent pieces.

### 7.5.2. Method

SSA-type methods process data specified on a regular grid within a parallelepiped. Therefore, application to irregular 3D data first requires flattening, followed by regularization.

Processing 3D data which is in a layer near an ellipsoidal surface consists of the following steps:

- Detection of data location: estimation of the ellipsoid center and finding the nuclear centroid positions relative to this. Enables data rotation for simpler, non-distorting flattening.
- Flattening the data and embedding them into a parallelepiped.
- Interpolation to a regular grid  $\{i = 1, \dots, N_1\} \times \{j = 1, \dots, N_2\} \times \{k = 1, \dots, N_3\}$  to obtain  $f_{ijk}$ .
- Application of 3D-SSA, perhaps by the shaped version to confine the analysis to sub-areas. 3D-SSA results in a decomposition of the form:  $f_{ijk} = s_{ijk} + r_{ijk}$ , where  $s_{ijk}$  corresponds to the expression pattern, or trend.
- Interpolation of  $s_{ijk}$  back to nuclei in the parallelepiped.
- The transformation back to the original coordinates of nuclei in the embryo.

This process results in the extracted pattern and residual noise, in the original geometry of the embryo. This allows for further study of the pattern's form (e.g. comparison with deterministic dynamic gene regulation models), as well as of the model for the noise (e.g., to detect whether noise is additive or multiplicative).

Below, we comment on the steps of the processing scheme in further detail. For simplicity, we assume data are located near the surface of a sphere (the case for zebrafish eggs).

#### Detection of data location

The origin of the coordinate system is the center of the sphere, estimated as the point mostly equidistant from all data points (more formally, we find a point minimizing the variance of distances between its position and those of the data points).

Two types of spatial data distributions are considered, which each have a specific procedure for re-orientation and flattening.

The first type is for data located near the spherical cap. In this case, the ‘z’-axis passes through the center of the sphere and the middle point of the data; we can rotate the data to obtain positive ‘z’-values for all nuclear coordinates.

In the second type, data are located in a strip along the equator of the sphere. In this case, the ‘z’-axis is chosen orthogonal to the equatorial plane and passes through the center of the sphere.

In both cases, ‘x’- and ‘y’-axes are chosen orthogonal to the ‘z’-axis and to each other, oriented to maintain the original axis orientation as much as possible.

## Flattening the data

Data is embedded into a parallelepiped with sides parallel to the axes. The first axis corresponds to depth in the data layer, with the second and the third axes corresponding to surface directions. We aim to keep the proportions of the data as unchanged as possible.

**Depth equalization of data** We assume that the data are located in a layer of approximately constant depth on a spherical surface. Before projection, the data should be corrected to be within an ideal spherical layer of constant depth. Suppose that all nuclei are located in an area which is bounded by two convex surfaces (e.g. in a spherical layer).

The first step of the procedure is to find these surfaces as convex external and internal hulls (envelopes) applying the classical ‘convex hull’ method to original data for finding the external hull and to inverted data to find the interior one. Then, the found exterior and interior hulls are transformed to spherical surfaces of radii  $R + D/2$  and  $R - D/2$  correspondingly, where the sphere radius  $R$  is estimated as the median of distances from the data points to the sphere center, which was estimated in Section 7.5.2, and the layer depth  $D$  is estimated as the median distance between the hulls.

The procedure can detect a too-thin layer of one nucleus depth when 2D-SSA should be used instead of 3D version. For data which is 2D rectangular, the 2D-SSA method discussed in Section 5.2.3 is applied. Otherwise, a new modified 3D approach is elaborated.

**Projection** Spherical projections can have different invariants. For data in spherical caps, we use the equidistant azimuthal projection [209, Section 25], to maintain distance from the center point (i.e. latitude) and therefore the original layer’s linear sizes. After flattening and projection, the data points (nuclei) will therefore have the following coordinates:  $\psi, \varphi$  are coordinates along orthogonal meridians in the azimuthal projection,  $d$  represents depth; all are measured in metrical units.

For data in equatorial strips, we use the equidistant cylindrical projection [209, Section 12], maintaining distances of points to the equator (i.e. latitudes) and distances between points on the equator. This gives similar coordinates  $\psi, \varphi$  and  $d$  (latitude, longitude and depth measured in metrical units). If an equatorial strip encircles the whole equator, we obtain a parallelepiped with the circular topology on the equatorial coordinate and can apply the circular version of SSA [41].

Note that we obtain new coordinates in approximately the same units (and the same proportions) as the original data. This is the main purpose of using equidistant projections, since proportions can strongly influence the interpolation to a regular grid.

We use relative coordinates for the flattened data. Thus, in all pictures  $d$  is reported as a percentage, with 0% at the inner surface and 100% at the outer surface;  $\psi, \varphi$  are reported as fractions of the equator length.

### **Interpolation of nuclei to a regular grid and back**

Interpolation of irregular data to a regular grid is known as a ‘3D scattered data interpolation problem’ (see [210] for a description and an overview of different approaches).

We use a ‘triangulated irregular network-based linear interpolation using the Delaunay triangulation’ approach, where the interpolation is constructed by linear interpolation of the vertex values from the corresponding triangulation simplex. Implementation is performed with the help of the library ‘CGAL’ [211].

Note the nuclei do not necessarily pack the whole parallelepiped, and edge effects are a consideration. Therefore, after interpolation, we obtain the expression data on a subset of the parallelepiped grid. For cap-shaped data, for example, the subset is a disc. This subset can be processed with the shaped version of 3D-SSA.

For back-interpolation to nuclei, the conventional trilinear interpolation is used for pattern reconstruction. Residual values are calculated as differences between original and extracted-pattern (trend) expression values.

### **Shaped 3D-SSA**

Shaped 3D-SSA methods need original data to be given on a subset of a regular grid (which we call the initial shape).

A parameter of the method is the shaped 3D window, which is inscribed in a parallelepiped of sizes  $L_1 \times L_2 \times L_3$ .

It is assumed that the chosen window covers all the points of the original shape. If not, the uncovered grid points are removed. For a proper window shape, the number of removed points should not be large. If so, another window shape or size should be chosen.

Section 5.1 describes Shaped SSA for 2D objects, just for simplicity of notation. The method Shaped 3D-SSA, which we use in the section, is a natural extension of Shaped 2D-SSA to the three-dimensional case.

### **Choice of parameters**

The decomposition of gene expression data on irregular nuclear positions has several parameters: for interpolation and flattening and for SSA (the window shape and size, and the number  $r$  of components for reconstruction).

To obtain a sufficient number of grid points with respect to the number of nuclei, the steps of the regular grid should not be too large. The upper bound for the grid points is limited only by computational costs.

Recommendations for 3D-SSA are similar to that for 2D- and 1D-SSA given in the previous chapters. Larger windows correspond to more refined decomposition and more accurate reconstruction if the signal (pattern) has a simple structure generating a few SVD components. For more complex patterns, medium to small windows are preferable.

Note that window size is measured with respect to pattern features and should not depend on the interpolation step. Therefore, window sizes are measured as a percentage of image sizes, not in the number of grid points. Starting window sizes can be chosen as approximately 10–20% of the image sizes in each direction. If the pattern is extracted imprecisely, the window can be enlarged; if the pattern is mixed with the residual (noise), the window should be decreased.

Identification of pattern components can be performed by analyzing the forms of eigenarrays or elementary reconstructed components (see the example in Section 5.3). Slowly-varying patterns can be constructed readily by the accumulation of slowly-varying elementary components. Since it is difficult to perform a visual analysis of 3D objects, it is preferable to depict slices (1D graphs or 2D images), obtained by fixing one or two coordinates.

The quality of pattern extraction can be checked through residual behaviour. For proper pattern extraction, residuals should vary around zero. Thus, it is recommended to choose slowly varying elementary components such that the residual has no part in the pattern.

## Model of residuals

For understanding biological sources of noise in gene expression, it is of interest to extract a model of the noise from the data. We suggest a method for noise model detection based on a standard test of heteroscedasticity of residuals with different normalizations.

For a decomposition of initial data into pattern and noise:  $x_i = s_i + r_i$ ,  $i = 1, \dots, N$ , where  $N$  is the number of nuclei (enumeration by one index instead of three does not affect the results), assume that noise in nuclei is independent and consider the model

$$r_i = \varepsilon_i \cdot |s_i|^\alpha, \quad (7.11)$$

where  $E\varepsilon_i = 0$ ,  $D\varepsilon_i = \text{const}$ .

If  $\alpha = 0$ , the noise is additive (its standard deviation does not depend on pattern values). If  $\alpha = 1$ , the noise is multiplicative (its standard deviation is proportional to pattern values). The intermediate value  $\alpha = 0.5$  corresponds to Poissonian noise where noise variance is proportional to pattern value.

The Park method [212] estimates  $\alpha$  as the slope of the linear regression line in the model

$$\log |r_i| = \log \sigma + \alpha \log |s_i| + v_i, \quad (7.12)$$

where  $v_i = \log |\varepsilon_i/\sigma|$  is a well behaved error term. The Park method appears to be robust to the distribution of the residuals. An estimate of  $\alpha$  in the model (7.12) can be obtained, for example, by the least-squares method.

## Implementation

All the described methods are implemented in R and are included in the BioSSA package.

### 7.5.3. Example for spherical-cap nuclear pattern

Here we work through an application of the 3D-SSA procedure on the close-to-spherical zebrafish egg. Nuclear coordinates are taken from the default MatchIT example. The ‘MatchIT’ tool was run with the default dataset and parameters, producing files with automatically detected nuclei. Processing is on the file ‘gsc\_ntla\_wt\_t008\_ch01.csv’, containing nuclear coordinates for the cells expressing the *gsc* (‘goosecoid’) gene at the ‘late shield’ developmental stage.

In this data, nuclei are located in a thick layer on a sphere. Each nucleus is marked by ‘1’ or ‘0’, for the gene expressing or not, respectively. For this data set, *gsc*-expressing nuclei (‘1’-s) are located within a small spherical cap. We extend the analyzed area to include all ‘1’-nuclei, plus a surrounding ring of nuclei.

We first present results for the example and explain the choice of parameters and pattern components. We then show how the method can be used to estimate the model of noise, after extracting the pattern. In the considered examples, we add Gaussian noise to patterns; however, it is not essential, since the SSA-family of methods is stable with respect to noise distribution and possible weak dependencies.

## Two-exponential expression pattern

Let us construct the expression values as

$$s(\psi, \varphi, d) = e^{2\varphi + 9\psi + 2d} + 2e^{-(2\varphi + 13\psi + d)}, \quad (7.13)$$

$$v(\psi, \varphi, d) = s(\psi, \varphi, d) + \varepsilon \cdot s(\psi, \varphi, d), \quad (7.14)$$

where  $\psi, \varphi$  are relative spherical coordinates,  $d$  is layer depth ( $d \in [0, 1]$ ) and  $\varepsilon$  is white Gaussian noise with standard deviation 0.35. This pattern, which we call CAP-2EXP, is the sum of two exponentials plus multiplicative noise. Note that the pattern depends on three coordinates and therefore it is varied in three directions. Moderate noise levels here and in the other considered simulated examples were chosen for the better visual color representation of the results for the considered patterns.

Inside and outside views of the nuclear hulls are shown in Figures 7.28a–7.28b, where colors correspond to expression levels, as in Figure 7.26. It can be seen that the coloring is variegated (nuclei of different color occupy similar positions), reflecting the presence of noise.

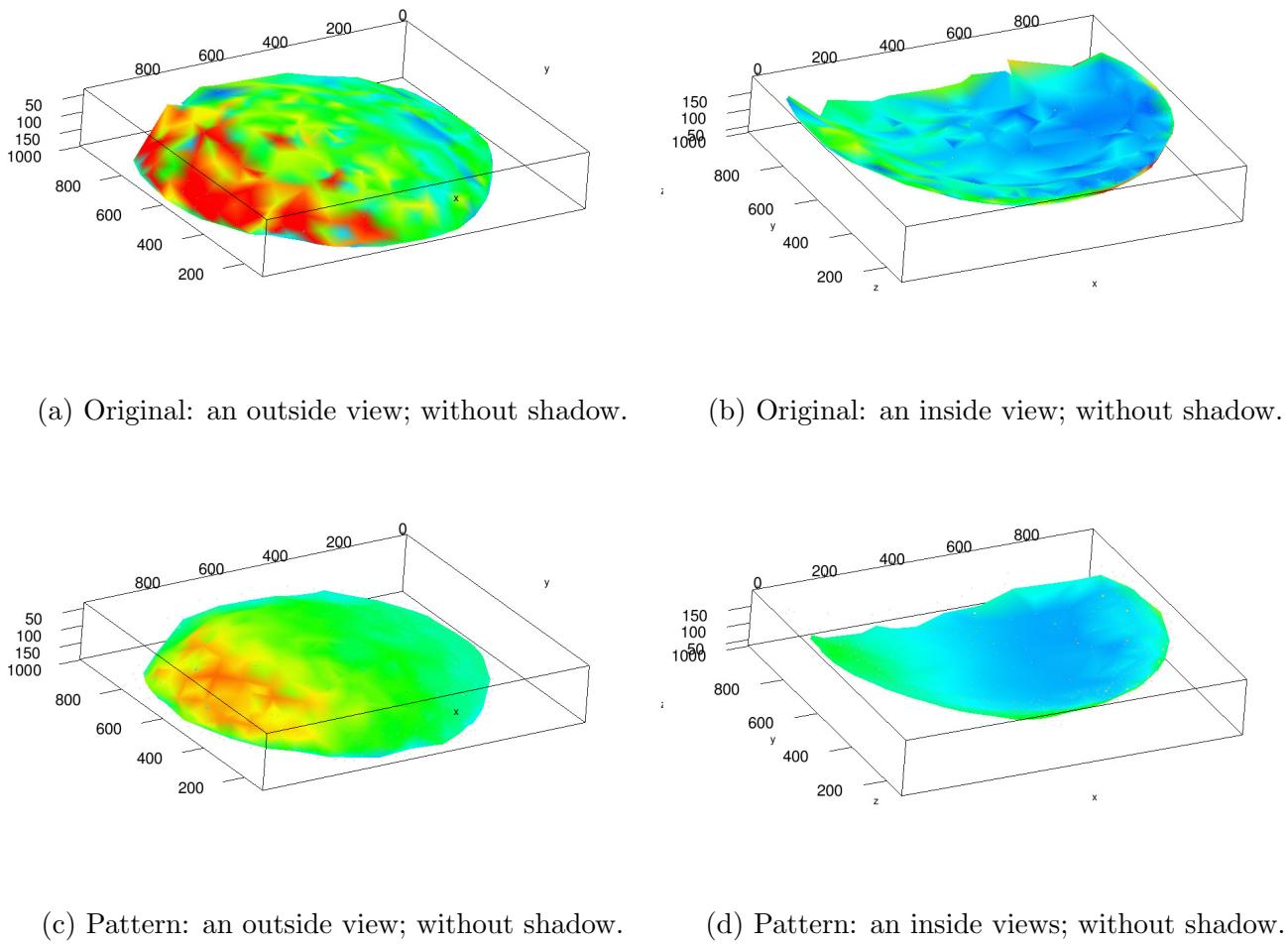


Figure 7.28: CAP-2EXP: the nuclear hulls, coloring based on original and pattern expression intensities.

The method described in Section 7.5.2 produces the reconstructed pattern depicted in Figures 7.28c and 7.28d. The results of noise removal are clearly seen.

For a better visual representation, the nuclei themselves can be depicted, see Figure 7.29 and 7.30. As before, the color of a nucleus reflects the expression level in the same scale as in Figure 7.26. The expression pattern can be clearly seen in the denoised data. The difference between the inside and the outside views demonstrates the pattern dynamic in the depth direction.

After pattern extraction, the noise model can be estimated (see Section 7.5.2). In (7.13), multiplicative noise with  $\alpha = 1$  was simulated. Applying the Park method provides an estimate of  $\hat{\alpha} = 1.073$ , recovering the multiplicative character of the generated noise and demonstrating how the processing can distinguish between, for example, additive, Poissonian and multiplicative noise in datasets.

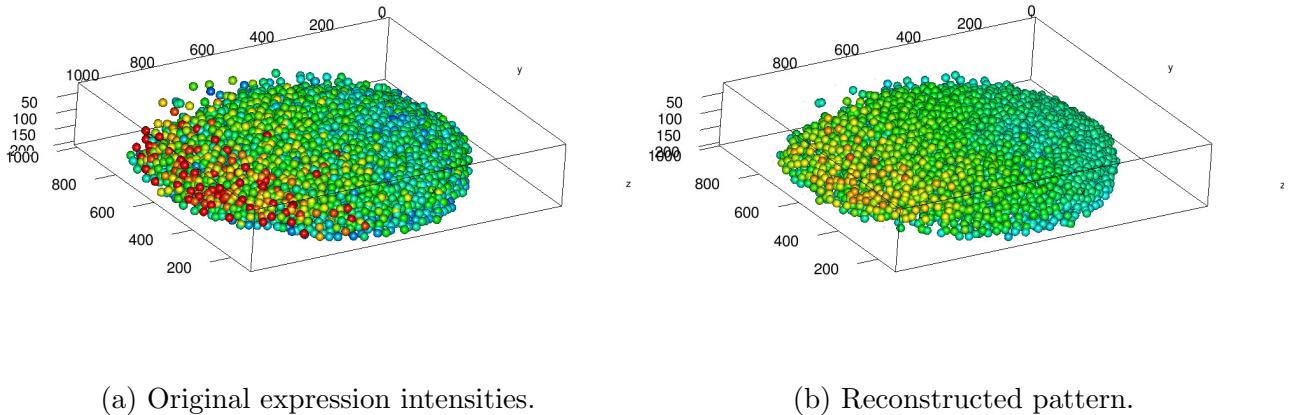


Figure 7.29: CAP-2EXP: nuclei colored; an outside view.

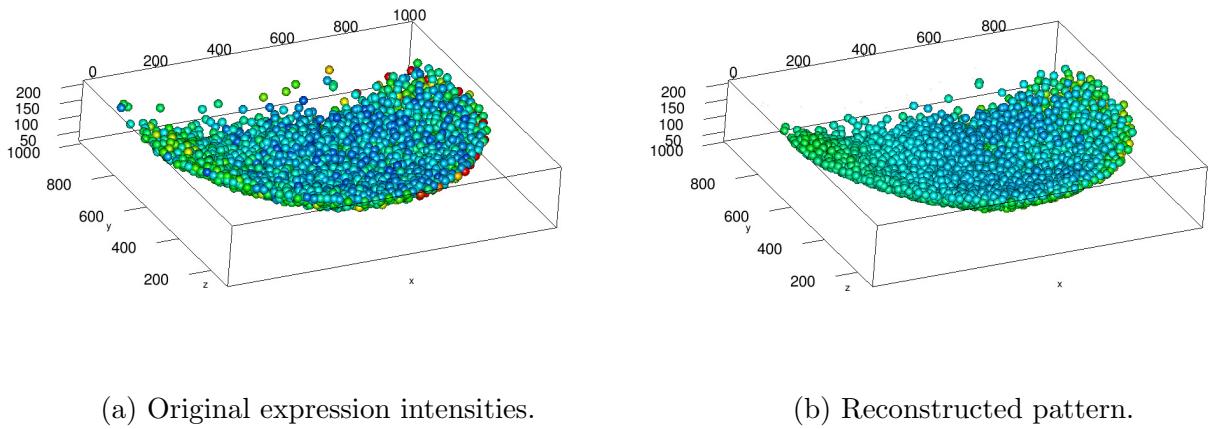


Figure 7.30: CAP-2EXP: nuclei colored; an inside view.

### The chosen parameters

After flattening, as described in Section 7.5.2, we obtain a parallelepiped with length  $l \approx 1030$ , width  $w \approx 885$  and depth  $d \approx 50$ . Since the shape of the flattened nuclear cloud is similar to a spherical cap, the equidistant azimuthal projection was applied.

The step size was chosen the same in all directions to obtain  $10^6$  grid points.

For Shaped 3D-SSA, the ellipsoid window was inscribed in a parallelepiped of size  $L_1 \times L_2 \times L_3$ , where the  $L_i$  are equal to 40% of the original image sizes. The total number of nuclei in the data file is 3595, while the chosen window covers approximately 160 nuclei on average. The number of nuclei covered by all positions of the chosen window is 3306; that is, a few side nuclei were not considered.

To identify pattern components, let us examine 2D slices of 3D eigenarrays and elementary reconstructed components. The first two ellipsoidal eigenarrays are smooth, the third one has some oscillations. Therefore, we choose the first two components for pattern reconstruction.

The six leading elementary components, which are generated by the six eigenarrays, together with the original and residual 2D slices are shown in Figure 7.31.

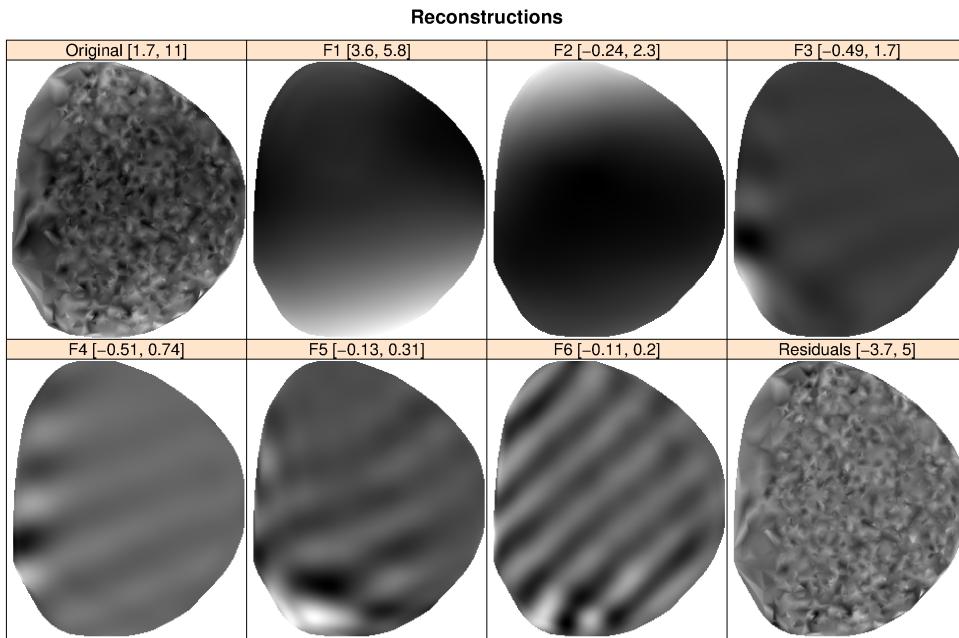


Figure 7.31: CAP-2EXP: Elementary component reconstructions, slices.

Figure 7.31 confirms the choice of the two leading components (that is,  $r = 2$ ) as corresponding to the pattern.

The pattern is reconstructed from the leading eigenarrays on the regular grid points, then interpolated back to flattened (irregular) nuclei, and then transformed to the original nuclear positions on the zebrafish egg.

The reconstruction result is quite robust to the window choice. For example, the pattern reconstruction is approximately the same if we choose a window size of 35% instead of 40% of the original size. However, for more complex patterns, smaller window sizes are preferable; see discussion regarding the choice of the window in [29] (1D case) and [208] (2D case).

### Check of proper pattern extraction

Inspection of the reconstructed images clearly demonstrates the noise removal for both test patterns. However, this does not prove that we have reconstructed the whole pattern. We now show, on the CAP-2EXP example, that the pattern reconstruction is complete.

Let us consider 2D slices of the reconstructed values on the regular grid of flattened nuclei fixing the depth. The vertical axis will represent expression values (colormapped in, e.g. Figure 7.28) and the horizontal axes correspond to  $\phi$ ,  $\psi$  nuclear or grid-point positional coordinates.

Since the nuclei may not be located exactly on the slice, we consider nuclei from the layer plus-minus 10% to each side. The extracted pattern on the nuclear slice is depicted by a solid surface, with nuclear expression values shown as individual dots, see Figure 7.32 (top). In Figure 7.32 (bottom), the residuals are obtained by subtraction of the pattern from the nuclear values. The even scatter of the residuals around the zero ‘xy’-plane indicates a good fit of the reconstruction to the data.

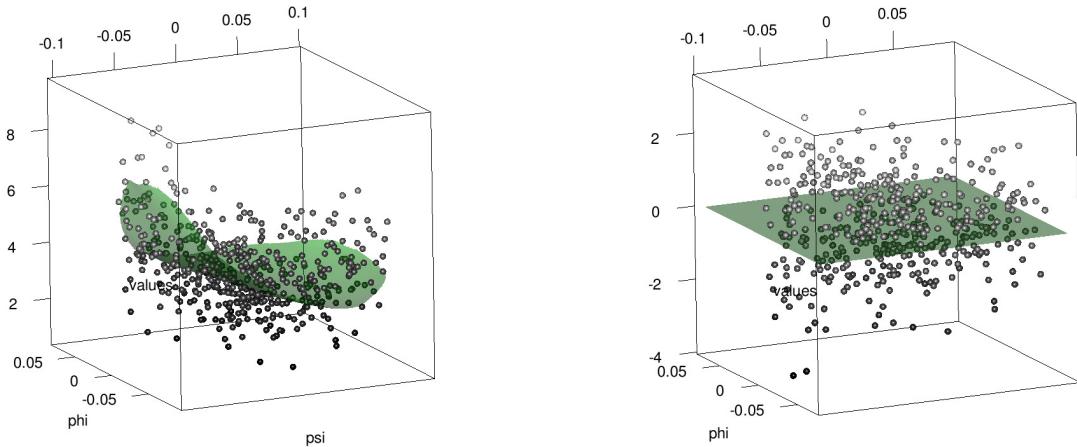


Figure 7.32: CAP-2EXP pattern: 2D slice with expression values, depth  $d = 82.5\%$ ; top: the surface depicts the reconstructed pattern (in the regular grid points), the points show expression values in individual nuclei from a  $\pm 10\%$  layer; bottom: nuclear residuals from the reconstruction are scattered evenly around the zero plane.

For a more refined analysis, we can construct 1D slices. For example, fixing the depth at 82.5% and the width ( $\varphi$ ) at 50%, we consider the two-exponential pattern for nuclei from a thin layer of ( $\pm 10\%$  around this depth and width). The extracted pattern on the nuclear slice is depicted by a solid line. Figure 7.33 confirms the quality of the reconstruction, with the residuals between the data and the reconstruction evenly spaced around the zero plane. Note that for estimation of the noise model, the choice of 1, 2, 3 or 4 leading components has little effect on the results.

Thus, we have demonstrated that 3D-SSA is able to extract 3D patterns from gene expression data given on an irregular grid and then analyse a structure of biological noise.

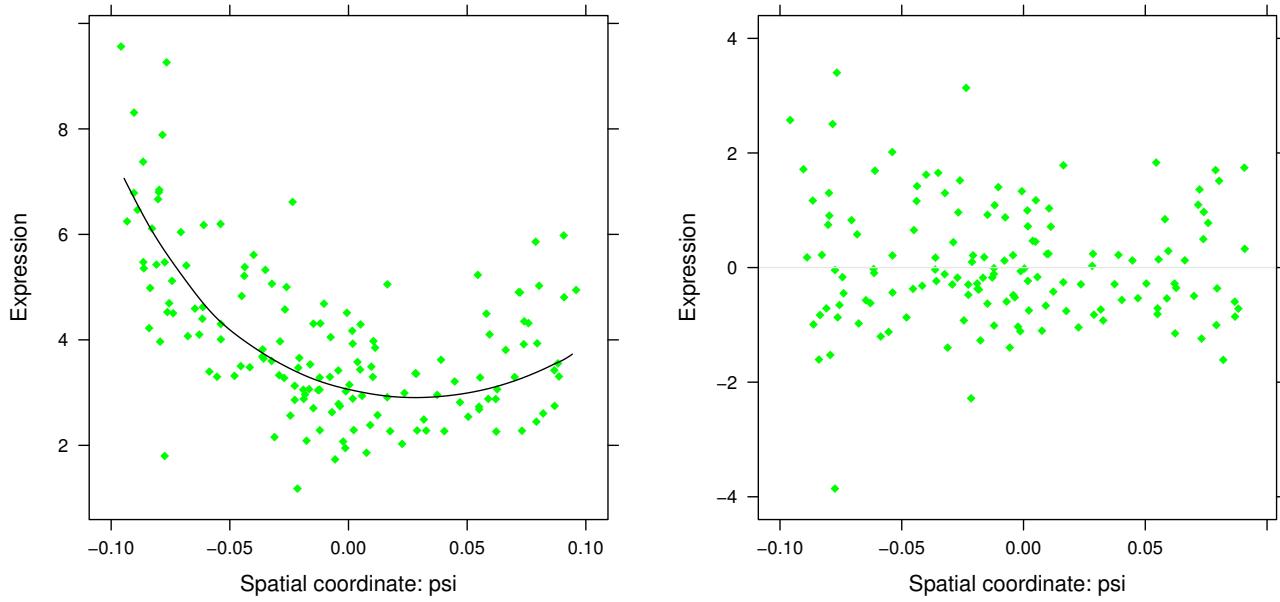


Figure 7.33: CAP-2EXP pattern: 1D slice at 82.5% depth( $d$ ) and 50% width ( $\varphi$ ). Left: pattern on the grid (curve) and original values on the nuclei (points). Right: residuals between the reconstruction and the nuclear data values.

## Conclusion

In the dissertation, various aspects of singular spectrum analysis have been considered. Let us describe the main results obtained and briefly discuss them.

One of the basic results is the construction of a general scheme of the family of SSA methods for the decomposition of an initial object (for example, a time series or a digital image) into interpretable components (Chapter 1). We obtained a scheme in which it is possible to fix the type of object and to change the way of decomposition into elementary components for a more accurate result, or, on the contrary, it is possible to fix the type of decomposition and to apply it to objects of different dimensions and shapes. This approach makes it possible to develop the theory in a unified style and to develop implementations of the proposed algorithms in a unified structure.

The second important topic relates to the decomposition stage of the general SSA scheme. The success of the decomposition of a time series into elementary components depends on whether they can then be grouped in such a way that the desired components, such as a trend or a periodic pattern, can be extracted from the series. In the existing basic version of SSA, the constructed decomposition had some disadvantages (although it is optimal among all adaptive decompositions). For example, two sinusoidal components could not be separated if they had the same amplitudes. It was proved that the proposed DerivSSA and SSA-ICA methods overcome this problem. Another example: to separate the trend from the periodicity in a short time series, the basic version imposes rather strict conditions on the length of the series and the length of the window. The Iterative OSSA method allows one to separate such components. In general, after the appearance of the proposed methods, the range of separable components has significantly expanded. The ProjSSA method helps to improve the separability of polynomial trends. All these methods are described in Chapter 2 for one-dimensional time series. Note that the proposed approaches naturally extend to multivariate objects. We have not written out the results for multivariate source data, since the algorithms are similar and writing them down would be too cumbersome. In the R package RSSA multivariate DerivSSA and Iterative OSSA methods are implemented.

The results of Chapter 3 cover another aspect of the application of SSA to one-dimensional time series. Namely, the case of a signal governed by a linear recurrence relation is considered. Algorithms for filling gaps in time series were proposed and justified. Note that the proposed approach through gap-filling of a part of components in a vector from a given subspace, which is basic in filling gaps in the time series, naturally extends to filling gaps in multidimensional objects as well. As well as for prediction, in the case of gap filling, although the theoretical justification implies that the signal is controlled by the LRR; however, the constructed methods are also applicable for the case when this is true only approximately. Since the class of series controlled by LRR consists of the sums of products of polynomials, exponents, and harmonics, formally any sequence can be approximated by a series from this class on a finite interval; the only question is the rank of the approximating series, which should be low.

Another result of Chapter 3 concerns the case when the signal should be exactly controlled

by an LRR of small dimensionality, which occurs, as a rule, in engineering problems. In this case, the problem can be described as a weighted least-squares problem, where the weights are adjusted to the autocovariance matrix of the autoregressive noise. In this case, both matrix and vector formulations of the problem have been proposed in various papers. It has been proved that for the case of autoregressive noise in matrix form, it is impossible to specify optimal weights (i.e., weights leading to the minimal variance of the signal estimate, asymptotically by series length). The proof for the case of autoregressive processes of orders 1 and 2 is complete, and only the basic result is proved for the general case. The result obtained does not mean that the rather simple solution of the problem in matrix form is inapplicable. The result implies that in the matrix formulation of the problem, the weights must be chosen numerically to be as close to optimal as possible.

In Chapter 3, results for signal detection in red noise were also included. In contrast to the other problems, this problem is quite specific, both in its requirement that red noise is present and in its only partial involvement of the SSA algorithm. However, the already known algorithm is called Monte Carlo SSA, so we considered it along with other methods. Within the Monte Carlo SSA approach, we proposed a multiple and weighted version of the algorithm for testing the hypothesis that there is no signal in the noise, which was statistically validated in terms of controlling the family-wise error rate of the first kind. The result obtained can be generalized to the case of MSSA, but the multivariate case is much more complicated in terms of the presence of different signals in different series from the analyzed time series system. Therefore, the generalization requires a separate study.

Chapter 3 also contains a study of the effect of window length on the accuracy of signal estimation and prediction. The study is mainly numerical, although it contains the result for the case of a constant signal obtained analytically.

Part of the dissertation work is devoted to multivariate generalizations of SSA. The most frequent generalization is the MSSA method for the analysis of a time series system because there is an idea that under some conditions the accuracy of signal extraction can become higher than analyzing each time series separately. The notion of matched time series is introduced, which is key to improving accuracy, and sufficient separability conditions for the components of the series are derived. The material concerning MSSA has been allocated to a separate chapter 4, since, on the one hand, MSSA is the closest extension of SSA, and, on the other hand, it has its own actual features.

The next point to note is the development of a general approach to objects of different dimensions and shapes through an approach called Shaped SSA. Chapter 5 proposes a formalization that allows all variants of objects in the univariate and multivariate cases to be considered in a unified style. Multidimensional digital image analysis is given as a special case, and it is shown that both SSA analysis of series with gaps and MSSA can be viewed as special cases of Shaped SSA. The approach through Shaped SSA is useful both in terms of unification and in terms of the analysis of non-rectangular digital images. As an example, a photo of Mars, which, has a circular shape, was considered; accordingly, the window was also circular.

Chapters 2, 4, and 5 additionally describe an approach for the automatic identification of elementary components in a decomposition for trend and periodicity extraction. The approach is based on similar principles but has its specifics for different types of data.

The general approaches to SSA algorithms described above and their modifications allowed us to create a package RSSA written in R language. The package contains a lot of different methods applicable to objects of different shapes, and perhaps it would not have been possible to create it without the development of the general approaches described above. The structure of the package is described in Chapter 6.

Chapter 7 concludes the dissertation with applications of the SSA method to real data. The first application is the application of SSA to the prediction of Earth rotation parameters, where the advantage of SSA was obtained in comparison with publicly posted predictions from two other sources. In another example, SSA's ability to smooth time series was used to analyze univariate data not related to time series, namely to estimate density from empirical data. The approach using smoothing the empirical distribution function, especially in the case of artificially discrete data, was found to give good results.

Participation in scientific projects devoted to gene expression analysis has led to three sections devoted to this very topic. Note that the data are also not time series; they are obtained by measuring gene activity at different spatial points. In Section 7.3, the application of SSA to estimating the parameters of the proposed model allowed us to examine the dynamics of embryonic development quantitatively, through changes in model parameters. The next two sections show examples of applications of 2D-SSA and 3D-SSA for signal extraction in the case of 2D and 3D gene expression data. One application of the results is the construction of a noise model, since there is still no definite opinion on whether the noise is additive or whether the magnitude of the noise is proportional to the magnitude of the pattern. The 2D-SSA method does a good job of separating the noise and has a means of testing how well the pattern is separated, but the biological conclusions are complicated by insufficient data quality and the presence of a so-called background in the measurements. Another effect that 2D-SSA has been able to find and eliminate is the effect of mixing the activity of different genes in simultaneous measurements. The success of the application relies, among other things, on the fact that it is possible to select regions of interest that are not necessarily rectangular and apply Shaped 2D-SSA to them.

The application of SSA to three-dimensional data is largely illustrative, as very little such real data are currently available and therefore partially artificial data were analyzed. However, the developed technique for analyzing gene activity in unequally spaced three-dimensional points seems useful as an example of application to data, which should appear in greater numbers in the future with the development of measurement techniques.

## Acknowledgment

I am grateful to my colleagues from Saint-Petersburg State University and especially from Department of Statistical Modelling for their help, my scientific supervisor and then colleague Vladimir Nekrutkin for teaching me attitude to work and life, my bachelor, master and PhD students for motivating and developing the SSA methodology together with me. Last but not least, all thanks and love to my family, husband and daughter for support and granddaughter for keeping the good mood.

## Bibliography

1. *Golyandina N., Nekrutkin V., Zhigljavsky A.* Analysis of Time Series Structure: SSA and Related Techniques. — Chapman&Hall/CRC, 2001.
2. *Markovsky I.* Low Rank Approximation: Algorithms, Implementation, Applications (Communications and Control Engineering). — 2nd edition. — Springer, 2019.
3. *Cadzow J. A.* Signal enhancement: a composite property mapping algorithm // *IEEE Transactions on Acoustics, Speech, and Signal Processing*. — 1988. — Vol. 36, no. 1. — Pp. 49–62.
4. *Golyandina N., Osipov E.* The “Caterpillar”-SSA method for analysis of time series with missing values // *Journal of Statistical Planning and Inference*. — 2007. — Vol. 137, no. 8. — Pp. 2642–2653.
5. *Golyandina N.* On the choice of parameters in Singular Spectrum Analysis and related subspace-based methods // *Statistics and Its Interface*. — 2010. — Vol. 3, no. 3. — Pp. 259–279.
6. Gene expression noise in spatial patterning: Hunchback promoter structure affects noise amplitude and distribution in Drosophila segmentation / D.M. Holloway, F.J.P. Lopes, L. da Fontoura Costa et al. // *PLoS Computational Biology*. — 2011. — Vol. 7, no. 2.
7. *Golyandina N., Pepelyshev A., Steland A.* New approaches to nonparametric density estimation and selection of smoothing parameters // *Computational Statistics and Data Analysis*. — 2012. — Vol. 56, no. 7. — Pp. 2206–2218.
8. *Golyandina N., Korobeynikov A.* Basic Singular Spectrum Analysis and forecasting with R // *Computational Statistics and Data Analysis*. — 2014. — Vol. 71. — Pp. 934–954.
9. *Golyandina N., Shlemov A.* Variations of singular spectrum analysis for separability improvement: Non-orthogonal decompositions of time series // *Statistics and Its Interface*. — 2015. — Vol. 8, no. 3. — Pp. 277–294.
10. Shaped singular spectrum analysis for quantifying gene expression, with application to the early Drosophila embryo / A. Shlemov, N. Golyandina, D. Holloway, A. Spirov // *BioMed Research International*. — 2015. — Vol. 2015.
11. Shaped 3D Singular Spectrum Analysis for Quantifying Gene Expression, with Application to the Early Zebrafish Embryo / A. Shlemov, N. Golyandina, D. Holloway, A. Spirov // *BioMed Research International*. — 2015. — Vol. 2015.
12. Multivariate and 2D extensions of singular spectrum analysis with the Rssa package / N. Golyandina, A. Korobeynikov, A. Shlemov, K. Usevich // *Journal of Statistical Software*. — 2015. — Vol. 67, no. 2.
13. *Zhigljavsky A., Golyandina N., Gryaznov S.* Deconvolution of a discrete uniform distribution // *Statistics and Probability Letters*. — 2016. — Vol. 118. — Pp. 37–44.
14. *Golyandina N.E., Lomtev M.A.* Improvement of separability of time series in singular spectrum analysis using the method of independent component analysis // *Vestnik St. Petersburg University. Mathematics*. — 2017. — Vol. 10, no. 1. — Pp. 10–16.

- University: Mathematics.* — 2016. — Vol. 49, no. 1. — Pp. 9–17.
15. Zvonarev N., Golyandina N. Iterative algorithms for weighted and unweighted finite-rank time-series approximations // *Statistics and Its Interface*. — 2017. — Vol. 10, no. 1. — Pp. 5–18.
  16. Golyandina N., Shlemov A. Semi-nonparametric singular spectrum analysis with projection // *Statistics and Its Interface*. — 2017. — Vol. 10, no. 1. — Pp. 47–57.
  17. Two-exponential models of gene expression patterns for noisy experimental data / T. Alexandrov, N. Golyandina, D. Holloway et al. // *Journal of Computational Biology*. — 2018. — Vol. 25, no. 11. — Pp. 1220–1230.
  18. Zhornikova P., Golyandina N., Spirov A. Noise model estimation with application to gene expression // *Journal of Bioinformatics and Computational Biology*. — 2019. — Vol. 17, no. 2.
  19. Golyandina N., Zhigljavsky A. Blind deconvolution of covariance matrix inverses for autoregressive processes // *Linear Algebra and Its Applications*. — 2020. — Vol. 593. — Pp. 188–211.
  20. Golyandina N. Particularities and commonalities of singular spectrum analysis as a method of time series analysis and signal processing // *Wiley Interdisciplinary Reviews: Computational Statistics*. — 2020. — Vol. 12, no. 4.
  21. Quantification reveals early dynamics in Drosophila maternal gradients / A. Shlemov, T. Alexandrov, N. Golyandina et al. // *PLoS ONE*. — 2021. — Vol. 16, no. 8 August.
  22. Zvonarev N., Golyandina N. Fast and stable modification of the Gauss–Newton method for low-rank signal estimation // *Numerical Linear Algebra with Applications*. — 2022. — Vol. 29, no. 4.
  23. Zvonarev N., Golyandina N. Mixed Alternating Projections with Application to Hankel Low-Rank Approximation // *Algorithms*. — 2022. — Vol. 15, no. 12. — P. ID 460.
  24. Zvonarev N., Golyandina N. Low-rank signal subspace: parameterization, projection and signal estimation // *Statistics and Its Interface*. — 2023. — Vol. 16, no. 1. — Pp. 117–132.
  25. Golyandina N. Detection of signals by Monte Carlo singular spectrum analysis: multiple testing // *Statistics and Its Interface*. — 2023. — Vol. 16, no. 1. — Pp. 147–157.
  26. Measuring gene expression noise in early Drosophila embryos: Nucleus-to-nucleus variability / N.E. Golyandina, D.M. Holloway, F.J.P. Lopes et al. // *Procedia Computer Science*. — 2012. — Vol. 9. — Pp. 373–382. — Proceedings of the International Conference on Computational Science, ICCS 2012.
  27. Measuring gene expression noise in early Drosophila embryos: The highly dynamic compartmentalized micro-environment of the blastoderm is one of the main sources of noise / A.V. Spirov, N.E. Golyandina, D.M. Holloway et al. // *Lecture Notes in Computer Science (Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics)* / Ed. by M. Giacobini, L. Vanneschi, W. S. Bush. — Vol. 7246. — Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. — Pp. 177–188.

28. *Okhotnikov G., Golyandina N.* EOP time series prediction using singular spectrum analysis // CEUR Workshop Proceedings (MAchine Learning for EArth ObservatioN Workshop 2019) / Ed. by Th. Corpetti, D. Ienco, R. Interdonato et al. — 2019. — Vol. 2466.
29. *Golyandina N., Zhigljavsky A.* Singular Spectrum Analysis for Time Series. Springer Briefs in Statistics. — Springer, 2013.
30. *Golyandina N., Korobeynikov A., Zhigljavsky A.* Singular Spectrum Analysis with R. — Springer-Verlag Berlin Heidelberg, 2018.
31. *Golyandina N., Zhigljavsky A.* Singular Spectrum Analysis for Time Series. Springer Briefs in Statistics. — 2nd edition. — Springer, 2020.
32. *Golyandina N., Danilov D.* Processing of multivariate time series by the “Caterpillar” method // Principal Components of Time Series / Ed. by D. Danilov, A. Zhigljavsky. — 1997. — (In Russian).
33. *Golyandina N., Nekrutkin V., Stepanov D.* Variants of the ‘Caterpillar’-SSA Method for Analysis of Multidimensional Time Series // Proceedings of II International Conference on System Identification and Control Problems (SICPRO 03). — Moscow: V.A.Trapeznikov institute of Control Sciences, 2003. — Pp. 2139–2168. — (In Russian).
34. *Stepanov D., Golyandina N.* Variants of the ‘Caterpillar’-SSA Method for Forecast of Multidimensional Time Series // Proceedings of IV International Conference on System Identification and Control Problems (SICPRO 05). — Moscow: V.A.Trapeznikov institute of Control Sciences, 2005. — Pp. 1831–1848. — (In Russian). URL: [http://gistatgroup.com/cat/mssa\\_forecast\\_ru.pdf](http://gistatgroup.com/cat/mssa_forecast_ru.pdf).
35. *Golyandina N., Stepanov D.* SSA-based approaches to analysis and forecast of multidimensional time series // Proceedings of the 5th St.Petersburg Workshop on Simulation, June 26-July 2, 2005, St. Petersburg State University, St. Petersburg. — 2005. — Pp. 293–298.
36. *Alexandrov Th., Golyandina N.* Automatic extraction and forecast of time series cyclic components within the framework of SSA // Proceedings of the 5th St.Petersburg Workshop on Simulation. — St. Petersburg State University, 2005. — Pp. 45–50.
37. *Alexandrov Th., Golyandina N.* Selection of parameters for automatic extraction of trend and periodic components of time series in the “Caterpillar” -SSA approach. // Proceedings of IV International conference ”System identification and control problems” SICPR0’05. — 2005. — In Russian.
38. *Golyandina N., Vlassieva E.* First-order SSA-errors for long time series: model examples of simple noisy signals // Proceedings of the 6th St.Petersburg Workshop on Simulation Vol.1, June 28-July 4, 2009, St. Petersburg. — St.-Petersburg State University, 2009. — Pp. 314–319.
39. *Golyandina N., Usevich K.* An Algebraic View on Finite Rank in 2D-SSA // Proceedings of the 6th St.Petersburg Workshop on Simulation, June 28-July 4, St. Petersburg, Russia. — 2009. — Pp. 308–313.
40. *Golyandina N.E., Usevich K.D.* 2D-Extension of Singular Spectrum Analysis: Algorithm and

- Elements of Theory // Matrix Methods: Theory, Algorithms and Applications. — 2010. — Pp. 449–473.
41. *Shlemov A., Golyandina N.* Shaped extensions of Singular Spectrum Analysis // 21st International Symposium on Mathematical Theory of Networks and Systems, July 7–11, 2014. Groningen, The Netherlands. — 2014. — Pp. 1813–1820.
  42. *Alexandrov Th., Golyandina N.* Automation of extraction for trend and periodic time series components within the method “Caterpillar”-SSA // *Exponenta Pro (Mathematics in applications)*. — 2004. — no. 3–4 (7–8). — Pp. 54–61. — In Russian.
  43. *Golyandina N., Florinsky I., Usevich K.* Filtering of Digital Terrain Models by 2D Singular Spectrum Analysis // *International Journal of Ecology & Development*. — 2007. — Vol. 8, no. F07. — Pp. 81–94.
  44. *Golyandina N., Usevich K., Florinsky I.* The analysis of singular spectrum for the digital terrain models filtration // *Geodesy and Cartography*. — 2008. — no. 5. — Pp. 21–28.
  45. *Alexandrov Th., Golyandina N., Spirov A.* Singular Spectrum Analysis of Gene Expression Profiles of Early Drosophila embryo: Exponential-in-Distance Patterns // *Research Letters in Signal Processing*. — 2008. — Vol. 2008. — Pp. 1–5.
  46. *Hall M. Jr.* Combinatorial theory. — Wiley, New York, 1998.
  47. Linear recurring sequences over rings and modules / V. L. Kurakin, A. S. Kuzmin, A.V. Mikhalev, A. A. Nechaev // *Journal of Mathematical Sciences*. — 1995. — Vol. 76, no. 6. — Pp. 2793–2915.
  48. *Bozzo E., Carniel R., Fasino D.* Relationship between Singular Spectrum Analysis and Fourier analysis: Theory and application to the monitoring of volcanic activity // *Computers & Mathematics with Applications*. — 2010. — 08. — Vol. 60, no. 3. — Pp. 812–820.
  49. *Harris T.J., Yan H.* Filtering and frequency interpretations of Singular Spectrum Analysis // *Physica D*. — 2010. — Vol. 239. — Pp. 1958–1967.
  50. *Kumar K. S., Rajesh R., Tiwari R. K.* Regional and residual gravity anomaly separation using the singular spectrum analysis-based low pass filtering: a case study from Nagpur, Maharashtra, India // *Exploration Geophysics*. — 2018. — Vol. 49, no. 3. — Pp. 217–220.
  51. A simple SSA-based de-noising technique to remove ECG interference in EMG signals / J. Barrios-Muriel, F. Romero, F. J. Alonso, K. Gianikellis // *Biomedical Signal Processing and Control*. — 2016. — Vol. 30. — Pp. 117–126.
  52. *Watson Ph. J.* Identifying the Best Performing Time Series Analytics for Sea Level Research // Time Series Analysis and Forecasting: Selected Contributions from the ITISE Conference / Ed. by I. Rojas, H. Pomares. — Cham: Springer International Publishing, 2016. — Pp. 261–278.
  53. On time-variable seasonal signals: Comparison of SSA and Kalman filtering based approach / Q. Chen, M. Weigelt, N. Sneeuw, T. van Dam // VIII Hotine-Marussi Symposium on Mathematical Geodesy: Proceedings of the Symposium in Rome, 17–21 June, 2013 / Ed. by N. Sneeuw, P. Novák, M. Crespi, F. Sansò. — Cham: Springer International Publishing, 2016. — Pp. 75–80.

54. *Hyndman R. J., Athanasopoulos G.* Forecasting: Principles and Practice. — 2nd edition. — Australia: OTexts, 2018.
55. Forecasting U.S. Tourist arrivals using optimal Singular Spectrum Analysis / H. Hassani, A. Webster, E. S. Silva, S. Heravi // *Tourism Management*. — 2015. — Vol. 46. — Pp. 322–335.
56. *Vile J. L. nnd Gillard J. W., Harper P. R., Knight V. A.* Predicting Ambulance Demand Using Singular Spectrum Analysis // *Journal of the Operational Research Society*. — 2012. — Vol. 63, no. 11. — Pp. 1556–1565.
57. *Iqelan B. M.* A Singular Spectrum Analysis Technique to Electricity Consumption Forecasting // *International Journal of Engineering Research and Applications*. — 2017. — Vol. 7, no. 3. — Pp. 92–100.
58. *Hassani H., Heravi S., Zhigljavsky A.* Forecasting European industrial production with singular spectrum analysis // *International Journal of Forecasting*. — 2009. — Vol. 25, no. 1. — Pp. 103–118.
59. *Hassani H., Heravi S., Zhigljavsky A.* Forecasting UK industrial production with multivariate singular spectrum analysis // *Journal of Forecasting*. — 2013. — Vol. 32, no. 5. — Pp. 395–408.
60. *Broomhead D., King G.* Extracting qualitative dynamics from experimental data // *Physica D*. — 1986. — Vol. 20. — Pp. 217–236.
61. *Fraedrich K.* Estimating dimensions of Weather and Climate Attractors // *Journal of the Atmospheric Sciences*. — 1986. — Vol. 43. — Pp. 419–432.
62. *Colebrook J. M.* Continuous plankton records — zooplankton and environment, northeast Atlantic and North Sea, 1948–1975 // *Oceanologica Acta*. — 1978. — Vol. 1. — Pp. 9–23.
63. *Vautard R., Ghil M.* Singular spectrum analysis in nonlinear dynamics, with applications to paleoclimatic time series // *Physica D*. — 1989. — Vol. 35. — Pp. 395–424.
64. *Vautard R., Yiou P., Ghil M.* Singular-Spectrum Analysis: A toolkit for short, noisy chaotic signals // *Physica D*. — 1992. — Vol. 58. — Pp. 95–126.
65. *Elsner J. B., Tsonis A. A.* Singular Spectrum Analysis: A New Tool in Time Series Analysis. — Plenum Press, 1996.
66. Glavnye komponenty vremennyh ryadov: metod ‘Gusenica’ [Principal Components of Time Series: the ‘Caterpillar’ Method] / Ed. by D. Danilov, A.A. Zhigljavsky. — St.Petersburg University Press, 1997. — In Russian.
67. Faktornyj analiz v neftyanoj geologii: Obzor [Factor Analysis in Petroleum Geology: Review] / M. D. Belonin, I. V. Tatarinov, O. M. Kalinin et al.; Ed. by A.M. Margolin. — Moskow, VIEMS, 1971. — In Russian.
68. *Belonin M. D., Golubeva V. A., Skublov G. T.* Faktornyj analiz v geologii [Factor analysis in geology]. — Moscow: Nedra, 1982. — In Russian.
69. *Buchstaber V.M.* Time series analysis and grassmannians // Applied Problems of Radon Transform / Ed. by S. Gindikin. — Providence: AMS, 1994. — Vol. 162 of *AMS Transactions — Series 2*. — Pp. 1–17.

70. Korobeynikov A., Shlemov A., Usevich K., Golyandina N. — Rssa: A collection of methods for singular spectrum analysis, 2021. — R package version 1.0.4. URL: <http://CRAN.R-project.org/package=Rssa>.
71. Yiou P., Sornette D., Ghil M. Data-adaptive wavelets and multi-scale singular-spectrum analysis // *Physica D*. — 2000. — Vol. 142. — Pp. 254–290.
72. Sanei S., Hassani H. Singular Spectrum Analysis of Biomedical Signals. — CRC press, 2015.
73. Hassani H., Thomakos D. A review on Singular Spectrum Analysis for economic and financial time series // *Statistics and Its Interface*. — 2010. — Vol. 3, no. 3. — Pp. 377–397.
74. Trickett S. F-xy eigenimage noise suppression // *Geophysics*. — 2003. — Vol. 68, no. 2. — Pp. 751–759.
75. Oropeza V. The Singular Spectrum Analysis method and its application to seismic data denoising and reconstruction. — Master Thesis in University of Alberta. — 2010.
76. Basilevsky A., Hum Derek P. J. Karhunen-Loéve Analysis of Historical Time Series With an Application to Plantation Births in Jamaica // *Journal of the American Statistical Association*. — 1979. — Vol. 74. — Pp. 284–290.
77. Efimov V., Galaktionov Yu. About possibility of forecasting of cyclic variability for mammal abundance // *Journal of General Biology*. — 1983. — Vol. 44, no. 3. — Pp. 343–352. — in Russian.
78. Tufts D. W., Kumaresan R., Kirsteins I. Data adaptive signal estimation by singular value decomposition of a data matrix // *Proc. IEEE*. — 1982. — Vol. 70, no. 6. — Pp. 684–685.
79. Ghil M., Vautard R. Interdecadal oscillations and the warming trend in global temperature time series // *Nature*. — 1991. — Vol. 350. — Pp. 324–327.
80. Gantmacher F. R. The Theory of Matrices. — Chelsea Publishing Company, New York 68, 1959.
81. de Prony G. Essai expérimental et analytique sur les lois de la dilatabilité des fluides élastiques et sur celles de la force expansive de la vapeur de l'eau et la vapeur de l'alkool à différentes températures // *J de l'Ecole Polytechnique*. — 1795. — Vol. 1, no. 2. — Pp. 24–76.
82. Zhigljavsky A. Singular spectrum analysis for time series: Introduction to this special issue // *Statistics and Its Interface*. — 2010. — Vol. 3, no. 3. — Pp. 255–258.
83. Singular Spectrum Analysis for Effective Feature Extraction in Hyperspectral Imaging / J. Zabalza, J. Ren, Z. Wang et al. // *IEEE Geoscience and Remote Sensing Letters*. — 2014. — Vol. 11, no. 11. — Pp. 1886–1890.
84. Novel two-dimensional singular spectrum analysis for effective feature extraction and data classification in hyperspectral imaging / J. Zabalza, J. Ren, J. Zheng et al. // *IEEE Geoscience and Remote Sensing Letters*. — 2015. — Vol. 53, no. 8. — Pp. 4418–4433.
85. von Buttlar J., Zscheischler J., Mahecha M. D. An extended approach for spatiotemporal gapfilling: dealing with large and systematic gaps in geoscientific datasets // *Nonlinear Processes in Geophysics*. — 2014. — Vol. 21. — Pp. 203–215.
86. Groth A., Ghil M. Monte Carlo Singular Spectrum Analysis (SSA) revisited: Detecting

- oscillator clusters in multivariate datasets // *Journal of Climate*. — 2015. — Vol. 28, no. 19. — Pp. 7873–7893.
87. Tool wear estimation for different workpiece materials using the same monitoring system / D. R. Salgado, I. Cambero, J. M. H. Olivenza et al. // *Procedia Engineering*. — 2013. — Vol. 63. — Pp. 608–615.
  88. ECDSA Key Extraction from Mobile Devices via Nonintrusive Physical Side Channels / D. Genkin, L. Pachmanov, I. Pipman et al. // Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. — CCS '16. — New York, NY, USA: ACM, 2016. — Pp. 1626–1638.
  89. Economic Cycles and Their Synchronization: A Comparison of Cyclic Modes in Three European Countries / L. Sella, G. Vivaldo, A. Groth, M. Ghil // *Journal of Business Cycle Research*. — 2016. — Vol. 12, no. 1. — Pp. 25–48.
  90. Filho A. S. F., Lima G. A. R. Gap Filling of Precipitation Data by SSA - Singular Spectrum Analysis // *Journal of Physics: Conference Series*. — 2016. — Vol. 759, no. 1. — P. 012085.
  91. Trends in high sea levels of German North Sea gauges compared to regional mean sea level changes / C. Mudersbach, Th. Wahl, I. D. Haigh, J. Jensen // *Continental Shelf Research*. — 2013. — Vol. 65. — Pp. 111–120.
  92. Internal climate memory in observations and models / D. P. Monselesan, T. J. O'Kane, J. S. Risbey, J. Church // *Geophysical Research Letters*. — 2015. — Vol. 42, no. 4. — Pp. 1232–1242.
  93. Pepelyshev A., Zhigljavsky A. SSA analysis and forecasting of records for Earth temperature and ice extents // *Statistics and Its Interface*. — 2017. — Vol. 10, no. 1. — Pp. 151–163.
  94. Tampering Detection in Speech Signals by Semi-Fragile Watermarking Based on Singular-Spectrum Analysis / J. Karnjana, M. Unoki, P. Aimmanee, C. Wutiwiwatchai // Advances in Intelligent Information Hiding and Multimedia Signal Processing: Proceeding of the Twelfth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Nov., 21-23, 2016, Kaohsiung, Taiwan, Volume 1 / Ed. by J.-S. Pan, P.-W. Tsai, H.-C. Huang. — Cham: Springer International Publishing, 2017. — Pp. 131–140.
  95. Hudson I. L., Keatley M. R. Singular Spectrum Analytic (SSA) recombination and reconstruction of flowering: Signatures of climatic impacts // *Environmental Modeling & Assessment*. — 2017. — Feb. — Vol. 22, no. 1. — Pp. 37–52.
  96. Localizing microaneurysms in fundus images through Singular Spectrum Analysis / S. Wang, H. L. Tang, L. I. Al turk et al. // *IEEE Transactions on Biomedical Engineering*. — 2017. — May. — Vol. 64, no. 5. — Pp. 990–1002.
  97. Singular Spectrum Analysis and ARIMA Hybrid Model for Annual Runoff Forecasting / Q. Zhang, B.-D. Wang, B. He et al. // *Water Resources Management*. — 2011. — Vol. 25, no. 11. — Pp. 2683–2703.
  98. A failure time series prediction method based on UML model / W. Xin, L. Chao, X. Weiren, L. Ying // 2015 4th International Conference on Computer Science and Network Technology

- (ICCSNT). — Vol. 01. — 2015. — Pp. 62–70.
99. Lakshmi K., Rao A. R. M., Gopalakrishnan N. Singular spectrum analysis combined with ARMAX model for structural damage detection // *Structural Control and Health Monitoring*. — 2017. — Vol. 24, no. 9. — P. e1960. — STC-15-0200.R3.
  100. Sivapragasam C., Lioung S.-Y., Pasha M.F.K. Rainfall and runoff forecasting with SSA–SVM approach // *Journal of Hydroinformatics*. — 2001. — Vol. 3, no. 3. — Pp. 141–152.
  101. Lisi F., Nicolis O., Sandri M. Combining Singular-Spectrum Analysis and neural networks for time series forecasting // *Neural Processing Letters*. — 1995. — Jul. — Vol. 2, no. 4. — Pp. 6–10.
  102. A hybrid model based on Singular Spectrum Analysis and Support Vector Machines Regression for failure time series prediction / X. Wang, J. Wu, C. Liu et al. // *Quality and Reliability Engineering International*. — 2016. — Vol. 32, no. 8. — Pp. 2717–2738. — QRE-16-0186.R2.
  103. A neuro-fuzzy combination model based on singular spectrum analysis for air transport demand forecasting / Y. Xiao, J. J. Liu, Y. Hu et al. // *Journal of Air Transport Management*. — 2014. — Vol. 39. — Pp. 1–11.
  104. Wu C. L., Chau K. W. Rainfall–runoff modeling using artificial neural network coupled with singular spectrum analysis // *Journal of Hydrology*. — 2011. — Vol. 399, no. 3. — Pp. 394–409.
  105. Unit operational pattern analysis and forecasting using EMD and SSA for industrial systems / Z. Yang, C. Bingham, W.-K. Ling et al. // *Advances in Intelligent Data Analysis XI: 11th International Symposium, IDA 2012, Helsinki, Finland, October 25–27, 2012. Proceedings* / Ed. by J. Hollmén, F. Klawonn, A. Tucker. — Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. — Pp. 416–423.
  106. Andrew A. L. Eigenvectors of certain matrices // *Linear Algebra and its Applications*. — 1973. — Vol. 7, no. 2. — Pp. 151–162.
  107. Usevich K. On signal and extraneous roots in Singular Spectrum Analysis // *Statistics and Its Interface*. — 2010. — Vol. 3, no. 3. — Pp. 281–295.
  108. Roy R., Kailath T. ESPRIT: Estimation of Signal Parameters via Rotational Invariance Techniques // *Acoustics, Speech and Signal Processing, IEEE Transactions on*. — 1989. — Vol. 37. — Pp. 984–995.
  109. Barkhuijsen H, de Beer R., van Ormondt D. Improved algorithm for noniterative time-domain model fitting to exponentially damped magnetic resonance signals // *Journal of Magnetic Resonance*. — 1987. — Vol. 73. — Pp. 553–557.
  110. Algorithm for time-domain NMR data fitting based on total least squares / S. Van Huffel, H. Chen, C. Decanniere, P. van Hecke // *Journal of Magnetic Resonance, Series A*. — 1994. — Vol. 110. — Pp. 228–237.
  111. Kumaresan R., Tufts D. W. Estimating the Angles of Arrival of Multiple Plane Waves // *IEEE Transactions on Aerospace and Electronic Systems*. — 1983. — Vol. AES-19, no. 1. — Pp. 134–139.

112. *Pakula L.* Asymptotic zero distribution of orthogonal polynomials in sinusoidal frequency estimation // *IEEE Transactions on Information Theory*. — 1987. — Vol. 33, no. 4. — Pp. 569–576.
113. *Korobeynikov A.* Computation- and Space-Efficient Implementation of SSA // *Statistics and Its Interface*. — 2010. — Vol. 3, no. 3. — Pp. 357–368.
114. *Abraham B., Ledolter J.* Statistical Methods for Forecasting. — Toronto: Wiley, 1983.
115. *Keeling C. D., Whorf T. P.* Atmospheric CO<sub>2</sub> concentrations — Mauna Loa Observatory, Hawaii, 1959–1997. — Scripps Institution of Oceanography (SIO), University of California, La Jolla, California USA 92093-0220. — 1997.
116. *Van Loan Ch. F.* Generalizing the Singular Value Decomposition // *SIAM Journal on Numerical Analysis*. — 1976. — Vol. 13. — Pp. 76–83.
117. *De Moor B., Golub G.* The Restricted Singular Value Decomposition: Properties and applications // *SIAM Journal on Matrix Analysis and Applications*. — 1991. — Vol. 12, no. 3. — Pp. 401–425.
118. *Paige C., Saunders M.* Towards a generalized singular value decomposition // *SIAM Journal on Numerical Analysis*. — 1981. — Vol. 18. — Pp. 398–405.
119. *Andrews D.F., Herzberg A.M.* Data. A Collection of Problems from Many Fields for the Student and Research Worker. — New York: Springer, 1985.
120. *Hyndman R., Yang Y.* tSDL: Time Series Data Library. — 2018. — URL: <https://github.com/FinYang/tSDL> (online; accessed: 10/02/2023).
121. *Hyvärinen A.* Fast and robust fixed-point algorithms for independent component analysis // *IEEE Transactions on Neural Networks*. — 1999. — may. — Vol. 10, no. 3. — Pp. 626–634.
122. *Hyvärinen A., Karhunen J., Oja E.* Independent component analysis. Adaptive and learning systems for signal processing, communications, and control. — New York: Wiley, 2001.
123. *Huber P. J.* Projection Pursuit // *Annals of Statistics*. — 1985. — Vol. 13, no. 2. — Pp. 435–475.
124. *Hyvärinen A., Oja E.* Independent component analysis: algorithms and applications // *Neural Networks*. — 2000. — Vol. 13, no. 4-5. — Pp. 411 – 430.
125. *Marchini J. L., Heaton C., Riple B. D.* — FASTICA: FastICA Algorithms to Perform ICA and Projection Pursuit, 2021. — URL: <http://CRAN.R-project.org/package=fastICA>.
126. Blind Source Separation of Cardiac Murmurs from Heart Recordings / A. Pietilä, M. El-Segaeier, R. Vigário, E. Pesonen // Independent Component Analysis and Blind Signal Separation / Ed. by J. Rosca, et al. — Springer Berlin / Heidelberg, 2006. — Vol. 3889 of *Lecture Notes in Computer Science*. — Pp. 470–477.
127. Indeterminacy and identifiability of blind identification / L. Tong, R.-W. Liu, V.C. Soon, Y.-F. Huang // *IEEE Transactions on Circuits and Systems*. — 1991. — may. — Vol. 38, no. 5. — Pp. 499–509.
128. A blind source separation technique using second-order statistics / A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, E. Moulines // *IEEE Transactions on Signal Processing*. — 1997. —

- Vol. 45, no. 2. — Pp. 434–444.
129. *Golyandina N., Zhornikova P.* On automated identification in singular spectrum analysis for different types of objects. — 2023. — URL: <https://arxiv.org/abs/2302.08993>.
  130. *Alexandrov Th.* Development of a software package for automatic selection and prediction of additive components of time series within the “Caterpillar”-SSA approach: PhD Thesis / St.Petersburg State University. — St.Petersburg, 2006. — 152 pp., In Russian.
  131. *Fisher R. A.* Tests of significance in Harmonic Analysis // *Proceedings of the Royal Society A.* — 1929. — Vol. 125. — Pp. 54–59.
  132. *Schoellhamer D.H.* Singular spectrum analysis for time series with missing data // *Geophysical Research Letters.* — 2001. — Vol. 28, no. 16. — Pp. 3187–3190.
  133. *Kondrashov D., Ghil M.* Spatio-temporal filling of missing points in geophysical data sets // *Nonlinear Processes in Geophysics.* — 2006. — Vol. 13, no. 2. — Pp. 151–159.
  134. *Beckers J., Rixen M.* EOF calculations and data filling from incomplete oceanographic data sets // *Journal of Atmospheric and Oceanic Technology.* — 2003. — Vol. 20. — Pp. 1839–1856.
  135. *Brown R. G.* Smoothing, forecasting and prediction of discrete time series. — Prentice-Hall Englewood Cliffs, N.J, 1963.
  136. *Chu M. T., Funderlic R. E., Plemmons R. J.* Structured low rank approximation // *Linear Algebra and its Applications.* — 2003. — Vol. 366. — Pp. 157–172.
  137. Exact and Approximate Modeling of Linear Systems / I. Markovsky, J. Willems, S. Van Huffel, B. De Moor. — SIAM, 2006.
  138. *Markovsky I., Usevich K.* Software for Weighted Structured Low-Rank Approximation // *Computational and Applied Mathematics.* — 2014. — Vol. 256. — Pp. 278–292.
  139. *Gillard J., Zhigljavsky A.* Optimization challenges in the structured low rank approximation problem // *Journal of Global Optimization.* — 2013. — Vol. 57, no. 3. — Pp. 733–751.
  140. *Zvonarev N. K.* Search for Weights in the Problem of Finite-Rank Signal Estimation in the Presence of Random Noise // *Vestnik St. Petersburg University, Mathematics.* — 2021. — jul. — Vol. 54, no. 3. — Pp. 236–244.
  141. *Kundur D., Hatzinakos D.* Blind image deconvolution // *IEEE Signal Processing Magazine.* — 1996. — May. — Vol. 13, no. 3. — Pp. 43–64.
  142. *Michailovich O., Tannenbaum A.* Blind deconvolution of medical ultrasound images: A parametric inverse filtering approach // *IEEE Transactions on Image Processing.* — 2007. — Dec. — Vol. 16, no. 12. — Pp. 3005–3019.
  143. *Gillard J., Zhigljavsky A. A.* Stochastic algorithms for solving structured low-rank matrix approximation problems // *Communication in Nonlinear Science and Numerical Simulation.* — 2015. — Vol. 21, no. 1. — Pp. 70–88.
  144. *Allen G. I., Grosenick L., Taylor J.* A generalized least-square matrix decomposition // *Journal of the American Statistical Association.* — 2014. — Vol. 109, no. 505. — Pp. 145–159.
  145. *Shaman P.* An Approximate Inverse for the Covariance Matrix of Moving Average and Autoregressive Processes // *Annals of Statistics.* — 1975. — 03. — Vol. 3, no. 2. — Pp. 532–

- 538.
146. *Siddiqui M. M.* On the Inversion of the Sample Covariance Matrix in a Stationary Autoregressive Process // *The Annals of Mathematical Statistics*. — 1958. — 06. — Vol. 29, no. 2. — Pp. 585–588.
  147. *Jury E. I.* Theory and application of the Z-transform method. — New York, NY: Wiley, 1964.
  148. *Vaughan, S.* A simple test for periodic signals in red noise // *Astronomy & Astrophysics*. — 2005. — Vol. 431, no. 1. — Pp. 391–403.
  149. *Louca S., Doeblei M.* Detecting cyclicity in ecological time series // *Ecology*. — 2015. — Vol. 96, no. 6. — Pp. 1724–1732.
  150. *Grenander U., Szegö G.* Toeplitz Forms and their Applications. — New York: Chelsea, 1984.
  151. *Allen M. R., Smith L. A.* Monte Carlo SSA: Detecting irregular oscillations in the Presence of Colored Noise // *Journal of Climate*. — 1996. — Vol. 9, no. 12. — Pp. 3373–3404.
  152. *Allen M. R., Robertson A. W.* Distinguishing modulated oscillations from coloured noise in multivariate datasets // *Climate Dynamics*. — 1996. — Vol. 12, no. 11. — Pp. 775–784.
  153. *Garnot V. S. F., Groth A., Ghil M.* Coupled Climate-Economic Modes in the Sahel's Interannual Variability // *Ecological Economics*. — 2018. — Vol. 153. — Pp. 111 – 123.
  154. Evidence of deterministic components in the apparent randomness of GRBs: clues of a chaotic dynamic. / G. Greco, R. Rosa, G. Beskin et al. // *Scientific reports*. — 2011. — Vol. 1. — P. 91.
  155. *Jemwa G. T., Aldrich C.* Classification of process dynamics with Monte Carlo singular spectrum analysis // *Computers & Chemical Engineering*. — 2006. — Vol. 30, no. 5. — Pp. 816 – 831.
  156. *Palus M., Novotná D.* Detecting modes with nontrivial dynamics embedded in colored noise: enhanced Monte Carlo SSA and the case of climate oscillations // *Physics Letters A*. — 1998. — Vol. 248, no. 2. — Pp. 191 – 202.
  157. *Palus M., Novotná D.* Enhanced Monte Carlo Singular System Analysis and detection of period 7.8 years oscillatory modes in the monthly NAO index and temperature records // *Nonlinear Processes in Geophysics*. — 2004. — Vol. 11, no. 5/6. — Pp. 721–729.
  158. *Trench W. F.* Spectral decomposition of Kac-Murdock-Szegő matrices. — Unpublished paper from the Selected works of William F. Trench. URL: [https://works.bepress.com/william\\_trench/133/](https://works.bepress.com/william_trench/133/) (online; accessed: 10/02/2023).
  159. Surrogate data for hypothesis testing of physical systems / G. Lancaster, D. Iatsenko, A. Pidde et al. // *Physics Reports*. — 2018. — Vol. 748. — Pp. 1–60.
  160. *Boyarov A.* — Study of statistical properties of Monte Carlo SSA. — Master's thesis, St.Petersburg State University, 2012.
  161. *Golyandina N.* R-scripts for multiple testing in Monte Carlo singular spectrum analysis. — <https://zenodo.org/record/4568495>. — 2021.
  162. *Nekrutkin V.* Perturbation expansions of signal subspaces for long signals // *Statistics and Its Interface*. — 2010. — Vol. 3. — Pp. 297–319.

163. *Björck Å.* Numerical Methods for Least Squares Problems. — SIAM, 1996.
164. *Badeau R., Richard G., David B.* Performance of ESPRIT for Estimating Mixtures of Complex Exponentials Modulated by Polynomials // *IEEE Transactions on Signal Processing*. — 2008. — Vol. 56, no. 2. — Pp. 492–504.
165. *Djermoune E.-H., Tomczak M.* Perturbation Analysis of Subspace-Based Methods in Estimating a Damped Complex Exponential // *IEEE Transactions on Signal Processing*. — 2009. — Vol. 57, no. 11. — Pp. 4558–4563.
166. *Rife D., Boorstyn R.* Single tone parameter estimation from discrete-time observations // *IEEE Trans Inf Theory*. — 1974. — Vol. 20, no. 5. — Pp. 591–598.
167. *Stoica P., Jakobsson A., L. Jian.* Cisoid parameter estimation in the colored noise case: asymptotic Cramer-Rao bound, maximum likelihood, and nonlinear least-squares // *IEEE Transactions on Signal Processing*. — 1997. — Vol. 45, no. 8. — Pp. 2048–2059.
168. Advanced spectral methods for climatic time series / M. Ghil, R. M. Allen, M. D. Dettinger et al. // *Reviews of Geophysics*. — 2002. — Vol. 40, no. 1. — Pp. 1–41.
169. *Weare Bryan C., Nasstrom John S.* Examples of Extended Empirical Orthogonal Function Analyses // *Monthly Weather Review*. — 1982. — Vol. 110, no. 6. — Pp. 481–485.
170. *Broomhead D.S., King G.P.* On the qualitative analysis of experimental dynamical systems // Nonlinear Phenomena and Chaos / Ed. by S. Sarkar. — Bristol: Adam Hilger, 1986. — Pp. 113–144.
171. *Hannachi A., Jolliffe I. T., Stephenson D. B.* Empirical Orthogonal Functions and Related Techniques in Atmospheric Science: A Review // *International Journal of Climatology*. — 2007. — Vol. 27, no. 9. — Pp. 1119–1152.
172. *Hassani H., Mahmoudvand R.* Multivariate singular spectrum analysis: a general view and vector forecasting approach // *International Journal of Energy and Statistics*. — 2013. — Vol. 01, no. 01. — Pp. 55–83.
173. *Golub G. H., Van Loan Ch. F.* Matrix computations (3rd ed.). — Baltimore, MD, USA: Johns Hopkins University Press, 1996.
174. *Mourrain B., Pan V. Y.* Multivariate polynomials, duality, and structured matrices // *Journal of Complexity*. — 2000. — Vol. 16, no. 1. — Pp. 110–180.
175. *Heinig G.* Generalized inverses of Hankel and Toeplitz mosaic matrices // *Linear Algebra and its Applications*. — 1995. — Vol. 216, no. 0. — Pp. 43–59.
176. *Rodríguez-Aragón L. J., Zhigljavsky A.* Singular Spectrum Analysis for Image Processing // *Statistics and Its Interface*. — 2010. — Vol. 3, no. 3. — Pp. 419–426.
177. *Ade F.* Characterization of Textures by 'Eigenfilters' // *Signal Processing*. — 1983. — Vol. 5, no. 5. — Pp. 451–457.
178. *Monadjemi A.* Towards Efficient Texture Classification and Abnormality Detection: Ph.D. thesis / University of Bristol. — 2004.
179. *Trickett S.* F-xy Cadzow Noise Suppression // 78th Annual International Meeting, SEG, Expanded Abstracts. — 2008. — Pp. 2586–2590.

180. *Rouquette S, Najim M.* Estimation of Frequencies and Damping Factors by Two-Dimensional ESPRIT Type Methods // *IEEE Transactions on Signal Processing*. — 2001. — Vol. 49, no. 1. — Pp. 237–245.
181. ESPIRiT — an Eigenvalue Approach to Autocalibrating Parallel MRI: Where SENSE Meets GRAPPA / M. Uecker, P. Lai, M. J. Murphy et al. // *Magnetic Resonance in Medicine*. — 2014. — Vol. 71, no. 3. — Pp. 990–1001.
182. *Korobeynikov A., Larsen R. M., Wu K. J., Yamazaki I.* — svd: Interfaces to various state-of-art SVD and eigensolvers, 2023. — URL: <http://CRAN.R-project.org/package=svd>.
183. Achievements of the Earth orientation parameters prediction comparison campaign / M. Kalarus, H. Schuh, W. Kosek et al. // *Journal of Geodesy*. — 2010. — Vol. 84. — Pp. 587–596.
184. *Malkin Z.* Employing combination procedures to short-time EOP prediction // *Artificial Satellites*. — 2010. — Vol. 45, no. 2. — Pp. 87–93.
185. *Kosek W.* Future improvements in EOP prediction // *Geodesy for Planet Earth*. — Springer, 2012. — Vol. 136. — Pp. 513–520.
186. *Malkin Z.* Free core nutation and geomagnetic jerks // *Journal of Geodynamics*. — 2013. — Vol. 72. — Pp. 53–58.
187. *Miller N.* Forecasting pole motion with SSA // *Izvestiya GAO in Pulkovo*. — 2016. — Vol. 223. — Pp. 119–123. — (in Russian).
188. Polar motion prediction using the combination of SSA and Copula-based analysis / S. Modiri, S. Belda, R. Heinkelmann et al. // *Earth, Planets and Space*. — 2018. — Vol. 70, no. 1. — P. 115.
189. International Earth rotation and reference systems service Earth orientation parameters EOP (IERS) 14 C04. — URL: [https://hpiers.obspm.fr/iers/eop/eopc04/eopc04\\_IAU2000.62-now](https://hpiers.obspm.fr/iers/eop/eopc04/eopc04_IAU2000.62-now).
190. Pulkovo EOP and Reference Systems Analysis Center (PERSAC). — URL: <http://www.gao.spb.ru/english/as/persac/eopc04/>.
191. BULLETIN A — all available versions. — URL: <https://www.iers.org/IERS/EN/Publications/Bulletins/bulletins.html>.
192. *Silverman B. W.* Density estimation for statistics and data analysis. — Routledge, 1998.
193. *Silverman B. W.* Using kernel density estimates to investigate multimodality // *Journal of the Royal Statistical Society: Series B (Methodological)*. — 1981. — sep. — Vol. 43, no. 1. — Pp. 97–99.
194. *Jones M. C., Marron J. S., Sheather S. J.* A brief survey of bandwidth selection for density estimation // *Journal of the American Statistical Association*. — 1996. — Vol. 91, no. 433. — Pp. 401–407.
195. *Good I. J., Gaskins R. A.* Density estimation and bump-hunting by the penalized likelihood method exemplified by scattering and meteorite data // *Journal of the American Statistical Association*. — 1980. — Vol. 75, no. 369. — Pp. 42–56.

196. *Lambert Ph., Eilers P.H.C.* Bayesian density estimation from grouped continuous data // *Computational Statistics & Data Analysis*. — 2009. — Vol. 53, no. 4. — Pp. 1388–1399.
197. Generation and Interpretation of Morphogen Gradients: A Subject Collection from Cold Spring Harbor Perspectives in Biology / Ed. by J. Briscoe, P. A. Lawrence, J.-P. Vincent. — Cold Spring Harbor, N.Y.: Cold Spring Harbor Laboratory Press, 2010.
198. *Houchmandzadeh B, Wieschaus E, Leibler S.* Establishment of developmental precision and proportions in the early Drosophila embryo // *Nature*. — 2002. — Vol. 415. — Pp. 798–802.
199. *Alonso F.J., Castillo J.M.Del, Pintado P.* Application of singular spectrum analysis to the smoothing of raw kinematic signals // *Journal of Biomechanics*. — 2005. — Vol. 38, no. 5. — Pp. 1085–1092.
200. Formation of the bicoid morphogen gradient: an mRNA gradient dictates the protein gradient / A.V. Spirov, K. Fahmy, M. Schneider et al. // *Development*. — 2009. — Vol. 136. — Pp. 605–614.
201. Removal of background signal from *in situ* data on the expression of segmentation genes in Drosophila / E. Myasnikova, M. Samsonova, D. Kosman, J. Reinitz // *Development Genes and Evolution*. — 2005. — Vol. 215. — Pp. 320–326.
202. Analysis of pattern precision shows that Drosophila segmentation develops substantial independence from gradients of maternal gene products / D.M. Holloway, L.G. Harrison, D. Kosman et al. // *Developmental Dynamics*. — 2006. — Vol. 235. — Pp. 2949–2960.
203. Characterization of the *Drosophila* segment determination morphome / S. Surkova, D. Kosman, K. Kozlov et al. // *Developmental Biology*. — 2008. — Vol. 313, no. 2. — Pp. 844–862.
204. Calibrationless parallel imaging reconstruction based on structured low-rank matrix completion / P. J. Shin, P. E. Z. Larson, M. A. Ohliger et al. // *Magnetic Resonance in Medicine*. — 2014. — Vol. 72, no. 4. — Pp. 959–970.
205. A quantitative spatiotemporal atlas of gene expression in the *Drosophila* blastoderm / Ch. Fowlkes, C. L. L. Hendriks, S. V. E. Keränen et al. // *Cell*. — 2008. — Vol. 133, no. 2. — Pp. 364–374.
206. A digital framework to build and visualize and analyze a gene expression atlas with cellular resolution in zebrafish early embryogenesis / C. Castro-González, M.A. Luengo-Oroz, L. Duloquin et al. // *PLoS Computational Biology*. — 2014. — Vol. 10, no. 6. — P. e1003670.
207. Automatic channel unmixing for high-throughput quantitative analysis of fluorescence images / C. L. Luengo Hendriks, S. V. Keränen, M. D. Biggin, D. W. Knowles // *Optics express*. — 2007. — Vol. 15, no. 19. — Pp. 12306–12317.
208. *Shlemov A., Golyandina N., Korobeynikov A. et al.* BioSSA. — 2014. — URL: <http://biossa.github.io/>.
209. *Snyder J. P.* Map projections: A working manual. Geological Survey Bulletin Series. — Geological Survey (U.S.), 1987.
210. *Anjyo K., Lewis J. P., Pighin F.* Scattered Data Interpolation for Computer Graphics // ACM SIGGRAPH 2014 Courses. — SIGGRAPH '14. — New York, NY, USA: ACM, 2014. —

- Pp. 27:1–27:69.
211. *Pion S., Teillaud M.* 3D Triangulations // CGAL User and Reference Manual. — CGAL Editorial Board, 2014.
  212. *Park R. E.* Estimation with Heteroscedastic Error Terms // *Econometrica*. — 1966. — Vol. 34, no. 4. — P. 888.