BLIND SIGNAL SEPARATION OF MIXTURES OF CHAOTIC PROCESSES: A COMPARISON BETWEEN INDEPENDENT COMPONENT ANALYSIS AND STATE SPACE MODELING

ANDREAS GALKA
Department of Neuropediatrics, University of Kiel, 24098 Kiel, Germany
Institute of Experimental and Applied Physics, University of Kiel, 24098 Kiel, Germany

KEVIN K.F. WONG
Massachusetts General Hospital, Harvard Medical School, Boston, MA 02114, USA

ULRICH STEPHANI
Department of Neuropediatrics, University of Kiel, 24098 Kiel, Germany

TOHRU OZAKI
Tohoku University, 28 Kawauchi, Aoba-ku, Sendai 980-8576, Japan

MICHAEL SINIAT'CHKIN
Department of Psychiatry, Psychosomatic Medicine and Psychotherapy, Goethe-University, 60528 Frankfurt am Main, Germany

Received (to be inserted by publisher)

We perform a systematic comparison between different algorithms for solving the Blind Signal Separation problem. In particular, we compare five well-known algorithms for Independent Component Analysis (ICA) with a recently proposed algorithm based on linear state space modeling (IC-LSS). The comparison is based on simulated mixtures of six source signals, five of which are generated by nonlinear deterministic processes evolving on chaotic attractors. The quality of the reconstructed sources is quantified by two measures, one based on a distance measure implemented by a Frobenius norm, and one based on residual mutual information. We find that the IC-LSS modeling algorithm offers several advantages over the ICA algorithms: it succeeds in unmixing gaussian sources, on short time series it performs, on average, better than static ICA algorithms, it does not try to remove coincidental dependencies resulting from finite data set size, and it shows the potential to reconstruct the sources even in the case of non-invertible mixing. As expected, for the case of non-gaussian sources, invertible mixing and sufficient time series length, the ICA algorithms typically outperform IC-LSS modeling.

Keywords: Blind Signal Separation, Independent Component Analysis, State Space Modeling

1. Introduction
The analysis of multivariate time series is often based on latent variable models [Bartholomew, 1987; West et al., 1999]. A specific choice of latent variables is given by the concept of independent components, i.e., a set of latent variables which are statistically independent. At each point of time the data vector is assumed
as being generated by the multiplication of a mixing matrix with an unknown vector of independent components. If also the mixing matrix is unknown, this represents a case of Blind Signal Separation (BSS). In this paper, we limit our scope to instantaneous mixing and do not consider the more general case of convolutive mixing.

Since the 1990s, numerous algorithms have been proposed for the purpose of estimating the mixing matrix and thereby the independent components for a given multivariate time series [Jutten & Hérault, 1991; Comon, 1994; Hyvärinen et al., 2001; Comon & Jutten, 2010]; usually these algorithms are summarised under the title Independent Component Analysis (ICA). ICA algorithms have been found to be useful in a wide variety of fields, such as geophysics, neuroscience, telecommunications and audio signal processing [Hyvärinen et al., 2001; Comon & Jutten, 2010].

Like the older latent variable models of Principal Component Analysis (PCA) and Factor Analysis (FA) [Basilevsky, 1994], most ICA algorithms ignore the temporal ordering of the data; as a consequence, it becomes impossible to separate components with gaussian distributions. We will call these algorithms static algorithms. However, some authors [Ziehe & Müller, 1998] have proposed generalizations of PCA, FA and static ICA, based on inclusion of temporal information; these algorithms will be called dynamic algorithms.

A well-known class of general models for time series which is based both on latent variables and on temporal information, is given by linear state space (LSS) models [Aoki, 1987; Durbin & Koopman, 2001; Ozaki, 2012]. LSS models, as a generalization of autoregressive moving-average (ARMA) models [Box & Jenkins, 1976], are based on optimal prediction of the time series data.

Recently, a special class of LSS models has been introduced especially for the purpose of estimating independent components [Galka et al., 2010]; we will refer to these models as IC-LSS models. The class of IC-LSS models can be regarded as belonging to the wider class of Dynamic Factor Analysis models [Molenaar, 1985]. Previous work has shown that approaching BSS problems by IC-LSS modeling offers certain advantages over ICA, such as the potential to estimate a number of independent components that is larger than the available number of mixtures (this represents a case of non-invertible mixing), the ability to separate several components with gaussian distributions and the possibility to include certain types of prior knowledge.

In this paper we present the results of a systematic comparison of IC-LSS modeling with a set of commonly employed ICA algorithms. In particular, we intend to discuss some strong and weak points of these different approaches to the BSS problem, and to attract attention to the potential that state space methods have to offer for further work in this field. The comparison is entirely based on simulated mixtures, such that the correct independent sources exist and are available for quantifying the quality of reconstructed sources. A number of aspects will be considered, such as

- accuracy of reconstructed independent sources;
- amount of residual mutual information;
- dependence of performance on time series length;
- dependence of performance on distribution (gaussian or non-gaussian).

Some further aspects which are specific to the state space approach, will also be investigated:

- dependence of performance on number of mixtures;
- effect of inclusion of moving-average (MA) terms;
- effect of application of an additional nonlinear observation function;
- performance of an additional smoother, as compared to the original Kalman filter.

The structure of this paper is as follows. In section 2 we will briefly comment on the history of the work on dynamic latent variable models. In section 3 we will first discuss ICA and state space modeling within the common framework of linear observation models, then we will review the state space approach to blind separation of independent source components, and we will discuss the practical fitting of these models. In section 4 we will discuss the design of the simulation and the definitions of the measures for quantifying the results of the simulation study. In section 5 the results of the simulation study themselves will be presented. Finally, in section 6 the results will be discussed and concluding remarks will be provided.
2. Dynamic latent variable models

The concept of modeling a univariate time series by a decomposition into several components has a long history in statistics. A good summary of the earlier work is given in the introduction of Godolphin & Johnson [2003]. As an example for an important application, we mention seasonal adjustment in econometric data; in the 1990s state space methods have been developed for this task [Kitagawa & Gersch, 1996; Kato et al., 1996]. Other authors have developed Bayesian methodology for univariate time series decomposition [West, 1997], which is closely related to state space methods.

In this paper we will employ state space models with block-diagonal state transition matrix for the purpose of a decomposition into independent components. This approach has also been studied by Godolphin & Johnson [2003]; however, we note that block-diagonal state transition matrices have been used much earlier in the system identification community [Pagan, 1975; Engle & Watson, 1981].

A second thread of work that has led towards dynamic latent variable models, has originated in fields such as econometrics and psychology, as a result of the efforts to reinterpret and generalize classical factor analysis within the framework of covariance structure modeling [Jöreskog, 1978]; typically, this work is dealing with multivariate time series. The aim is to find parametric models that are able to describe both instantaneous and lagged covariances. Early papers from this thread include Geweke’s work on Dynamic Factor Analysis [Geweke, 1977], the seminal papers of Engle & Watson [Engle & Watson, 1981; Watson & Engle, 1983] and the work of Molenaar [1985]. Later a number of related approaches, such as Dynamic Component Analysis [Attias & Schreiner, 1998], Independent Factor Analysis [Attias, 1999] and Temporal Factor Analysis [Xu, 2000; Cheung & Xu, 2003a,b; Cheung, 2006], were proposed. Presently, Dynamic Factor Models are still being developed further [Jungbacker et al., 2011].

We note that already Engle & Watson and Molenaar have employed Kalman filtering for the estimation of the latent variables (i.e., the states), but without explicit reference to the BSS framework (which had not yet been formulated). Later, Cheung & Xu [Cheung & Xu, 2003b; Cheung, 2006] independently realized that the BSS problem can be approached by state space modeling and Kalman filtering; these ideas were introduced as a variant of their Temporal Factor Analysis algorithm. Furthermore we mention that a theoretical analysis of Dynamic Factor Analysis in frequency domain has been provided by Picci & Pinzoni [1986].

Zhang & Cichocki [2000] and Waheed & Salem [2005] have employed a linear state space model as a model for a complicated observation process, corresponding to the case known as convolutive mixing in the BSS field; the vector of independent components (corresponding to factors or latent variables) enters this observation process as an unknown input, similar to the unobserved dynamical noise term of classical state space models. This model differs from models based on instantaneous mixing by employing a state space model for the purpose of describing the observation process, while in the framework of instantaneous mixing state space models are usually employed for the purpose of describing how the correlation structure of the data can be generated by filtering white noise.

Recently, Zhang & Hyvärinen [2011] have proposed a state space model for reconstructing a set of mutually dependent components; by assuming the dynamical noise to have non-Gaussian probability distribution (except for at most one component), the model becomes identifiable. They suggest to estimate states and model parameters by a two-step procedure based on the combination of a variant of colored subspace analysis [Theis, 2010] and multichannel blind deconvolution [Cichocki & Amari, 2003], instead of Kalman filtering and maximization of the likelihood. By this work, along with two of its main predecessors [Gómez-Herrero et al., 2008; Haufe et al., 2010], a link to the analysis of the connectivity structure of a set of source components is established.

3. Models for separating independent source components

3.1. Linear instantaneous observation models

Let the multivariate time series be denoted by $y(t) = (y_1(t), \ldots, y_N(t))^T$, $t = 1, \ldots, N_t$, where $N$ denotes the dimension of the data vectors and $N_t$ the length of the time series, i.e., the number of time points at which the data were sampled; the superscript $T$ denotes matrix transposition.
A commonly employed observation model for multivariate data is given by a linear instantaneous observation equation

\[ y(t) = Cx(t) + \epsilon(t) \quad (1) \]

where \( x(t) = (x_1(t), \ldots, x_M(t))^T \), \( t = 1, \ldots, N_t \), denotes a time series of \( M \)-dimensional source vectors, which are not directly observed, \( C \) denotes a constant \( (N \times M) \)-dimensional observation matrix or mixing matrix, and \( \epsilon(t) = (\epsilon_1(t), \ldots, \epsilon_M(t))^T \), \( t = 1, \ldots, N_t \), denotes a time series of \( N \)-dimensional noise vectors, which are also not directly observed.

We shall assume that \( y(t), x(t) \) and \( \epsilon(t) \) are real-valued and have zero mean with respect to time \( t \).

Let \( S_x \) and \( S_\epsilon \) denote the covariance matrices of \( x(t) \) and \( \epsilon(t) \), respectively.

Both Independent Component Analysis (ICA) [Hyvärinen et al., 2001] and Factor Analysis (FA) [Bartholomew, 1987] regard the components of \( x(t) \) as the source components (or independent components, or common factors) to be estimated from given data \( y(t) \), while the elements of \( C, S_x \) and \( S_\epsilon \) represent the model parameters.

### 3.2. Independent Component Analysis (ICA)

Starting from the linear instantaneous observation model of Eq. (1), most ICA algorithms are based on the following set of model assumptions [Hyvärinen et al., 2001]:

- the mixing matrix is square: \( M = N \)
- the probability distributions \( p(x_i) \) of the independent components \( x_i \) are non-gaussian (except for at most one component)
- the components are uncorrelated and standardized to zero-mean and unit-variance: \( S_x = I_M \), \( \langle x(t) \rangle = 0 \) (where \( \langle x \rangle \) denotes the mean of the distribution of \( x \))
- the components are independent: \( p(x_1, \ldots, x_M) = \prod_{i=1}^M p(x_i) \)
- the observation noise is negligible or absent: \( S_\epsilon = 0 \)

Numerous ICA algorithms have been proposed so far; in this paper we select five algorithms for a detailed study:

- **extended InfoMax** [Bell & Sejnowski, 1995; Lee et al., 1999]; we use Revision 1.27 of the MATLAB code which is part of the EEGLab toolbox (http://www.sccn.ucsd.edu/eeglab).
  InfoMax was one of the earliest ICA algorithms to be developed; it is given by an iterative learning rule for the (inverted) mixing matrix. This learning rule was derived from the condition of maximizing the information flow through a network of nonlinear units, but it can also be obtained by a maximum-likelihood argument, applied to a specific type of non-gaussian distributions. The extended version of InfoMax generalizes the learning rule to sub-gaussian sources. In this paper always the extended version will be used.
- **Joint Approximate Diagonalization of Eigen-matrices (JADE)** [Cardoso, 1999]; we use Version 1.8 of the MATLAB code that is available at http://perso.telecom-paristech.fr/~cardoso/Algo/Jade/jadeR.m.
  JADE is based on the idea of quantifying higher-order correlations within the mixtures by a 4th-order cumulant tensor, which can be interpreted as a multivariate version of kurtosis. For 4th-order tensors, the concept of eigenvectors (as known from 2nd-order tensors, i.e., matrices) is generalized to eigen-matrices. JADE performs a minimization of a measure of higher-order correlation by diagonalizing a set of eigen-matrices of the 4th-order cumulant tensor.
- **FastICA** [Hyvärinen, 1999]; we use Version 2.5 of the MATLAB code that is available at http://www.cis.hut.fi/projects/ica/fastica.
  FastICA aims at finding projections of the data with maximum non-gaussianity. The optimization is implemented by a fixed-point iteration applied to an estimator of negentropy, yielding one independent
component at a time. Several independent components can be estimated by performing the fixed-point iteration within a decorrelation scheme, such as Gram-Schmidt orthogonalization.

- **Mutual Information Least-dependent Component Analysis (MILCA)** [Stögbauer et al., 2004]; we use the “copyright 2009” version of the MATLAB and compiled C++ code that is available at http://www.klab.caltech.edu/~kraskov/MILCA. MILCA employs a k-nearest-neighbor estimator of mutual information and aims at directly minimizing the resulting estimate through iterative one-dimensional optimizations of rotation angles in 2-dimensional subspaces. The estimation of mutual information and the update of the estimate after each rotation is implemented in a very efficient way.

- **Second-Order Blind Identification** [Belouchrani et al., 1997]; we use Version 2.01 of the MATLAB implementation provided by Ziehe & Müller as *Time-delayed Decorrelation Separation (TDsep)* [Ziehe & Müller, 1998] (unfortunately, the code is no longer available at the website of the Berlin FIRST institute; however, the main diagonalization routine was provided by Cardoso and can be found at http://perso.telecom-paristech.fr/~cardoso/jointdiag.html). TDsep works by simultaneous diagonalization of a set of time-delayed autocovariance matrices; through a preprocessing step, also the instantaneous autocovariance matrix is diagonalized.

As can be seen from the short summaries given in this list, this set of selected ICA algorithms represents very different approaches to the BSS problem. These algorithms aim at independence by different approaches, such as non-gaussianity, cumulant tensors, residual mutual information and time-delayed autocovariance matrices. Another important difference lies in the approaches to numerical optimization of the various target functions. We emphasize that the first four of these algorithms are static algorithms, i.e. they ignore the temporal ordering of the data; only TDsep is a non-static algorithm, since it employs delayed covariance matrices.

### 3.3. Linear state space (LSS) modeling

State space models are based on time series prediction, therefore they take the temporal ordering of the data into account, including the *direction of time*: i.e., state space modeling is a non-static algorithm. State space models consist of two equations, the observation equation and the dynamical equation. The observation equation is the same as in FA and ICA, given by Eq. (1). The dynamical equation describes the temporal evolution of the state vector \( \mathbf{x}(t) \); in the linear case it is given by a first-order multivariate autoregressive model:

\[
\mathbf{x}(t) = A \mathbf{x}(t-1) + \mathbf{\eta}(t) ;
\]

Here \( \mathbf{\eta}(t) = (\eta_1(t), \ldots, \eta_M(t))^T \), \( t = 1, \ldots, N_t \), denotes a time series of dynamical noise vectors, which are not directly observed. The noise term \( \mathbf{\eta}(t) \) represents dynamical noise (also called driving noise), with covariance matrix \( \mathbf{S}_\eta \), in contrast to the observation noise term \( \mathbf{\epsilon}(t) \). The \((M \times M)\)-dimensional parameter matrix \( A \) represents the *state transition matrix* of the dynamics.

The elements of the four matrices \( A, C, S_\eta \) and \( S_\epsilon \) form the set of main model parameters of the linear state space model, to be summarized by the parameter vector \( \vartheta \). Each of these matrices serves a specific role within the model: \( S_\eta \) describes instantaneous correlations between the components of the state vector \( \mathbf{x}(t) \), while \( A \) describes delayed correlations between the components of \( \mathbf{x}(t) \); \( C \) describes the instantaneous mixing of the components of \( \mathbf{x}(t) \) in the data \( \mathbf{y}(t) \), and \( S_\epsilon \) finally describes additional instantaneous correlations between the components of \( \mathbf{y}(t) \).

### 3.4. Linear state space model for independent components (IC-LSS)

The general linear state space model of Eqs. (1) and (2) can be adapted to the purpose of modeling independent components by choosing a specific structure for the parameter matrices \( A, C, S_\eta \) and \( S_\epsilon \). For
this purpose we choose a common block-diagonal structure for $A$ and $S_\eta$ [Godolphin & Johnson, 2003]:

$$A = \begin{pmatrix} A_1 & 0 & \ldots & 0 \\ 0 & A_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & A_{N_c} \end{pmatrix}, \quad S_\eta = \begin{pmatrix} S_1 & 0 & \ldots & 0 \\ 0 & S_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & S_{N_c} \end{pmatrix}$$

(3)

where $N_c$ denotes the number of blocks; each block will represent one independent component, therefore $N_c$ also denotes the number of independent components. $A_j$ and $S_j$ denote two sets of $(n_j \times n_j)$-dimensional square matrices, where $\sum_{j=1}^{N_c} n_j = M$. Note that only in the case of $n_j = 1$, $j = 1, \ldots, N_c$, the actual state space dimension $M$ will be equal the number of independent components $N_c$; in general we will have $M > N_c$. We emphasize that while each block contributes $n_j$ dimensions to the state space dimension $M$, it contributes only one independent component. Any combination of integer values may be chosen for the $n_j$, but in this paper only the cases $n_j = 1$ and $n_j = 2$ will be chosen.

The common block-diagonal structure of $A$ and $S_\eta$ generates a corresponding partition of the state vector

$$x(t) = (x_1^T(t), x_2^T(t), \ldots, x_{N_c}^T(t))^T$$

(4)

and of the observation matrix

$$C = (C_1, C_2, \ldots, C_{N_c}) \ .$$

(5)

By choosing this structure, we describe the data as a superposition of $N_c$ non-interacting linear stochastic processes, defined by their parameter matrices $A_j, C_j$ and $S_j$. These processes are unable to interact, since both $A$ and $S_\eta$ are chosen as block-diagonal, such that all parameters connecting two different processes vanish. By this model design, the independence assumption of ICA can be implemented indirectly, on the level of the structure of the predictive model, instead directly on the level of the estimated independent source components, as it is explicitly or implicitly done with most ICA algorithms. We remark that the block-diagonal structure proposed here represents a generalization over the Temporal Factor Analysis algorithm [Xu, 2000; Cheung, 2006] where the matrices corresponding to $A$ and $S_\eta$ were defined as diagonal.

It is furthermore assumed that the covariance matrix of the observation noise $S_\eta$ can be chosen as a diagonal matrix with non-zero diagonal elements, which also form part of the set of model parameters.

For the non-interacting stochastic processes, we choose a structure corresponding to ARMA($n_j$, $n_j - 1$) models, where $n_j$ is a positive integer model order; it is essential that these model orders be chosen equal to the block sizes introduced above. For a univariate variable $x_j(t)$, an ARMA($n_j$, $n_j - 1$) model would be given by

$$x_j(t) = \sum_{\tau=1}^{n_j} a_j(\tau)x_j(t-\tau) + \sum_{\tau=0}^{n_j-1} b_j(\tau)\eta_j(t-\tau) \ ,$$

(6)

where $\eta_j(t)$ denotes a driving noise term, assumed to be zero-mean unit-variance gaussian noise. The sets of AR parameters and MA parameters are denoted by $a_j(\tau), \tau = 1, \ldots, n_j$ and by $b_j(\tau), \tau = 0, \ldots, n_j - 1$, respectively. It is advisable to set $b_j(0) = 1$ as a scaling convention for the driving noise term.

Various state space representations of ARMA($n_j$, $n_j - 1$) models exist; here we choose the observer canonical form model [Franklin et al., 1998]. Then the block matrices $A_j$ and $S_j$ of the state space model follow from the AR and MA parameters of the ARMA($n_j$, $n_j - 1$) models according to

$$A_j = \begin{pmatrix} a_j(1) & 1 & 0 & \ldots & 0 \\ a_j(2) & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_j(n_j - 1) & 0 & 0 & \ldots & 1 \\ a_j(n_j) & 0 & 0 & \ldots & 0 \end{pmatrix} \ ,$$

(7)
For together they provide a nonlinear function that maps the real numbers onto itself, without being bounded by that allows us to stay within the realm of linear state space modeling. For this purpose we replace Eq. (1) linear model" discussed by Godolphin & Triantafyllopoulos [2006], but here we prefer an easier alternative data; a nonlinear element needs to be added to the model. This could be done by the "dynamic generalized data deviates from a gaussian distribution, linear stochastic modeling is not sufficient for describing the We need to discuss one generalization of the linear state space model introduced so far. If the distribution of Nonlinear observation functions 3.5.

According to the observer canonical form of state space models, the observation matrix corresponding to Eq. (6) is given by the \((1 \times n_j)\)-dimensional matrix \(C_j = (1 \quad 0 \ldots 0)\). The structure of \(C_j\) illustrates how \(n_j\) state dimensions ultimately represent a single independent component: only the first state dimension is observed, while the remaining \(n_j - 1\) state dimensions remain unobserved; these dimensions are only “technical dimensions” that are necessary for casting ARMA models of order \((n_j, n_j - 1)\) into state space models of order 1.

If these state space models of the individual ARMA\((n_j, n_j - 1)\) processes, for \(j = 1, \ldots, N_c\), are embedded within a single large state space model, with block-diagonal structure, for the prediction of a data vector of dimension \(N\), each of these \((1 \times n_j)\)-dimensional observation matrices needs to be replaced by a \((N \times n_j)\)-dimensional block observation matrix

\[
C_j = \begin{pmatrix}
c_{1j} & 0 & \ldots & 0 \\
c_{2j} & 0 & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
c_{Nj} & 0 & \ldots & 0
\end{pmatrix}.
\]  

Note that in the univariate case \(N = 1\), as discussed by Godolphin & Johnson [2003], it would suffice to use \(C_j = (1 \quad 0 \ldots 0)\) for all \(j\), and to shift the corresponding scaling factor to the dynamical noise term; however, this is no longer possible in the multivariate case where we need a different observation parameter for each combination of a component within the data vector and an (independent) component within the state vector. The set of these observation parameters \(c_{ij}\), \(i = 1, \ldots, N\), \(j = 1, \ldots, N_c\) corresponds to the mixing matrix of the ICA model; the additional observation parameters of value zero just represent the additional state components which are needed due to the reformulation of ARMA\((n_j, n_j - 1)\) models within the standard first-order state space model. These state components are not observed.

The resulting large state space model with block-diagonal structure, as given by Eqs. (3 – 9), will be called the Independent Components Linear State Space (IC-LSS) model. More general state space models may be defined by relaxing the independence constraint, both with respect to the state transition matrix \(A\) and the dynamical noise covariance matrix \(S_\eta\) [Galka et al., 2011].

3.5. Nonlinear observation functions

We need to discuss one generalization of the linear state space model introduced so far. If the distribution of the data deviates from a gaussian distribution, linear stochastic modeling is not sufficient for describing the data; a nonlinear element needs to be added to the model. This could be done by the “dynamic generalized linear model” discussed by Godolphin & Triantafyllopoulos [2006], but here we prefer an easier alternative that allows us to stay within the realm of linear state space modeling. For this purpose we replace Eq. (1) by

\[
y(t) = g(x(t)) + \epsilon(t),
\]

where \(g(.)\) is a monotone nonlinear function [Ozaki & Iino, 2001]. In this paper we use

\[
g(x) = \begin{cases} 
\frac{1}{\gamma} \text{arsinh}(\gamma x) & \text{if } \gamma > 0 \\
x & \text{if } \gamma = 0 \\
\frac{1}{\gamma} \sinh(\gamma x) & \text{if } \gamma < 0
\end{cases}
\]

where \(\gamma = \begin{cases} \gamma_1 & \text{if } x > 0 \\
\gamma_2 & \text{if } x < 0
\end{cases}\)

For \(\gamma = 0\) the linear gaussian case is retrieved. The sinh function and its inverse arsinh were chosen since together they provide a nonlinear function that maps the real numbers onto itself, without being bounded.
or being subject to other constraints; for small argument this function becomes linear. Depending on the sign of the parameter $\gamma$, the distribution of the argument $x$ will be compressed or expanded, such that both sub-gaussian and super-gaussian distributions can be modeled. Furthermore, by allowing $\gamma$ to have different values for positive and negative arguments, it is also possible to model asymmetric distributions. Instead of sinh / arsinh also other nonlinearities, such as 3rd power / 3rd root, could have been chosen.

For practical work, the inverse function of $g(.)$ is applied to the data $y(t)$, and then linear state space modeling is applied to the resulting transformed data. Here it is tacitly assumed that the corresponding distortion of the distribution of $\epsilon(t)$ can be neglected; but preferably we should assume that after applying the nonlinear transformation to the data a representation by a linear observation equation with gaussian observation noise is possible.

This generalization adds two new model parameters $\gamma_1, \gamma_2$ to the vector of model parameters $\vartheta$. An appropriate correction term has to be added to the logarithmic likelihood, corresponding to the standard transformation of probability density functions with respect to coordinate transformations; for details we refer to Galka et al. [2011].

### 3.6. Estimation of states and of model parameters

Given a multivariate time series, we are facing a twofold estimation problem, since both the states and the model parameters are unknown. For a given set of model parameters state estimates may be obtained by Kalman filtering [Kalman, 1960; Sorenson, 1970; Grewal & Andrews, 2001; Molenaar, 1985; Cheung & Xu, 2003b]. Appropriate smoothing will provide improved state estimates; in this paper we employ the Rauch-Tung-Striebel (RTS) smoother [Rauch et al., 1965].

According to the established procedures of state space modeling [Mehra, 1974; Åström, 1980; Otter, 1986; Durbin & Koopman, 2001] model parameters may be estimated by a maximum-likelihood approach. The likelihood of the data with respect to the parameters of the proposed state space model, as given by Eqs. (1) and (2), can be computed by employing the innovation approach [Ozaki, 2012].

The state space model provides predictions of the data; the resulting prediction errors are also known as innovations. If the model is able to remove most correlations from the data, both instantaneous and delayed, and if the dynamics of the underlying process is continuous, the innovations will approximately be distributed as white gaussian noise [Protter, 1990]. Then the likelihood of the time series of innovations is equal to the likelihood of the corresponding original time series, given the set of of model parameters [Kailath, 1968; Galka et al., 2006].

Let $\nu(t), t = 1, \ldots, N_t$, denote the time series of innovations, and $Y$ the set of all available data vectors, $Y = \{y(1), \ldots, y(N)\}$; then the logarithmic likelihood of the data follows as

$$
\log L(Y|\vartheta) = -\frac{1}{2} \left( N_t \log |S_\nu(t)| + \sum_{t=1}^{N_t} \nu^T(t)S_\nu^{-1}(t)\nu(t) + N_t N \log(2\pi) \right),
$$

where $S_\nu(t)$ denotes the covariance matrix of the innovations, which is provided by the Kalman filter. The logarithmic likelihood needs to be maximized by suitable numerical optimization procedures or learning rules; equivalently a corresponding information criterion may be minimised, such as the Akaike Information Criterion (AIC). AIC is defined as $(-2)$ times the logarithmic likelihood, plus a penalty term for model size, which is given by 2 times the number of model parameters [Kuha, 2004].

After final estimates of parameters have been obtained, the RTS smoother is applied, since it further improves state estimates. For the computation of the likelihood by the innovation approach, the smoother plays no role, therefore it is not included in the optimization procedure.

### 3.7. Practical procedure for optimization

We employ two standard algorithms for optimization, the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton algorithm [Dyrholm et al., 2007] and the Nelder-Mead simplex algorithm [Baldick, 2006]; usually the quasi-Newton algorithm provides faster improvement, but in situations where it fails due to numerical problems, the simplex algorithm may succeed, since it does not rely on local gradients.
Our practical experience has shown that simultaneous optimization of all model parameters does not represent the best approach. Rather we choose to iteratively optimize subsets of parameters; as recommended already by Watson & Engle [1983] we also switch between quasi-Newton and simplex optimization steps. As parameter subsets we use the five main groups of parameters \((A; C; S_\eta; S_\epsilon; (\gamma_1, \gamma_2))\), but we define additional subgroups by collecting all model parameters that belong to each data component, for all components within the data vectors, and to each (independent) component, for all components of the IC-LSS model.

The optimization is initialised by fitting a multivariate autoregressive (MAR) model to the data which is then transformed into a IC-LSS model; a detailed description of this approach is given by Galka et al. [2011]. The estimate of the initial state \(x(1)\) is initialised with zeros. As a consequence of this, a transient is allowed for, and the logarithmic likelihood is evaluated only after this transient; a value of 20 time steps is chosen for this transient.

Once the main groups of parameters have been fitted, it is possible to improve also the estimate of the initial state \(x(1)\) via maximum-likelihood, or minimum-AIC, such that the initial state forms a further group of model parameters. For the optimization of \(x(1)\) the full logarithmic likelihood is evaluated, without transient. A similar optimization can be performed for the estimate of the initial state covariance matrix \(S_x(1)\), although it usually offers only a minor improvement of the model.

Alternating optimizations of \(S_x(1)\) and of the other groups of model parameters are iterated, until the optimization process converges. As convergence criterion we choose a change of the logarithmic likelihood, or equivalently the AIC, of less than two units for a complete pass through all optimization steps, two being a natural unit for AIC and related measures.

### 3.8. Some remarks on model design issues

The simplest choice for the model order \(n_j\) of the ARMA\((n_j, n_j - 1)\) processes would be \(n_j = 1\) for all \(j\), giving us a set of AR(1) processes; but this choice would be inappropriate for oscillatory dynamics. The second simplest choice, \(n_j = 2\), gives us a set of ARMA\((2,1)\) processes, and this is the model order which we will use in this paper. However, higher model orders may also be chosen, as will be discussed below. The choice of the \(n_j\) forms part of the model design step.

The AR part of a ARMA\((n_j, n_j - 1)\) process can be represented by \(n_j\) eigenvalues, some of which may form complex-conjugated pairs; each of these pairs represents an oscillatory component of the dynamics. Therefore, in principle the choice of \(n_j\) could be based on observing such peaks in the power spectrum of the data, or on the absence of such peaks. However, this approach would probably fail in cases where the optimal model orders are different for different source components, i.e., where \(n_j\) does depend on the index \(j\). In this case there would be a need for lengthy model comparison procedures. Different models would need to be compared with respect to their prediction performance, e.g. by comparing their likelihood values; model complexity would need to be taken into account as well, which could be done by correction terms, leading to information criteria such as AIC.

A related problem is given by the problem of estimating the correct number of independent source components. In this paper we do not address this problem, instead we regard this number as known a priori. For most ICA algorithms, the standard assumption is that the number of independent source components is equal to the data dimension, \(N = M\). If too small a number of independent source components is assumed, some reconstructed components will still be mixtures of several true components; if too large a number is assumed, there is the risk of creating spurious source components. In principle, also the problem of estimating the correct number of independent source components could be addressed by a model comparison approach, but in this paper we will not address this option further.

In our opinion, for most practical time series analysis problems the notion of the “true” number of independent source components, or of the “true” model order of a predictive model, are of little relevance, the reason being that for many natural systems no true model with simple structure exists; this point represents a major difference between simulated and real-world time series. As an example, the “true” dynamical model of human brain is certainly inaccessible for any practical purpose, and all models which are fitted to multivariate time series recorded from human brain have to be regarded as extremely simplified.
For such data any attempt to estimate “true” model orders or “true” numbers of independent components will be meaningless [Ozaki, 2012].

It has also been argued that for multivariate time series obtained from spatially extended and highly interconnected systems like human brain the assumption of the existence of a meaningful set of independent components itself is questionable (with the exception of the case that solely artefacts of external origin are to be removed). With respect to this doubt, our practical experience has shown that this assumption is a useful “zeroth-order” approximation, in order to obtain a first impression of the properties of the data. In such cases, the application of IC-LSS modeling can be regarded as a generalization of exploratory techniques like PCA and Projection Pursuit.

For practical IC-LSS modeling, an important factor for the choice of model orders and numbers of components will be given by the computational expense of the model fitting step. Typically, real-world data will always be willing to reveal additional components, if a model with an increased number of components is fitted: and it happens frequently that even an information criterion like AIC keeps decreasing upon an increase of the model order, up to very large model orders. Increased model order corresponds to increased state space dimension, which will slow down the Kalman filter and RTS smoother iterations; and it also corresponds to increased number of model parameters, such that the maximum-likelihood model fitting step will consume more computation time.

Another disadvantage of an increased number of model parameters is the increased risk that the maximum-likelihood optimization gets stuck in local maxima, displaying suboptimal model performance. In the numerical results that will be presented below, we will see examples of this problem which we will interprete as resulting from weaknesses of the numerical optimization process.

4. Simulation: Design

The main topic of this paper is the comparison of the IC-LSS modeling algorithm, as presented above, with a number of commonly employed ICA algorithms. We will compare the performance in a simulated BSS task which has been designed such that it fits well most assumptions of typical ICA algorithms, in particular, non-gaussianity of the true source components. The design of our simulation is partly motivated by simulations recently presented by Hyvärinen [2005]; however, instead of linear stochastic discrete-time source processes we choose mainly nonlinear deterministic continuous-time processes. Simulation results for the case of linear stochastic discrete-time source processes have been presented by Galka & Ozaki [2011], as well as some preliminary results for the case of nonlinear deterministic continuous-time processes.

In the context of the current simulation, the trouble with linear stochastic discrete-time source processes is that they would offer our IC-LSS modeling approach a major advantage over all other algorithms, since IC-LSS modeling is itself based on fitting linear stochastic models to the data (i.e., we would be committing a kind of inverse crime). For this reason we prefer to create our simulated data from a class of processes with fundamentally different properties.

4.1. Measures of performance

In order to quantify the performance of the algorithms we need to define suitable measures. In this paper, two measures will be employed.

First, we choose to estimate the mutual information (MI) of the set of reconstructed source components and compare it with the mutual information of the set of true source components; MI is a standard measure of dependency within a set of time series. Note that while we may always compute the MI of a set of reconstructed source components, only in a simulation we can do the same for the true source components.

Although the true source components are chosen to be independent, their MI will be non-zero nevertheless; this is an effect resulting from the presence of coincidental dependencies which are unavoidable in any data set of finite length. For this reason, it would not be appropriate to compare the MI of the set of reconstructed source components with the theoretical value of zero, but it needs to be compared with its correct non-zero value, as computed by the same estimator.

We estimate MI by the kth-nearest-neighbor estimator proposed by Kraskov et al. [2004], with k = 6; we have found this estimator to be convenient especially for estimating MI for time series which have more
than two dimensions.

Our second measure directly compares reconstructed and true source components by forming the $(M \times M)$-dimensional cross-correlation matrix between these two sets of time series. For this purpose, all individual components are standardized to zero mean and unit variance. For perfect reconstruction, we would expect this cross-correlation matrix to have in each row and in each column exactly one element with a value of $\pm 1$, while all other elements should vanish, i.e., we would expect this cross-correlation matrix to be a permutation matrix with an additional binary information corresponding to a “phase flip”. This shape of the cross-correlation matrix corresponds to the residual indeterminacies with respect to polarity and ordering which cannot be resolved by any algorithm for blind signal separation.

The actual reconstruction will not be perfect, therefore the actual cross-correlation matrix will not have exactly this shape, but we can determine the closest permutation matrix, with $\pm 1$ elements, and quantify the difference between these two matrices by the Frobenius norm of the difference matrix. For perfect reconstruction, we would expect this Frobenius norm to be zero.

### 4.2. Simulated source processes

For the source processes to be mixed and subsequently reconstructed, we choose five nonlinear deterministic processes evolving on chaotic attractors:

- Lorenz system (LOR)
- Thomas cyclically symmetric system (THO)
- Hadley circulation system (HAD)
- Hénon-Heiles system (HEH)
- Mackey-Glass differential delay system at a delay of 55 time units (MGL)

These systems were selected from the list of dynamical systems provided by Sprott [2003], where the equations of these systems and the corresponding references can be found. We also keep all model parameters at the “usual parameters” given by Sprott.

The MGL system is numerically integrated by the discretization approach described by Ding et al. [1993]; the other four systems are integrated by the “Local Linearization” method [Ozaki, 1992]. Time discretization is 0.01 time units for LOR, HAD and HEH systems, and 0.1 time units for THO and MGL systems. After integration, the simulated time series are subsampled by factors of 4 (LOR), 12 (THO), 20 (HAD and MGL) and 40 (HEH). These values were chosen since they provide a sufficiently dense sampling of the chaotic trajectories, such that the power spectra of the source time series roughly coincide.

In addition to these five nonlinear deterministic processes, a linear stochastic AR(8) process driven by white gaussian noise is employed as a further source component process; the model parameters of this process are chosen by fitting an AR(8) model to a human electroencephalogram (EEG) time series displaying pronounced alpha activity, sampled at 256 Hz.

The five nonlinear deterministic systems produce time series with broad distribution of power over frequencies; unlike linear AR processes, their dynamics cannot be characterised by a set of constant eigenfrequencies. Furthermore, these time series display sub-gaussian distributions of amplitudes. In this simulation, the AR(8) system represents the only gaussian source; typical static ICA algorithms are able to cope with at most one gaussian source component.

We create simulated time series of length $L = 1024$, $L = 2048$, $L = 4096$ and $L = 8192$. From the sets of state variables of the LOR, THO, HAD and HEH systems, the first variable is chosen (x in the notation used by Sprott [2003]), while the remaining variables are discarded; MGL and AR(8) produce univariate time series by definition. The six source time series are individually standardized to unit variance, then they are instantaneously mixed by a random $(6 \times 6)$-dimensional matrix, drawn from gaussian noise of zero mean and unit variance. A small amount (2%) of gaussian observation noise is added to each mixture.

In this way, a set of 100 simulated time series is created, for each value of time series length $L$, such that each of these time series is created from different realizations of the initial values, of the driving (dynamical) and observation noise terms, and of the mixing matrix; note that only the AR(8) system requires driving noise, since the other systems are deterministic.
In Fig.1 an example of the simulated source components is shown, for the case \( L = 1024 \); the corresponding mixtures are shown in Fig.2.

![Simulated Source Components](image)

**Fig. 1.** Source components created by five deterministic nonlinear processes plus one stochastic linear process. The processes are (from top to bottom): Lorenz system, Thomas cyclically symmetric system, Hadley circulation system, Hénon-Heiles system, Mackey-Glass differential delay system and linear stochastic AR(8) system driven by white gaussian noise.

As mentioned before, the five nonlinear deterministic processes LOR, THO, HAD, HEH and MGL provide non-gaussian time series; we may map these to time series with perfect gaussian distribution by appropriate nonlinear static transforms. In practice this can be done conveniently by employing a reordering algorithm introduced for “amplitude-adjusted surrogate data analysis” [Theiler et al., 1992]: A set of gaussian white noise is reordered such that it matches the rank ordering of the non-gaussian time series. By this approach the temporal dynamics of the simulated time series remains unchanged, but its distribution becomes gaussian. This approach will in effect add some noise to the resulting time series, which is due to the step of replacing the original time series by a randomly chosen set of white noise that is subsequently reordered.

We apply this “gaussianization” approach to all simulated source time series (even to the AR(8) time series, although it will have little effect for this system) and create a second set of mixtures from these gaussianized source time series. For this second set we omit the step of adding gaussian observation noise, since the approach of gaussianization in effect adds a similar amount of observation noise to the resulting mixtures, as already mentioned.

### 4.3. Data analysis

We analyse the simulated time series by IC-LSS modeling, as presented above. The structure of the state space model is chosen as \( 6 \times \text{ARMA}(2,1) \); we also evaluate the simpler case of omitting MA terms, i.e., choosing \( 6 \times \text{AR}(2) \). For each case, we may either include or exclude the use of a nonlinear observation function.

Note that by choosing 6 processes in state space we are assuming that the correct number of sources, \( M = 6 \), is known a priori. We also estimate IC-LSS models for the case of \( N < M \), by simply omitting one or two of the simulated mixtures, i.e., the cases \( N = 5 \) and \( N = 4 \); also in these cases \( M = 6 \) is kept fixed.

Initial parameter estimates are obtained from MAR models, as mentioned above; the MAR model order is chosen as \( p_{\text{MAR}} = 7 \). For the case of \( N < M \) a larger model order needs to be chosen, in order to obtain a sufficient number of eigenvalues in all cases; here we use \( p_{\text{MAR}} = 15 \). The choice of these model orders is of minor importance since the main model fitting step is performed by the subsequent optimization procedure. The observation noise covariance matrix \( S_\epsilon \) is initialized with \( \exp(-10)I_N \), i.e., with a diagonal matrix with a very small positive value on the diagonal.
For optimised models the reconstructed source components are provided by the filtered state estimates; by additional application of the RTS smoother we may obtain smoothed state estimates, which provide us with another set of reconstructed source components. Both sets will be evaluated.

Furthermore, we analyse the simulated time series by applying the five ICA algorithms mentioned above, namely extended InfoMax, FastICA, JADE, MILCA and TDsep. As far as possible, we try to optimize internal parameters and options of these algorithms for the given separation task, otherwise we keep them at the choices and values recommended by the authors of the individual software packages (see above for URLs). In particular, for FastICA, the “deflation” approach with cubic nonlinearity is chosen, and stabilization is switched on; for MILCA, the rectangular estimator of MI with $k = 6$ nearest neighbors is chosen; and for TDsep, joint diagonalization of covariance matrices for lags $0, 1, 2, \ldots, 50$ is chosen. The ICA algorithms cannot be applied to the case of $N < M$.

5. Simulation: Results

The sets of reconstructed source components are evaluated by the two measures of performance introduced above, namely the difference of mutual information for reconstructed and true sources (to be abbreviated as DMI), and the Frobenius norm of the matrix obtained by subtracting the cross-correlation matrix of reconstructed and true sources from the closest permutation matrix, with $\pm 1$ elements (to be abbreviated as FN). The set of results for these two measures of performance can be presented and summarised by scatterplots, histograms and median values. We begin by presenting those results which allow a comparison between ICA and IC-LSS.

5.1. Comparison between ICA and IC-LSS

For each method and each set of 100 mixtures we obtain a set of results that can be displayed as a scatterplot of values of FN versus DMI. Examples are shown in Fig.3 for non-gaussian sources, and in Fig.4 for gaussian sources.

In these figures, we present results for IC-LSS modeling, i.e., Kalman filtering, for simulated time series of length $L = 1024$, $L = 2048$, $L = 4096$ and $L = 8192$; for the case of $L = 8192$ we also present results for the ICA algorithms and for RTS smoothing (based on corresponding IC-LSS models).

These results may be presented in a different way by projecting them onto the FN and DMI axes and forming histograms. As an example, we show in Fig.5 histograms for all seven methods, for analysing time series of length $L = 8192$, created from non-gaussian sources. We omit the corresponding figure for gaussian sources.

The average performance of each method can be summarised by the median value of the distribution.
Fig. 3. Scatterplots of Frobenius norm vs. difference of mutual information, for analysing time series of different length by IC-LSS modeling (Kalman filtering and RTS smoothing, upper two rows of panels), and for analysing time series of length $L = 8192$ by five ICA algorithms (bottom two rows of panels), for non-gaussian sources. Grey symbols in the background represent results for all seven methods for the corresponding value of time series length. All panels share the same scaling of horizontal and vertical axes.

of results. It is advisable to employ the median instead of the arithmetic mean, in order to reduce the influence of outliers. In Fig.5 the median values are denoted by dashed vertical lines, and the values are explicitly given. In Fig.6 we present median values for time series of length $L = 1024$, $L = 2048$, $L = 4096$ and $L = 8192$ and for all seven methods, both for gaussian and non-gaussian sources.
Fig. 4. Scatterplots of Frobenius norm vs. difference of mutual information, for analysing time series of different length by IC-LSS modeling (Kalman filtering and RTS smoothing, upper two rows of panels), and for analysing time series of length $L = 8192$ by five ICA algorithms (bottom two rows of panels), for gaussian sources. Grey symbols in the background represent results for all seven methods for the corresponding value of time series length. All panels share the same scaling of horizontal and vertical axes; note that the scaling of the horizontal axis is different from that used in Fig.3.

5.2. Discussion of results: non-gaussian sources

For non-gaussian sources and $L = 8192$ it can be seen from the histograms, and the median values, that all five ICA algorithms achieve good results, with respect to FN. We mention that for the specific setting of the simulation study described above, any source reconstruction with a FN value of less than 0.7 represents a successful reconstruction, such that all six sources are uniquely reconstructed, while at larger FN at
Fig. 5. Histograms of difference of mutual information (left) and of Frobenius norm (right), for analysing time series of length $L = 8192$ by different methods, for non-gaussian sources. Dashed vertical red lines denote median values of the distributions; the values of the median are given in the upper right corner of each panel.

At least two sources remain mixed. Here the best median FN (0.1857) is obtained for MILCA, and the worst (0.3487) for InfoMax. For all five algorithms we see a tail of outliers towards larger FN. These results are also well illustrated by the scatterplots.

The DMI results reveal that almost all reconstructions provided by the five ICA algorithms show a negative DMI, i.e., their residual mutual information is smaller than that of the true sources. This result has already been briefly described in our earlier paper [Galka & Ozaki, 2011], where we introduced the notion of “over-independent” reconstructed source components. Again there is a tail of outliers, but now towards more negative values, while no outliers towards positive values are found. Inspection of the corresponding scatterplots reveals that the tails of the FN and DMI histograms coincide, such that a curved relationship between DMI and FN results: the smaller the residual mutual information, the larger FN, i.e., the worse the quality of the reconstruction. This can be seen clearly in the scatterplot for $L = 8192$, while for smaller $L$ the pattern only gradually emerges.

For the reconstructions obtained by IC-LSS modeling, i.e., by Kalman filtering (KF) and RTS smoothing (RTSS), a different picture is seen: in the histograms there are peaks around FN=0.3, but the tail towards larger FN is stronger than for the ICA algorithms, such that the median is raised to about FN=0.4. For DMI, we find a roughly symmetrical distribution around zero for KF, with some outliers towards positive values, and a skewed distribution with a tail towards positive values for RTSS. Inspection of the scatterplots again reveals that these tails coincide: the larger the residual mutual information, the worse the quality of the reconstruction, but DMI does not reach as large positive values as reached by the ICA algorithms towards the negative side.
The plots of median values against time series length $L$ show that for all seven methods FN almost always decreases with increasing $L$, as should be expected. But we note the surprising detail that for $L = 1024$ the KF and RTSS reconstructions perform second best among the seven methods, with only TDsep performing slightly better, while for $L = 8192$ all ICA algorithms perform better than KF and RTSS. However, we need to emphasize that these assertions are based only on inspection of the median values, thereby ignoring the full distributions of results.

For DMI, the plots of median values against time series length $L$ are showing less clear tendencies; but it can be seen that for all seven methods DMI starts to converge towards zero when $L$ is increased from 4096 to 8192. We note that for RTSS the values of DMI deviate from zero considerably stronger than for KF; this behaviour is also clearly visible in the scatterplots and the histograms.

### 5.3. Discussion of results: gaussian sources

For gaussian sources and $L = 8192$ it can be seen from the scatterplots and the median values, that for the four static ICA algorithms InfoMax, JADE, FastICA and MILCA values of FN scatter around 2.0 which corresponds to a complete failure of the reconstruction; this is the expected results for static ICA algorithms and gaussian sources. Among the ICA results, only the TDsep algorithm performs well, with a median of FN=0.2150; no outliers are seen.

![Fig. 6. Median values of difference of mutual information (left panels) and of Frobenius norm (right panels), for analysing time series of different length by different methods (see legend in topright panel), for non-gaussian (top panels) and gaussian sources (bottom panels). For the Frobenius norm, Kalman filter and RTS smoother results almost coincide.](image)
As also for the case of non-gaussian sources, negative values are obtained for DMI for almost all reconstructions provided by the ICA algorithms, and this observation also applies to TDsepf; but we note that TDsep has always the least negative median DMI among the ICA algorithms.

For most time series KF and RTSS succeed in reconstructing the gaussian sources well; for FN we find median values of FN=0.1932 (KF) and FN=0.1933 (RTSS), and for DMI we find DMI=0.0058 (KF) and DMI=0.0063 (RTSS). There is a weak tail of outliers towards larger values of FN and larger positive values of DMI.
The plots of median values against time series length \( L \) show that for KF, RTSS and TDsep, FN almost always decreases with increasing \( L \), while for the static ICA algorithms no systematic dependency is found. KF and RTSS perform better than TDsep, but with increasing \( L \) TDsep catches up and reaches almost the same performance as KF and RTSS.

For the ICA algorithms median values of DMI converge to zero for increasing \( L \); especially for the static ICA algorithms this convergence is very pronounced, since they start at low \( L \) with comparatively large negative values. For KF and RTSS, DMI is weakly positive, and decreases from \( L = 1024 \) to \( L = 4092 \), but no longer from \( L = 4092 \) to \( L = 8192 \).

### 5.4. Results for \( N < M \)

We apply IC-LSS modeling to the case that the number of mixtures \( N \), i.e., the dimension of the data vectors, is smaller than the number of sources \( M \); we show results for the cases \( N = 4 \) and \( N = 5 \), while keeping \( M = 6 \). In Fig.7 we show the resulting scatterplots for non-gaussian sources; results for gaussian sources are similar (for KF and RTSS) and therefore omitted. For comparison, scatterplots for the case \( N = 6 \) are also shown; these results have partly already been shown in Fig.3.

In the figure it can be seen that the typical distribution of results for the case \( N = 6 \), a tail displaying a positive correlation between DMI and FN, is also present for \( N = 5 \), such that for a fraction of the time series successful reconstructions have been achieved, as indicated by values of FN below 0.7; this is found for all values of length \( L \). It can also be seen that for \( N = 5 \), there is a set of time series which yield negative DMI scattering around -0.4, while FN assumes values scattering around 1.5; this set forms a cluster that can be clearly distinguished from the main part of the distribution. No corresponding cluster is found for \( N = 6 \), \( L = 8192 \), but we note that for \( N = 6 \), \( L < 8192 \) a few isolated outliers are found at the position of this cluster.

Closer inspection of histograms for the case of non-gaussian sources (not shown) confirms that for \( N = 5 \) distributions of DMI are bimodal, as should be expected for scatterplots displaying two clusters. For the case of gaussian sources, scatterplots (not shown) appear more blurred, and no bimodal distributions are observed for DMI.

For the Kalman filter at \( L = 8192 \) and \( N = 5 \) the median value of FN is 1.2366, for the RTS smoother it is 1.2417. For gaussian sources, the corresponding median values are 1.1032 and 1.0618.

In Fig.7 it can also be seen that for the case \( N = 4 \) the majority of time series yields negative DMI, and that there are essentially no time series that represent successful reconstructions. For the Kalman filter at \( L = 8192 \) and \( N = 4 \) the median value of FN is 1.7405, for the RTS smoother it is 1.7287. For gaussian sources, the corresponding median values are 1.6655 and 1.6398.

### 5.5. Effects of MA terms and of nonlinear observation function

Finally we report the results of comparing IC-LSS modeling with or without moving-average terms, and with or without nonlinear observation function. We limit this investigation to length \( L = 8192 \) and data dimensions \( N = 5 \) and \( N = 6 \). Results for median values are shown in Fig.8; we omit scatterplots and histograms for this investigation.

As can be seen from the figure, the median of FN for the case of excluding both MA terms and nonlinear observation functions, 0.5156, improves to 0.4069, if these generalizations are included (for non-gaussian sources, \( N = 6 \) and KF); MA terms seem to be more important for this improvement than nonlinear observation functions (which actually increase the median of FN somewhat if employed without MA terms). For gaussian sources, the case of including MA terms but excluding nonlinear observation functions achieves the smallest median value of FN. For the case \( N = 5 \), the opposite situation is observed: including both generalizations yields the smallest median value of FN in the case of gaussian sources, but not in the case of non-gaussian sources. In all cases, results for KF and RTSS are very similar. We remark that in the case of gaussian sources we should not expect any improvement from the introduction of nonlinear observation functions, since the data already follows a gaussian distribution, such that nonlinearities should not be required.
Fig. 8. Median values of difference of mutual information (left panels) and of Frobenius norm (right panels), for analysing time series of length $L = 8192$ and dimension $N = 5$ and $N = 6$ by IC-LSS modeling (see legend in bottomleft panel), for non-gaussian (top panels) and gaussian (bottom panels) sources. Horizontal axis corresponds to the four cases of including / excluding moving-average terms (abbreviated by “MA”) and including / excluding nonlinear observation functions (abbreviated by “nonlin.”). Note that for several cases Kalman filter and RTS smoother results almost coincide.

For the case of non-gaussian sources, median values of DMI are found to decrease when the generalizations are included, again with MA terms having a stronger effect than nonlinear observation functions. For the case of gaussian sources, DMI results are very close to zero for $N = 6$, while for $N = 5$ they increase when the generalizations are included. Closer inspection of histograms for the case of gaussian sources (not shown) confirms that all distributions are unimodal, apart from outliers in the tails.

6. Discussion and Conclusion

In this paper we have presented the results of an extensive simulation study of Blind Signal Separation. The design of the simulation was deliberately chosen such that standard ICA algorithms were well suited for reconstructing the source signals. This was done by creating the source signals from a set of six processes five of which represented nonlinear deterministic processes, while the sixth process was a linear stochastic process. The five nonlinear processes produce time series with non-gaussian distributions, so there was only one source with gaussian distribution. Results of a simulation study based on sets of six linear stochastic source processes have been shown elsewhere [Galka & Ozaki, 2011].

As expected, the static ICA algorithms InfoMax, FastICA, JADE and MILCA, and the non-static ICA algorithm TDsep performed well in reconstructing the sources, despite the presence of a small amount of
observational noise. Among these algorithms, MILCA performed best, in terms of the Frobenius norm (FN), and InfoMax worst; this result illustrates the superiority of the elaborate approach to the minimization of mutual information (MI) which forms the core of MILCA. However, in terms of the residual MI, we have found that all five ICA algorithms tend to produce reconstructed sources that contain too little residual MI, i.e., the reconstructed sources suffer from an “over-independency” effect. We interpret this result as the static ICA algorithms and also TDsep failing to distinguish between dependencies introduced by mixing of sources and dependencies arising coincidentally within the set of true source components due to finite data set size. Both types of dependencies will be detected, but not separated, by numerical estimators of MI. Among the ICA algorithms, the “over-independency” effect is strongest for MILCA and weakest (but still present) for TDsep.

MILCA is an example of an algorithm which directly minimizes an estimate of MI; other algorithms maximize non-gaussianity, but it is well known that these two tasks are closely related [Hyvärinen et al., 2001]. As we have shown in this paper, underestimation of MI correlates with larger values of FN, therefore we conclude that “over-independency” leads to distortions of the reconstructed sources.

We have furthermore demonstrated that the performance of IC-LSS modeling is surprisingly good, given the fact that this method is based on the assumption of linear stochastic dynamics, but in our simulation study has been applied to a mixture of source signals generated mostly by nonlinear deterministic processes. In terms of FN, for the majority of simulated time series IC-LSS modeling performs as well as the ICA algorithms; furthermore, there is no “over-independency” effect for sources reconstructed by IC-LSS modeling.

On the other hand, we have found a subset of time series for which the reconstructed sources show larger values of FN, corresponding to a partial or complete failure of the separation task: in contrast to sources with larger FN reconstructed by the ICA algorithms, these cases are characterized by an excess of MI, compared to the true sources. This excess is more pronounced for sources reconstructed by the RTS smoother than for sources reconstructed solely by the Kalman filter. We believe that these failures do not result from a weakness of IC-LSS modeling itself, but only from numerical problems of the maximum-likelihood optimization procedure. In fact, if for one of the high-FN outliers we repeat the optimization procedure, starting from a different initial solution, then frequently we obtain different reconstructed sources with considerably reduced FN. Here, additional work is required, in order to improve the optimization procedures accordingly. A similar remark may also apply to some of the high-FN outliers for the ICA algorithms.

We further note that the relative performance of IC-LSS modeling, when compared to the ICA algorithms, improves for shorter time series length \( L \); at \( L = 1024 \) it performs better than all static ICA algorithms, in terms of the median of the distribution of results, and almost as good as TDsep.

As expected, IC-LSS modeling performs better for sources with gaussian distribution than for non-gaussian sources, while the static ICA algorithms fail completely. TDsep, being a non-static ICA algorithm, can cope with gaussian distributions and performs well, albeit not as well as IC-LSS modeling; however, for increasing \( L \) it approaches the performance of IC-LSS modeling.

Altogether we can therefore conclude that Second-Order Blind Identification (as represented by TDsep in this paper) is a strong competitor to IC-LSS modeling, especially since it is a very fast algorithm, in contrast to the time-consuming optimization procedures of IC-LSS model fitting. Future work should be devoted to reducing the computational time consumption of IC-LSS modeling. On the other hand, it seems that only state space methods like IC-LSS modeling are able to avoid the problem of “over-independent” reconstructed sources.

For the non-invertible case of \( N < M \), we have shown evidence that, in principle, for \( N = 5 \) IC-LSS modeling is capable of reconstructing all \( M = 6 \) sources correctly. Since in practice this reconstruction has succeeded only for a fraction of the time series, we conclude that this task presents a harder challenge for the maximum-likelihood optimization procedure, compared to the case \( N = M \). The state space model itself can easily be shown to be observable, therefore the reconstruction of all six sources should be possible for all time series. On the other hand, for \( N = 4 \) correct reconstruction of all six sources has not been possible for any of the time series, although also this case is observable; and even the case of \( N = 1 \) would still be observable. We assume that, in order to improve the performance of IC-LSS modeling for the
non-invertible case, it may not suffice to improve the optimization procedure further, but also the model structure itself will have to be generalized.

Within the state space framework, we have employed low-order linear stochastic ARMA models as predictors for time series that were generated mostly by strongly nonlinear deterministic processes; the result that for $N = 6$ we have been successful in reconstructing the sources for the majority of time series illustrates the fact that the Kalman filter is sufficiently flexible with respect to considerably misspecified dynamical models. Nevertheless, we would expect that by employing generalized dynamical models, it should be possible to achieve improved performance, even in the case of $N < M$. For example, we could replace ARMA(2,1) models by ARMA($p$, $p - 1$) models with $p > 2$. This step would certainly improve the predictive performance of the dynamical models, since each source process could be approximated with more than just one complex conjugated pair of eigenvalues, as it is given by the case of $p = 2$. However, it would then be a challenge to correctly identify the groups of eigenvalues representing each source process; if each pair of eigenvalues is uniquely assigned to a source process, this problem does not arise.

Finally we have investigated the contributions of MA terms and of nonlinear observation functions to the performance of IC-LSS modeling. Our results give a somewhat inconsistent picture in this respect, but it can be said that on average both generalizations improve the performance, with MA terms having a considerably stronger effect. The weak effect of nonlinear observation functions can be understood as follows. While the distributions of some of the nonlinear source time series are strongly non-gaussian, this non-gaussianity will be much weaker in the simulated data, due to the effect of the Central Limit Theorem. Therefore allowing the data to have non-gaussian distribution should not be expected to contribute crucially to improving the performance of IC-LSS modeling. This would be different, if non-gaussianity were allowed for in the reconstructed source time series themselves, but then we would have to leave the realm of linear state space modeling.

Both MA terms and nonlinear observation functions provide additional freedom for modeling the data, but they do so at the cost of increasing the number of model parameters. The consequence of augmenting the set of model parameters is an increase of the computational time needed for parameter optimization, and also a potential increase of the risk that this optimization provides suboptimal results, due to complicated shape of the dependence of the likelihood function on the parameters. The latter effect may explain our inconsistent results to some extent.

It is not the purpose of this paper to claim that the presented algorithm based on IC-LSS modeling and Kalman filtering could easily replace the available ICA algorithms; rather we have aimed at providing a comparison of some particular strong and weak points of these different approaches to the problem of Blind Signal Separation, and to attract attention to the potential that state space methods have to offer for further work in this field.

We have demonstrated several attractive features of IC-LSS modeling, such as the ability to deal both with gaussian and non-gaussian amplitude distributions, the good performance for short time series and the absence of “over-independency” of the reconstructed source components. The latter feature seems to be a unique characteristic of IC-LSS modeling, resulting from the approach of indirectly imposing the independence constraint through the model structure, as opposed to directly imposing it by minimization of residual mutual information or an equivalent measure. While non-static ICA algorithms, like TDsep, are also able to deal with gaussian distributions, they still suffer from the “over-independency” problem.

A further attractive feature of IC-LSS modeling is given by its potential to reconstruct sources for the non-invertible case of $N < M$ (the “overcomplete basis” case). In principle, state space modeling does not require the observation matrix to be square or to obey some other constraint relating to its dimensions. If the correct model is known, and if observation matrix and state transition matrix fulfill the condition of observability [Kailath, 1980], the Kalman filter can correctly reconstruct the state vector, after an initial transient has died out. We mention that the question of whether it is possible to estimate the correct model itself from given data poses a considerably more difficult problem (known as an identifiability problem) which has been subject of extensive theoretical work [Tong et al., 1991]. However, further discussion of this problem is beyond the scope of this paper.

The IC-LSS model offers an additional advantage over ICA by representing the main frequencies of the data directly as model parameters. Static ICA algorithms ignore the temporal information contained in
the data, and even Second-Order Blind Identification algorithms access this information only indirectly. As a consequence, components with narrow-band power spectrum, such as stochastic or deterministic linear oscillations, can be extracted much better by IC-LSS modeling than by ICA algorithms. This represents an inherent advantage of a linear predictive model.

The strongest advantage of ICA algorithms like InfoMax, FastICA, JADE and TDsep lies in their comparatively low computational time consumption; MILCA consumes somewhat more time, but in most situations still less than full numerical optimization of the likelihood for state space models, which can easily consume hours of CPU time for a single data set of moderate size. Future work should aim at improving the optimization procedures with respect to their time consumption.

Acknowledgments

The work of A. Galka was supported by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) through Sonderforschungsbereich SFB 855 “Biomagnetic Sensing”.

References


REFERENCES


