**Marquette University**

**e-Publications@Marquette**

***Mathematics and Statistical Sciences Faculty Research and Publications/College of Arts and Sciences***

***This paper is NOT THE PUBLISHED VERSION*.**

Access the published version via the link in the citation below.

*Stat*, Vol. 10, No. 1 (December 2021): e330. [DOI](https://doi.org/10.1002/sta4.330). This article is © Wiley and permission has been granted for this version to appear in [e-Publications@Marquette](http://epublications.marquette.edu/). Wiley does not grant permission for this article to be further copied/distributed or hosted elsewhere without the express permission from Wiley.

Functional Singular Spectrum Analysis

Hossein Haghbin

Persian Gulf University

Seyed Morteza Najibi

Lund University

Rahim Mahmoudvand

Bu-Ali Sina University

Jordan Tinka

Marquette University

Mehdi Maadooliat

Marquette University

# Abstract

In this paper, we develop a new extension of the singular spectrum analysis (SSA) called functional SSA to analyze functional time series. The new methodology is constructed by integrating ideas from functional data analysis and univariate SSA. Specifically, we introduce a trajectory operator in the functional world, which is equivalent to the trajectory matrix in the regular SSA. In the regular SSA, one needs to obtain the singular value decomposition (SVD) of the trajectory matrix to decompose a given time series. Since there is no procedure to extract the functional SVD (fSVD) of the trajectory operator, we introduce a computationally tractable algorithm to obtain the fSVD components. The effectiveness of the proposed approach is illustrated by an interesting example of remote sensing data. Also, we develop an efficient and user-friendly R package and a shiny web application to allow interactive exploration of the results.

# Abbreviations

**dFPCA:** dynamic FPCA

**FDA:** functional data analysis

**FPCA:** functional PCA

**FSSA:** functional SSA

**fSVD:** functional SVD

**FTS:** functional time series

**MSSA:** multivariate SSA

**NDVI:** normalized difference vegetation index

**PCA:** principal component analysis

**SSA:** singular spectrum analysis

**SVD:** singular value decomposition.

# 1 INTRODUCTION

One of the popular approaches in the decomposition of time series is accomplished using the rates of change. In this approach, the observed time series is partitioned (decomposed) into informative trends plus potential seasonal (cyclical) and noise (irregular) components. Aligned with this principle, singular spectrum analysis (SSA) is a model-free procedure that is commonly used as a non-parametric technique in analyzing the time series. SSA is intrinsically motivated as an exploratory and model building tool rather than a confirmatory procedure (Golyandina, Nekrutkin, & Zhigljavsky, **2001**). SSA does not require restrictive assumptions such as stationarity, linearity, and normality. It can be used for a wide range of purposes such as trend and periodic component detection and extraction, smoothing, forecasting, change-point detection, gap filling, and causality (see, e.g., Golyandina & Osipov, **2007**; Golyandina et al. **2001**; Kondrashov, Shprits, & Ghil, **2010**; Mahmoudvand & Rodrigues, **2016**; Mohammad & Nishida, **2011**; Moskvina & Zhigljavsky, **2003**; Rodrigues & Mahmoudvand, **2016**).

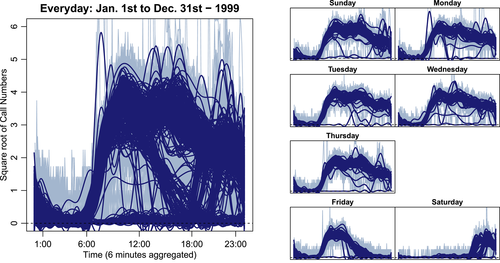
The implementation of SSA over time series is similar to that of principal component analysis (PCA) of multivariate data. In contrast to PCA, which is applied to a data matrix with independent rows, SSA is applied to a time series. It provides a representation of the given time series in terms of eigentriples of a so-called trajectory matrix (Alexandrov, **2009**).

Up to this day, many studies have been published with extensions and applications of SSA. Extension to multivariate SSA (MSSA) is straightforward (Golyandina & Zhigljavsky, **2013**; Golyandina, Korobeynikov, & Zhigljavsky, **2018**; Golyandina, Korobeynikov, Shlemov, & Usevich, **2015**; Hassani & Mahmoudvand, **2013**, **2018**), and an extension of SSA to a two-dimensional setting can be found in Golyandina and Usevich (**2010**), Golyandina et al. (**2015**), and references therein. In the regular SSA, we assume that the observation at each time point is scalar, vector, or array. As a matter of interest, one may consider a series of curves observed over time and use the basics of Hilbert space in the functional data analysis (FDA) framework to introduce the concept of functional SSA (FSSA). Specifically, in this work, we introduce the FSSA for a time series of functions, where each function can be represented using basis expansion models (e.g., B-spline or Fourier).

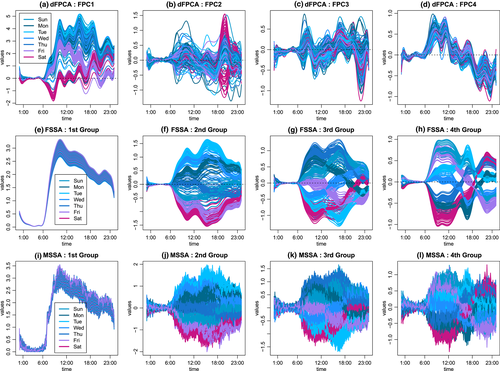
While the research in FDA has grown extensively in recent years, there have been relatively few contributions dealing with functional time series (FTS) (see, e.g., Bosq, **2000**; Hörmann & Kokoszka, **2012**). Although most of the current FTS approaches focus on a parametric fit for inferences and forecasting, there exist some other approaches in the literature that extend functional PCA (FPCA) to incorporate the temporal correlation of FTS. For instance, Hörmann, Kidziński, and Hallin (**2015**) introduced dynamic FPCA (dFPCA) to analyze FTS. This approach assumes the strong assumption of stationarity, which is not generally held in practice. It would be of interest to non-parametrically decompose a nonstationary FTS to reveal the respective trends plus seasonal and irregular components in an appropriate manner. Consistent with this approach, and as a first step, Fraiman, Justel, Liu, and Llop (**2014**) introduced a new concept of trends for the FTS. Furthermore, Hörmann, Kokoszka, and Nisol (**2018**) considered the periodic components for the FTS and derived several procedures to test the periodicity using frequency domain analysis. To the best of our knowledge, existing studies mainly focus on detecting rather than extracting interpretable components.

Since one of the primary missions of SSA is to extract trends and periodic components of a regular (non-functional) time series, it would be rational to establish a similar elegant non-parametric procedure to extract such components for FTS. In this paper, we use the basics of univariate SSA and FDA to develop FSSA. In a nutshell, we develop a matrix operator with functional entities and provide a procedure to obtain the functional singular value decomposition (fSVD) of this operator, which can be seen as a generalization of the regular SVD used in the SSA literature. The new methodology, FSSA, not only can serve as a non-parametric dimension reduction tool to decompose the FTS, it can also be used as a visualization tool to illustrate the concept of seasonality and periodicity in the functional space over time.

In order to depict the idea of our approach and to show its utility, consider the following motivating example involving a real dataset, which is described in detail in the **Supporting Information**. These data provide the intraday number of calls to a call center, during different times of the days for 1 year. The associated 365 curves are represented in an overlapping pattern in Figure **1** (left). In Figure **1** (right), we investigate the pattern among weekdays and weekend days. As we can see, the intraday patterns of weekends (Friday and Saturday) are significantly different from workdays, while workdays seem to have similar patterns. Investigators used variants of FPCA to analyze the call center data in literature (Huang, Shen, & Buja, **2008**; Maadooliat, Huang, & Hu, **2015**; Shen & Huang, **2005**). For illustration purpose, we compare the results of the proposed method (FSSA) and dFPCA on this dataset. Figure **2** (top) presents the projection of the data into the first four FPCs of the dFPCA obtained from the freqdom.fda package in R (Hörmann et al. **2015**). We used seven different colors to differentiate between different days of a week. As one may observe, visually, there is no clear separation in either one of the FPC graphs in the top row. This may not be surprising, as the purpose of PCA of any type is to reduce the dimensionality and not necessarily to decompose the data into regular trends and periodic and irregular components. In contrast, the grouping results that we obtained using the FSSA on the call center data are given in Figure **2** (middle). We also present the result of MSSA obtained from the Rssa package (Golyandina et al. **2015**) in Figure **2** (middle). It can be seen that the functional behavior of 7 days of a week can be well distinguished, visually, using either one of the last two groups (Groups 3 and 4) of the FSSA.

[](https://onlinelibrary.wiley.com/cms/asset/47e154aa-f1aa-485d-89d0-b085e27296d8/sta4330-fig-0001-m.jpg)

**FIGURE 1** The number of calls to a call center between January 1 to December 31 in the year 1999

[](https://onlinelibrary.wiley.com/cms/asset/b6f4b82b-1a99-40b4-aaa1-03ca0c00b089/sta4330-fig-0002-m.jpg)

**FIGURE 2** Four leading dFPCA FPCs associated to the call center data (top) plus the reconstructed series after grouping for FSSA (middle) and MSSA (bottom)

The rest of the paper is organized as follows. Section **2** reviews the core of SSA for completeness. Section **3** presents the theoretical foundations and some properties of the proposed method (FSSA), and Section **4** provides implementation details. Section **5.1** reports simulation results to illustrate the use of the proposed approach in analyzing FTS and to compare it with MSSA and dFPCA. Application to a real data example on remote sensing is given in Section **5.2**. Section **6** provides some discussions and concluding remarks.

# 2 GENERAL SCHEME OF SSA

As we mentioned in Section **1**, SSA can be used for many purposes. However, as we intend to introduce the functional version of SSA for decomposing FTS, we review a general scheme of SSA to perform time series decomposition in this section.

## 2.1 Univariate SSA

Throughout this section, we consider that  are elements of Euclidean space ℝ. Suppose that is a realization of size  from a time series. The basic SSA algorithm consists of four steps: embedding, decomposition, grouping, and reconstruction.

### 2.1.1 Step 1. Embedding

This step generates a multivariate object by tracking a moving window of size *L* over the original time series, where  is called *window length* parameter and . Embedding can be regarded as a mapping operator  that transfers the series **y***N* into a so-called *trajectory* matrix  of dimension , defined by

(1)

where  and , for , are called *lagged vectors*. Note that the trajectory matrix  is a *Hankel* matrix, which means that all the elements along the antidiagonals are equal. Indeed, the embedding operator  is a one-to-one mapping from  into , where  is the set of all  Hankel matrices.

### 2.1.2 Step 2. Decomposition

In this step, the singular value decomposition (SVD) for the trajectory matrix is computed as

(2)

where *r* is the rank of the matrix ,  is the *i*th singular value of ,   and  are the associated (orthonormal) left and right singular vectors, and is called the respective *elementary* matrix.

### 2.1.3 Step 3. Grouping

Consider a partition of the set of indices  into *m* disjoint subsets . For any positive integer , that is, , the matrix is defined as . Thus, by the expansion (2), we have the *grouped matrix decomposition*

(3)

Each group in (**3**) should correspond to a component in a time series decomposition. These components can be considered as trend, cycle, seasonal, noise, and so forth.

### 2.1.4 Step 4. Reconstruction

Finally, the resulting matrices in (3) are transformed back into the form of the original series  by an inverse operator . In order to do this, first, it is necessary that each matrix  be approximated by a matrix in . This approximation is performed optimally in the sense of orthogonal projection of  on  with respect to the *Frobenius* norm. Denote this projection by. It is shown that the projection  is the averaging of the matrix elements over the antidiagonals (where By combining the results of this step and (3), we obtain the final decomposition of the series in the form of

(4)

where , for .

The above algorithm can be extended to perform MSSA for analyzing multivariate time series. The only difference is in defining the trajectory matrix, which can be defined by stacking univariate trajectory matrices horizontally or vertically (Hassani & Mahmoudvand, **2018**).

It is well known that SSA does not require restrictive assumptions; however, it is ideal to have a time series with separable components. Therefore, we present tools to measure the separability of components in the next subsection.

## 2.2 Separability

Let for  be two time series and consider an additive model as  . The series  and  are called separable when each lagged vector of  is orthogonal to the lagged vectors of . To measure the degree of separability between two time series  and , Golyandina et al. (2001) introduced the so-called *w*-correlation

(5)

where  for and . We call two series and *w*-orthogonal if  for appropriate values of *L* (see the next subsection for more details). Note that , , is the reconstructed component produced by the group , and the matrix of is called a *w*-correlation matrix.

## 2.3 Parameter selection

There are two basic parameters in the SSA procedure, window length and grouping parameters. Choosing improper values for these parameters yields an incomplete reconstruction and misleading results in subsequent analysis. In spite of the importance of choosing *L* and grouping parameters for SSA, no ideal solution has been yet proposed. A thorough review of the problem shows that the optimal choice of the parameters depends on the intrinsic structure of the data and the purposes of the study (Golyandina & Zhigljavsky, **2013**; Golyandina et al. **2001**). However, there are several recommendations and rules that work well for a wide range of scenarios. It is recommended to select the window length parameter,  to be a large integer that is a multiple of the periodicities of the time series, but not larger than .

In addition, there are several utilities for effective grouping. These tools include analyzing the periodogram, paired plot of the singular vectors, scree plot of the singular values, and *w*-correlation plot (see Golyandina et al. **2001** for more details).

# 3 THEORETICAL FOUNDATIONS OF FSSA

We start this section with the mathematical foundations that are used to develop the fSSA procedure. Hereafter, we consider  as an FTS of length . This means that each element  belongs to , the space of square integrable real functions defined on the interval . Here, the space ℍ is a Hilbert space, equipped with inner product . For a given positive integer , the space  denotes the Cartesian product of  copies of ℍ; that is, for an element , it has the form , where , and . Then  is a Hilbert space equipped with the inner product . The norms will be denoted by  and  in the spaces  and , respectively. For , and , where  and  are two Hilbert spaces, we define the tensor (outer) product corresponding to the operator  as (, where .

For positive integers  and , we denote as the linear space spanned by operators , specified by , where

(6)

We call an operator  Hankel if  for some  where . The space of such Hankel operators will be denoted by . For two given operators  and  in define

It follows immediately that  defines an inner product on . We will call it the Frobenius inner product of two operators in The associated Frobenius norm is . Before discussing the FSSA algorithm, here we present a lemma that will be used in the last step of the proposed algorithm. We offer proofs of all lemmas, propositions, and theorems in the supporting information.

**Lemma 1.** Let  be elements of the Hilbert space . If , then

for all

## 3.1 FSSA algorithm

For an integer 1 < *L* < *N*/2, let  and define a set of multivariate functional vectors in  by

(7)

where the  denote the functional -lagged vectors. The following algorithm provides the FSSA results in four steps.

### 3.1.1 Step 1. Embedding

Define the operator by

We call  the *trajectory operator*. It is easy to see that , where  is an operator from  to . Evaluating  at a given vector **s** ∈ [0, 1]*k* is same as the matrix product , where  is an  Hankel matrix given by

(9)

***Proposition 1.*** The operator  is a bounded linear operator. If we define given by

(10)

then  is an adjoint operator for .

### 3.1.2 Step 2. Decomposition

In this step, we decompose the trajectory operator  into a set of rank one operators. To this end, we denote the range of  by . Clearly,  is *r*-dimensional, where . Therefore,  is a finite-rank (*r*-dimensional) operator.

***Theorem 1.***Consider the trajectory operator  in (**8**). There exist linearly independent elements  from  and  from  that are orthonormal and

(11)

where the  are non-ascending positive scalars. Then

(12)

***Proposition 2.*** In Theorem **1**, the set  forms a basis system for , and each  can be written as

(13)

One may consider Theorem 1 as the fSVD of . We shall call  the singular value,  the left singular function, and  the right singular vector of the trajectory operator. Therefore, the collection  can be seen as the *i*th eigentriple of , and the right singular vectors, the , can be used to produce paired plots similar to the ones in the SSA literature (Golyandina et al. 2001). Now define , for . One can use (11) and decompose  as

(14)

Clearly, the  s are rank one operators. We refer to the  s as elementary operators.

### 3.1.3 Step 3. Grouping

The grouping step is the procedure of rearranging and partitioning the elementary operators  in (14). To do this, we mimic the approaches used in Step 3 of Section 2 for the univariate SSA and implement the equivalent functional version of those in Haghbin, Najibi, Trinka, and Maadooliat (2019). Note that, in practice, we use a finite set of elementary operators, and one can consider a partition  of the set of indices such that we have the expansion

(15)

### 3.1.4 Step 4. Reconstruction

At this step, for any given , we would like to use  to transform back each operator  in (15) to  and hence construct a functional version of the decomposition given in (4). But since , first, it is necessary to project  to . Note that  is a closed subspace of , therefore by the projection theorem, there exists a unique  such that

To specify , we denote the elements of  and  by  and , respectively. Using Lemma 1, it is easy to extend the diagonal averaging approach in Golyandina et al. (2001) to  and obtain  s as follows:

(16)

where  and  stands for the number of  pairs such that  Denote this projection by , and set . Now we can define  and reconstruct the FTS.

## 3.2 Separability

Let , where , , are FTS. Using a fixed window length  for each series , denote  as a sequence of functional lagged vectors and  as the linear space spanned by . Analogous to univariate SSA, separability of the series  and  is equivalent to , which is the same as  for all . Furthermore, a necessary condition for separability can be defined based on *w*-correlation measure. To do this, consider the weighted inner product of two series  and  as

(17)

where . We call the series  and  *w*-orthogonal if

(18)

***Theorem 2.***If the series  and  are separable, then they are *w*-orthogonal.

Also, to quantify the degrees of separability of two FTS, the functional version of the *w*-correlation measure can be obtained by replacing the new definition of the weighted inner product (**17**) into (**5**).

# 4 IMPLEMENTATION STRATEGY

In practice, functional data are being recorded discretely and then converted to functional objects using proper smoothing techniques. We refer to Ramsay and Silverman (2007) for more details on preprocessing the raw data. Let  be a known basis system (not necessarily orthogonal) of the space . Each functional observation in  can be projected into the subspace , where  can be determined by a variety of techniques (e.g., cross-validation). Therefore, each  is uniquely represented by

(19)

Let us define the quotient sequence, , and remainder sequence, , by

(20)

Note that for any given , one may use (**20**) to determine  and  uniquely, so these sequences are well defined. Now, consider  as the functional vector of length  with all zero functions except the th element, which is .

***Lemma 2.*** The sequence  is a basis system for , where  is the Cartesian product of  copies of .

Now using Lemma 2 and (7), each lag vector  admits a unique representation as

(21)

where  and

(22)

Note that here, . Hence, for any , we have

(23)

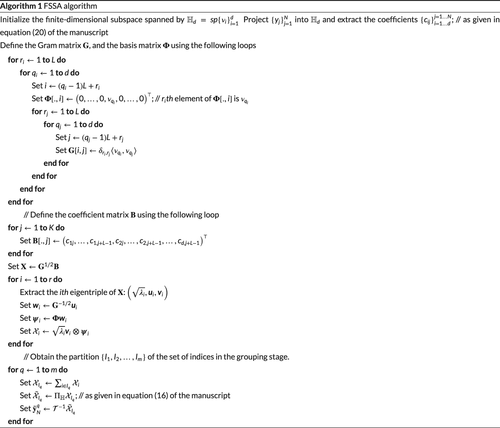
where .

***Theorem 3.*** Suppose , where  is the Gram matrix. Let us denote the collection  as the *i*th eigentriple of . Now define , where . The following holds:

1. is an orthonormal basis for .

***Corollary 1.***The collection of triples  defined in Theorem **3** defines the fSVD of .

Note that the Gram matrix  used in Theorem 3 can be further simplified to , where  if  and zero otherwise. Now we have the recipes to proceed with Algorithm 1, which we use to obtain the FSSA results.

[](https://onlinelibrary.wiley.com/cms/asset/6c3276ae-74ee-4525-9b6f-4351df3ca686/sta4330-gra-0001-m.jpg)

# 5 NUMERICAL STUDY

In this section, first, we present a simulation study to elaborate the use of the FSSA compared with dFPCA and MSSA under different scenarios. To do so, we utilize the implementation of the proposed model that is available as an R package named Rfssa in the CRAN repository (Haghbin et al. **2019**). We also use the freqdom.fda (Hörmann et al. **2015**) and Rssa (Golyandina et al. **2015**) packages to obtain the dFPCA and MSSA results. In the second subsection, we analyze a remote sensing dataset using Rfssa and provide some visualization tools that come in handy in the grouping step.

We developed a shiny app, included in the Rfssa package, also available at **https://fssa.shinyapps.io/fssa/**, to demonstrate and reproduce different aspects of the simulation setup. Furthermore, it can be used to compare the results of dFPCA, MSSA and FSSA on the remote sensing data, call center dataset, or any other FTS provided by the end user.

## 5.1 Simulation study

For the simulation setup, consider the FTS of lengths , and , which are observed in  fixed equidistant discrete points on  from the following model:

(24)

A cubic B-spline basis functions with 15 degrees of freedom is used to convert the  into smooth (continuous) functional curves. In this model,  is considered to be a periodic component defined as

(25)

where  is the model frequency with three different values (, 0.1, and 0.25). Figure S1 depicts a perspective and a heatmap view of an FTS given in (25) for  and .

The  in (**24**) is a stochastic term that is generated under four different settings with an increasing trend in complexity. In the first setting, we consider that  are drawn from an independent Gaussian white noise (GWN) process with zero mean and standard deviation equal to 0.1. It is expected to obtain acceptable performance from FPCA for reconstructing the FTS in the first setting as intuitively FPCA performs well under this ideal framework (see Maadooliat et al. **2015**, for more details).

In the remaining three settings, the  processes are simulated from a functional autoregressive model of Order 1, *FAR*(1), defined by

(26)

where  is an integral operator with a parabolic kernel as follows:

The constant  is chosen such that the Hilbert–Schmidt norm defined by

acquires the values , 0.5, and 0.9 for the remaining three settings, respectively. In these settings, the white noise terms  are considered as independent trajectories of the standard Brownian motion over the interval . It is worth noting that as we increase the Hilbert–Schmidt norm, , in the FAR(1) models, the dependency structure of consecutive FTS gets more twisted, and we expect it would be more challenging to reconstruct the true structures, .

To compare the performance of FSSA and MSSA, we further consider three window length parameters in our simulation setup. For the sake of consistency in all of the reconstruction procedures (dFPCA, MSSA and FSSA), we use the first two leading eigencomponents. The ratio of amplitude of the signal to error is used as the signal-to-noise ratio . As a measure of goodness of fit, we use the root-mean-square error (RMSE) defined as

where  is the FTS reconstructed by each method. We repeat each setting 1000 times and report the mean of the RMSEs in Table 1.

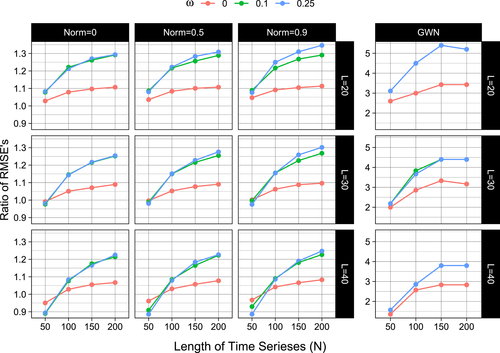
**TABLE 1.**The mean of RMSE for 1000 generations of the simulated model by dFPCA, MSSA, and FSSA approaches

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | **MSSA** |  |  | **FSSA** |  |  |  |
| **Model** | ***ω*** |  | **dFPCA** |  |  |  |  |  |  | **SNR** |
|  | 0.00 | 50 | 0.092 | 0.026 | 0.022 | 0.019 | 0.010 | 0.011 | 0.014 | 241.91 |
|  |  | 100 | 0.088 | 0.024 | 0.020 | 0.018 | 0.008 | 0.007 | 0.007 | 241.57 |
|  |  | 150 | 0.074 | 0.024 | 0.020 | 0.017 | 0.007 | 0.006 | 0.006 | 240.37 |
|  |  | 200 | 0.071 | 0.024 | 0.019 | 0.017 | 0.007 | 0.006 | 0.006 | 241.08 |
|  | 0.10 | 50 | 0.062 | 0.028 | 0.024 | 0.022 | 0.009 | 0.011 | 0.014 | 146.95 |
| Setting 1 |  | 100 | 0.045 | 0.027 | 0.023 | 0.020 | 0.006 | 0.006 | 0.007 | 147.20 |
| GWN |  | 150 | 0.037 | 0.027 | 0.022 | 0.019 | 0.005 | 0.005 | 0.005 | 146.23 |
|  |  | 200 | 0.033 | 0.026 | 0.022 | 0.019 | 0.005 | 0.005 | 0.005 | 146.60 |
|  | 0.25 | 50 | 0.046 | 0.028 | 0.024 | 0.022 | 0.009 | 0.011 | 0.014 | 147.34 |
|  |  | 100 | 0.038 | 0.027 | 0.022 | 0.020 | 0.006 | 0.006 | 0.007 | 145.87 |
|  |  | 150 | 0.032 | 0.027 | 0.022 | 0.019 | 0.005 | 0.005 | 0.005 | 147.35 |
|  |  | 200 | 0.030 | 0.026 | 0.022 | 0.019 | 0.005 | 0.005 | 0.005 | 147.07 |
|  | 0.00 | 50 | 1.008 | 0.251 | 0.246 | 0.270 | 0.244 | 0.248 | 0.284 | 7.62 |
|  |  | 100 | 0.993 | 0.205 | 0.185 | 0.179 | 0.190 | 0.176 | 0.174 | 5.73 |
|  |  | 150 | 0.974 | 0.192 | 0.166 | 0.153 | 0.175 | 0.155 | 0.145 | 5.76 |
|  |  | 200 | 0.960 | 0.186 | 0.158 | 0.143 | 0.168 | 0.145 | 0.134 | 5.68 |
|  | 0.10 | 50 | 0.636 | 0.220 | 0.211 | 0.235 | 0.204 | 0.216 | 0.264 | 3.34 |
| Setting 2 |  | 100 | 0.633 | 0.193 | 0.165 | 0.154 | 0.158 | 0.144 | 0.143 | 4.04 |
| ◂◽:▸=0 |  | 150 | 0.633 | 0.188 | 0.158 | 0.141 | 0.149 | 0.130 | 0.120 | 4.39 |
|  |  | 200 | 0.632 | 0.186 | 0.154 | 0.136 | 0.144 | 0.123 | 0.112 | 3.66 |
|  | 0.25 | 50 | 0.636 | 0.223 | 0.214 | 0.235 | 0.206 | 0.218 | 0.263 | 3.44 |
|  |  | 100 | 0.633 | 0.194 | 0.167 | 0.156 | 0.160 | 0.146 | 0.144 | 4.33 |
|  |  | 150 | 0.633 | 0.188 | 0.157 | 0.140 | 0.148 | 0.129 | 0.120 | 3.92 |
|  |  | 200 | 0.632 | 0.185 | 0.153 | 0.136 | 0.143 | 0.122 | 0.111 | 4.25 |
|  | 0.00 | 50 | 0.841 | 0.316 | 0.310 | 0.350 | 0.305 | 0.311 | 0.364 | 7.20 |
|  |  | 100 | 0.818 | 0.271 | 0.240 | 0.231 | 0.250 | 0.228 | 0.224 | 5.36 |
|  |  | 150 | 0.806 | 0.262 | 0.223 | 0.203 | 0.238 | 0.207 | 0.192 | 5.67 |
|  |  | 200 | 0.801 | 0.258 | 0.216 | 0.194 | 0.233 | 0.198 | 0.180 | 5.58 |
|  | 0.10 | 50 | 0.696 | 0.277 | 0.267 | 0.300 | 0.255 | 0.270 | 0.330 | 3.27 |
| Setting 3 |  | 100 | 0.709 | 0.241 | 0.207 | 0.193 | 0.198 | 0.180 | 0.178 | 4.04 |
|  |  | 150 | 0.710 | 0.235 | 0.197 | 0.176 | 0.187 | 0.162 | 0.151 | 4.23 |
|  |  | 200 | 0.710 | 0.232 | 0.192 | 0.170 | 0.180 | 0.153 | 0.139 | 3.48 |
|  | 0.25 | 50 | 0.707 | 0.214 | 0.205 | 0.225 | 0.198 | 0.209 | 0.254 | 2.93 |
|  |  | 100 | 0.707 | 0.187 | 0.160 | 0.149 | 0.153 | 0.139 | 0.138 | 4.20 |
|  |  | 150 | 0.705 | 0.181 | 0.151 | 0.135 | 0.141 | 0.123 | 0.114 | 3.54 |
|  |  | 200 | 0.705 | 0.178 | 0.148 | 0.130 | 0.136 | 0.116 | 0.106 | 4.12 |
|  | 0.00 | 50 | 0.926 | 0.486 | 0.477 | 0.535 | 0.464 | 0.476 | 0.553 | 6.29 |
|  |  | 100 | 0.915 | 0.441 | 0.391 | 0.373 | 0.404 | 0.368 | 0.358 | 4.70 |
|  |  | 150 | 0.901 | 0.432 | 0.371 | 0.339 | 0.391 | 0.341 | 0.318 | 5.16 |
|  |  | 200 | 0.823 | 0.432 | 0.363 | 0.326 | 0.388 | 0.331 | 0.301 | 5.02 |
|  | 0.10 | 50 | 0.830 | 0.341 | 0.330 | 0.384 | 0.313 | 0.331 | 0.413 | 2.84 |
| Setting 4 |  | 100 | 0.831 | 0.286 | 0.245 | 0.229 | 0.235 | 0.212 | 0.210 | 3.78 |
|  |  | 150 | 0.832 | 0.279 | 0.233 | 0.208 | 0.220 | 0.190 | 0.176 | 3.63 |
|  |  | 200 | 0.829 | 0.275 | 0.227 | 0.200 | 0.213 | 0.179 | 0.163 | 2.96 |
|  | 0.25 | 50 | 0.829 | 0.210 | 0.200 | 0.226 | 0.195 | 0.205 | 0.255 | 2.28 |
|  |  | 100 | 0.829 | 0.175 | 0.149 | 0.138 | 0.140 | 0.129 | 0.127 | 3.60 |
|  |  | 150 | 0.829 | 0.169 | 0.141 | 0.125 | 0.129 | 0.112 | 0.105 | 2.85 |
|  |  | 200 | 0.833 | 0.167 | 0.138 | 0.121 | 0.124 | 0.106 | 0.097 | 3.40 |

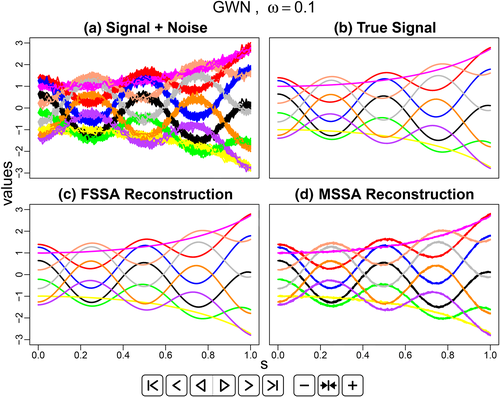
By comparing the results in Table **1**, it can be seen that FSSA outperforms dFPCA in different scenarios. This may not be surprising, as the main task of dFPCA is dimension reduction. Except for the first setting, MSSA also outperforms dFPCA significantly. Furthermore, FSSA performs better than MSSA in most of the cases except in the case where the length of the FTS is small and the window size, , is getting closer to . However, it is clear that FSSA is the optimal method for reconstructing the longer FTS .

For all methods, RMSE decreases as the length of the series increases. For two smaller frequencies (), the average of RMSE increases as the noise structure becomes more complex in Settings 1 through 4, while it decreases for . This might be happening due to the unpredicted cross-correlation of the functional noise structures and the periodic form of FTS.

The efficiency (ratio of RMSE) of MSSA to FSSA for different window lengths , time series lengths , and frequencies is examined in Figure **3**. The overall pattern confirms the improvement in RMSE for FSSA as the time series gets longer (larger ). Overall, as  is increasing, the pattern of the ratio of RMSEs remains unchanged. Although as the window length becomes larger, the improvement either diminishes for longer FTS or disappears (or reverses) for smaller . It is also worth noting that in Setting 1 (GWN), based on the right panel of Figure **3** and Table **1**, the FSSA dominates the other two methods in all combinations of parameters with a better efficiency scale. The visuanimation of Figure **4** shows a sample of simulation setups where subplot (a) shows simulated functional data, subplot (b) shows the true signal, and subplots (c) and (d) show the results of FSSA and MSSA, respectively for varying the noise model and the frequency parameter . We see from Figure **4** that FSSA graphically appears to perform better in extracting the true signal as compared to MSSA.

[](https://onlinelibrary.wiley.com/cms/asset/ad311b23-6c56-4786-81b6-6e0bfe19df32/sta4330-fig-0003-m.jpg)

**FIGURE 3** Ratio of RMSE of MSSA to FSSA in simulation study with 1000 repetitions

[](https://onlinelibrary.wiley.com/cms/asset/dee2824d-798e-4108-b15b-1720fd046632/sta4330-fig-0004-m.jpg)

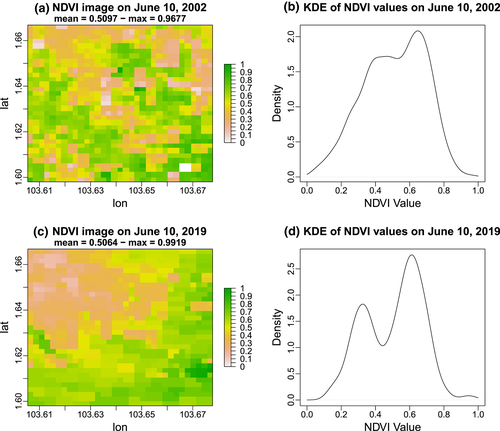
**FIGURE 4** Various simulation set-ups by varying the noise structure and the frequency (*ω*) for fixed  and

## 5.2 Application to remote sensing data

Tracking changes in vegetation over time has become of interest to researchers who want to preserve wildlife. One technique used to track how much vegetation is present in a region is through field surveys. The problem with this technique is that it is especially difficult to implement in low population areas (Panuju & Trisasongko, **2012**). The use of remote sensing data in this context is preferred in order to detect man-made or natural changes in vegetation. One source of remote sensing data is from NASA's MODerate-resolution Imaging Spectroradiometer (MODIS) satellite, which provides images, twice daily, of regions around the globe at varying spatial resolutions (Tuck et al. **2014**). Normalized difference vegetation index (NDVI) is a commonly used pixel-wise index in MODIS satellite images. The NDVI values are bounded between 0 and 1, where index values that are closer to 1 indicate that more vegetation is present and smaller values indicate the absence of vegetation (Tuck et al. **2014**).

Many studies have used the spectral NDVI measure and its variants in order to remotely track changes in vegetation over time (Lambin, **1999**). The temporal average and temporal variability of NDVI images have been used as explanatory variables for the number of different types of vegetation present in many regions (Tuck et al. **2014**). Also, Panuju and Trisasongko (**2012**) used the maximum value of NDVI images taken of the Jambi province in Indonesia to form a time series. The resulting time series was then analyzed using an X12-ARIMA model in order to identify trend and seasonal changes in woody vegetation (Panuju & Trisasongko, **2012**). While these statistics (e.g., maximum and average) have seen some success in tracking the changes of the NDVI images, they may fail to capture the distribution of the vegetation. Therefore, one may seek a more comprehensive measure (e.g., an FTS) to describe the distribution of the NDVI images.

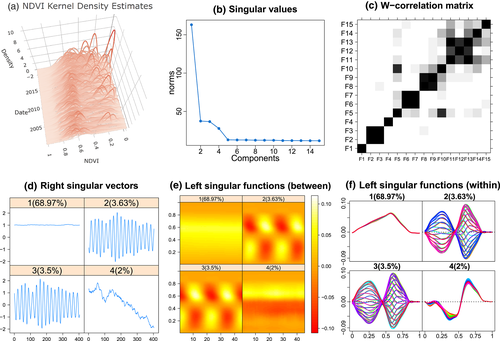
Here we use 448 NDVI images taken in 16-day increments between February 18, 2002, and July 28, 2019. The images have a spatial resolution of 250 m and are of a square region just outside of the city of Jambi, Indonesia between 103.61–103.68°E and 1.60–1.67°S. Figure **5** shows the respective NDVI images taken on June 10, 2002, and June 10, 2019. It is interesting to note that, although the respective NDVI images are not similar, the sample means of the NDVI values are very close (they differ by only about 0.0032 units). As shown in Figure **5**, the kernel density estimates (KDEs) are much more informative in representing the distribution of the NDVI images.

[](https://onlinelibrary.wiley.com/cms/asset/67fa564d-2a87-41ef-9f19-7eb15d782ab5/sta4330-fig-0005-m.jpg)

**FIGURE 5** NDVI images and the associated KDEs on June 10, 2002, and June 10, 2019, in the Jambi region

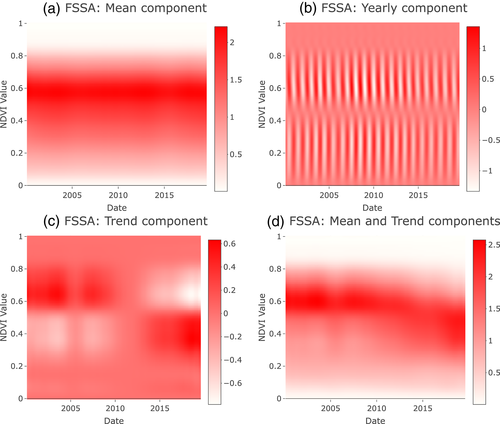
We follow Silverman's rule of thumb (Silverman, **1986**) for bandwidth selection and obtain the KDEs. Then we project the results onto a functional space spanned by a cubic B-spline basis, selected via the GCV criterion. Figure **6a** shows the projected KDEs onto the B-spline space. We pass the results (448 FTS) as input to the FSSA algorithm with the lag , and study the behavior of the NDVI images over almost two decades, using the Rfssa package.

According to the subplots in Figure **6** (b and c: the singular values and *w*-correlation plots), a suitable partition would be grouping the first and fourth components separately, plus the second and third components jointly . The remaining subplots in Figure **6** (d–f: the right singular vectors and left singular functions) indicate that the first component captures mean behavior, the second and third capture annual behavior, and the fourth captures trend.

[](https://onlinelibrary.wiley.com/cms/asset/a68efa0e-9007-44bf-b371-37fec0188185/sta4330-fig-0006-m.jpg)

**FIGURE 6** The KDEs of the 448 NDVI images, plus the FSSA plots for the grouping steps of the NDVI dataset

An important goal of analyzing the NDVI data is to investigate the existence of a temporal trend in the NDVI images over time. It is interesting to see that FSSA can distinguish and separate the overall mean structure (top-left subfigures in Figure **6d–f**) and the trend pattern (bottom-right subfigures in Figure **6d–f**) in two different components (the first and fourth components). The trend component is causing an interesting change-point behavior after almost a decade (bottom-right subfigures in Figures **6d–f** and **7c**). We would like to report that SSA algorithm with the lag  cannot separate this trend structure and the overall mean pattern. Therefore, SSA combines these two (the overall mean and the trend components) into one component (the SSA results are omitted due to space constraints).

[](https://onlinelibrary.wiley.com/cms/asset/9ce6bf5a-a922-4378-ba2f-0f264bb4a7aa/sta4330-fig-0007-m.jpg)

**FIGURE 7** NDVI image reconstructions using (a) the overall mean component (first group), (b) annual harmonic components (second group), (c) trend component (third group) and (d) sum of the trend and overall mean components (first and third groups)

Next, we build the reconstruction of the NDVI images using the grouping suggested by the FSSA exploratory plots, , as shown in Figure **7**. It is clear that Figure **7a** shows the reconstructed overall mean, where it does not change over time. Figure **7b** provides the annual harmonic behavior, and one may observe the change-point behavior after a decade in Figure **7c**. The last subfigure, Figure **7d**, presents the sum of the trend and overall mean components.

It would be of interest to confirm the properties of the reconstructed groups using some rigorous statistical procedures. In order to do this, we provide the *multivariate trace* periodicity test of Hörmann et al. (**2018**) to test for the annual seasonal effect of FTS and a *bootstrap* procedure to test existence of a trend (Grosjean & Ibanez, **2018**) over the time series of the mean of the coefficients associated to the B-spline basis. We obtain the results of these two tests (periodicity and trend) on four sets of FTS (original signal , , , and ), where  represents residual curves obtained via removing the reconstructed FTS by the first *i* groups from the original signal . Table **2** provides the *p* values of the tests for the periodicity and the trend. It is clear that the periodicity test captures the annual pattern for  and  that contains the seasonal components (*p* value = 0). Also the trend test is significant for all of the FTS except , which does not contain the third group (fourth component).

**TABLE 2.** values of the *multivariate trace* periodicity test of Hörmann et al. (2018) and the *bootstrap* trend test (Grosjean & Ibanez, 2018) on four FTS: (, , , and )

|  |  |  |
| --- | --- | --- |
| **FTS** | **Periodicity test** | **Trend test** |
| *yt*(*s*) | 0.00 | 0.02 |
| *R*1 | 0.00 | 0.03 |
| *R*2 | 0.45 | 0.03 |
| *R*3 | 0.30 | 0.07 |

While the FSSA procedure was able to separate out the year-long seasonal pattern, it was also able to detect the less obvious trend component present in the NDVI data, which indicates a loss of vegetation over the previous 20 years. It was determined that between 2001 and 2015, grassland and shrubs in the Jambi region accounted for a large amount of the vegetation lost due to controlled burning for human use (Prasetyo, Dharmawan, Nasdian, & Ramdhoni, **2016**). The region specified in images that we study here is primarily upland agriculture and grass, and it appears to have been a hotspot for controlled fires (Prasetyo et al. **2016**).

# 6 DISCUSSION

In this paper, we constructed the FSSA procedure by incorporating the FDA techniques in basic SSA via MFPCA. The contribution of the proposed model is to provide practitioners with some tools to utilize the advantages of SSA in FTS. Accordingly, the researchers can analyze functional sequences (e.g., time series, longitudinal or spatial data) via FSSA. Alternatively one may approach the problem using MSSA, given that the data points are measured in fixed, regular grid points over time.

As for the ease of use, an efficient and user-friendly R implementation of FSSA is developed in the Rfssa package. Furthermore, a shiny web application is also included in the package, and it is available at **https://fssa.shinyapps.io/fssa/** for reproducing the results of this paper or analyzing any other FTS.

# ACKNOWLEDGEMENTS

This work used the high-performance computing cluster Perè at Marquette University funded by NSF awards OCI-0923037 and CBET-0521602. We thank an associate editor and an anonymous referee for their constructive and thoughtful comments that helped us to improve the paper.

# References

Alexandrov, T. (2009). A method of trend extraction using singular spectrum analysis. RevStat, 7(1), 1– 22.

Bosq, D. (2000). Linear processes in function spaces: Theory and applications, Lecture Notes in Statistics. New York: Springer.

Fraiman, R., Justel, A., Liu, R., & Llop, P. (2014). Detecting trends in time series of functional data: A study of Antarctic climate change. Canadian Journal of Statistics, 42(4), 597– 609.

Golyandina, N., Korobeynikov, A., Shlemov, A., & Usevich, K. (2015). Multivariate and 2D extensions of singular spectrum analysis with the Rssa package. Journal of Statistical Software, Articles, 67(2), 1– 78. https://www.jstatsoft.org/v067/i02

Golyandina, N., Korobeynikov, A., & Zhigljavsky, A. (2018). Singular spectrum analysis with R. Berlin Heidelberg: Springer.

Golyandina, N., Nekrutkin, V., & Zhigljavsky, A. A. (2001). Analysis of time series structure: SSA and related techniques. Boca Raton, Florida: Chapman and Hall/CRC.

Golyandina, N., & Osipov, E. (2007). The “Caterpillar”-SSA method for analysis of time series with missing values. Journal of Statistical Planning and Inference, 137(8), 2642– 2653.

Golyandina, N. E., & Usevich, K. D. (2010). 2D-extension of singular spectrum analysis: Algorithm and elements of theory, Matrix methods: Theory, algorithms and applications: Dedicated to the memory of Gene Golub. Toh Tuck Link, Singapore: World Scientific, pp. 449– 473.

Golyandina, N., & Zhigljavsky, A. (2013). Singular spectrum analysis for time series. Heidelberg: Springer Science & Business Media.

Grosjean, P., & Ibanez, F. (2018). pastecs: Package for analysis of space-time ecological series. https://CRAN.R-project.org/package=pastecs R package version 1.3.21.

Hörmann, S., Kidziński, L., & Hallin, M. (2015). Dynamic functional principal components. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 77(2), 319– 348.

Hörmann, S., & Kokoszka, P. P. (2012). Functional time series, Time series analysis, Handbook of Statistics, Vol.  30. Netherlands: Elsevier B.V., pp. 157– 186.

Hörmann, S., Kokoszka, P., & Nisol, G. (2018). Testing for periodicity in functional time series. The Annals of Statistics, 46(6A), 2960– 2984.

Haghbin, H., Najibi, S. M., Trinka, J., & Maadooliat, M. (2019). Rfssa: Functional singular spectrum analysis. https://CRAN.R-project.org/package=Rfssa R package version 1.0.0.

Hassani, H., & Mahmoudvand, R. (2013). Multivariate singular spectrum analysis: A general view and new vector forecasting approach. International Journal of Energy and Statistics, 01(01), 55– 83. https://doi.org/10.1142/S2335680413500051

Hassani, H., & Mahmoudvand, R. (2018). Singular spectrum analysis: Using R. London: Springer.

Huang, J. Z., Shen, H., & Buja, A. (2008). Functional principal components analysis via penalized rank one approximation. Electronic Journal of Statistics, 2, 678– 695.

Kondrashov, D., Shprits, Y., & Ghil, M. (2010). Gap filling of solar wind data by singular spectrum analysis. Geophysical Research Letters, 37(15), L15101.

Lambin, E. F. (1999). Monitoring forest degradation in tropical regions by remote sensing: Some methodological issues. Global Ecology and Biogeography, 8(3–4), 191– 198.

Maadooliat, M., Huang, J. Z., & Hu, J. (2015). Integrating data transformation in principal components analysis. Journal of Computational and Graphical Statistics, 24(1), 84– 103.

Mahmoudvand, R., & Rodrigues, P. C. (2016). Missing value imputation in time series using singular spectrum analysis. International Journal of Energy and Statistics, 4(01), 1650005.

Mohammad, Y., & Nishida, T. (2011). On comparing SSA-based change point discovery algorithms. In 2011 IEEE/SICE International Symposium on System Integration (SII), IEEE, Kyoto, Japan, pp. 938– 945.

Moskvina, V., & Zhigljavsky, A. (2003). An algorithm based on singular spectrum analysis for change-point detection. Communications in Statistics-Simulation and Computation, 32(2), 319– 352.

Panuju, D. R., & Trisasongko, B. H. (2012). Seasonal pattern of vegetative cover from NDVI time-series. Tropical Forests, ( 255– 268). London: IntechOpen. https://doi.org/10.5772/30344

Prasetyo, L. B., Dharmawan, A. H., Nasdian, F. T., & Ramdhoni, S. (2016). Historical forest fire occurrence analysis in Jambi Province during the period of 2000–2015: Its distribution & land cover trajectories. Procedia Environmental Sciences, 33, 450– 459.

Ramsay, J. O., & Silverman, B. W. (2007). Applied functional data analysis: Methods and case studies. New York: Springer.

Rodrigues, P. C., & Mahmoudvand, R. (2016). Correlation analysis in contaminated data by singular spectrum analysis. Quality and Reliability Engineering International, 32(6), 2127– 2137.

Shen, H., & Huang, J. Z. (2005). Analysis of call centre arrival data using singular value decomposition. Applied Stochastic Models in Business and Industry, 21(3), 251– 263.

Silverman, B. W. (1986). Density estimation for statistics and data analysis. London: Chapman & Hall.

Tuck, S. L., Phillips, H. R. P., Hintzen, R. E., Scharlemann, J. P. W., Purvis, A., & Hudson, L. N.(2014). MODISTools—Downloading and processing MODIS remotely sensed data in R. Ecology and Evolution, 4(24), 4658– 4668. https://onlinelibrary.wiley.com/doi/abs/10.1002/ece3.1273