ELSEVIER

Neurocomputing 50 (2003) 87–103

NEUROCOMPUTING

www.elsevier.com/locate/neucom

Further studies on temporal factor analysis: comparison and Kalman filter-based algorithm

Yiu-ming Cheung[a,*], Lei Xu[b]

[a] Department of Computer Science, Hong Kong Baptist University, Hong Kong
[b] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong

Received 20 October 2000; accepted 23 November 2001

Abstract

A temporal extension of the classical factor analysis (FA) (Proceedings of the Berkeley Symposium on Mathematical Statistics and Probability, May 3, Berkeley, University of California, 1956, pp. 111–150.) has been made under the framework of *temporal Bayesian Ying–Yang* system (Proceedings of the International Conference on Neural Information Processing (ICONIP'98), Vol. 2, 1998, pp. 877–884; IEEE Trans. Signal Process. 48 (7) (2000) 2132 and Proceedings of the 1999 International Joint Conference on Neural Networks, Vol. 2, Washington, DC, July 1999, pp. 1071–1076). This temporal FA (TFA) not only extends the independent component analysis to Gaussian process, but also provides a new way for state-space identification without knowledge of the model parameters. In this paper, we implement the TFA algorithm provided in Xu (1998, 2000), and compare it with non-temporal one to show the importance of considering temporal relationship in factor analysis. Furthermore, we set up a connection between the TFA and traditional filtering problems in control theory, and present an alternative TFA algorithm. This new algorithm estimates the factors (also called *states*) and its variance by Kalman filter as an alternative to the gradient method used in that algorithm of Xu (1998, 2000), resulting in better performance in general. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Temporal Bayesian Ying–Yang system; Temporal factor analysis; Independent component analysis; State-space identification; Kalman filter

1. Introduction

As a tool of data analysis, factor analysis [2] has been extensively used in many fields such as psychology, business fields, social and biological sciences. The factor analysis

[*] Corresponding author.
E-mail addresses: ymc@comp.hkbu.edu.hk (Y.-m. Cheung), lxu@cse.cuhk.edu.hk (L. Xu).

PII: S0925-2312(01)00699-3

formulates multivariate observations as a linear mixture of independently and identically distributed (i.i.d.) unobservable factors. This technique may therefore deteriorate when the factors are time series, rather than i.i.d.

Recently, a temporal extension of factor analysis, named *temporal factor analysis* (TFA), has been proposed under the *temporal Bayesian Ying–Yang* (TBYY) system [5,6]. The TFA describes the relations between an observation $\mathbf{x}_t = [x_t^{(1)}, x_t^{(2)}, \ldots, x_t^{(d)}]^{\mathrm{T}}$ and a factor vector (also called a *state*) $\mathbf{y}_t = [y_t^{(1)}, y_t^{(2)}, \ldots, y_t^{(k)}]^{\mathrm{T}}$ by the following linear state-space equations:

$$\mathbf{y}_t = \mathbf{\Lambda}\mathbf{y}_{t-1} + \boldsymbol{\varepsilon}_t, \tag{1}$$

$$\mathbf{x}_t = \mathbf{A}\mathbf{y}_t + \mathbf{e}_t, \quad t = 1, 2, \ldots, N, \tag{2}$$

where $\mathbf{y}_t$ is Gaussian distributed, $\boldsymbol{\varepsilon}_t$ and $\mathbf{e}_t$ are zero-mean Gaussian white noises with covariances $\mathbf{\Sigma}_{\boldsymbol{\varepsilon}}$ and $\mathbf{\Sigma}_{\mathbf{e}}$, respectively. In the TFA, the probability density function (pdf) of $\mathbf{y}_t$ needs to be component-wise independent. That is,

$$\begin{aligned} p(\mathbf{y}_0) &= \prod_{j=1}^{k} p(y_0^{(j)}) \quad \text{and} \\ p(\mathbf{y}_t|\mathbf{y}_{t-1}) &= \prod_{j=1}^{k} p(y_t^{(j)}|\mathbf{y}_{t-1}) \\ &= \prod_{j=1}^{k} p(y_t^{(j)}|y_{t-1}^{(j)}) \end{aligned} \tag{3}$$

with

$$p(y_t^{(j)}|\mathbf{y}_{t-1}) = p(y_t^{(j)}|y_{t-1}^{(j)}),$$

where $\mathbf{y}_0$ is the initial state. The TFA objective is to estimate the states up to any constant scales through the observations without any knowledge of the model parameters $\{\mathbf{\Lambda}, \mathbf{A}, \mathbf{\Sigma}_{\boldsymbol{\varepsilon}}, \mathbf{\Sigma}_{\mathbf{e}}\}$. Not only does the TFA include the classical FA [2] as a special case, but also extends the independent component analysis to Gaussian process, and provides a new way for state-space identification without knowledge of the model parameters.

Papers [5,6] have proposed a simple TFA algorithm (denoted as *TFA-A* hereafter) through minimization of one Kullback-divergence cost function in the TBYY. The algorithm models the posteriori pdf of $\mathbf{y}_t$ in a parametric form with its parameters, as well as the model parameters, tuned by the stochastic gradient method.

This paper further studies the TFA problem. We implement the TFA-A algorithm, and compare it with non-temporal one to show the importance of considering state temporal relationship in factor analysis. Furthermore, we build a connection between the TFA and traditional filtering problems in control theory, and present an alternative TFA algorithm (denoted as *Alt-TFA-A* hereafter). The new algorithm uses Kalman filter [1,3] to estimate the state and its variance. The experiments have shown that the new algorithm outperforms the TFA-A algorithm in the measure of mean-square-error (MSE) values.

This paper is organized as follows: Section 2 overviews the general TBYY system. In particular, the TFA implementation within this system is described whereby the TFA-A algorithm is introduced. Section 3 shows the advantage of using states' internal temporal dependence in TFA through the comparison of TFA-A algorithm with its degenerate one that does not consider the state temporal relationship. In Section 4, we present the Alt-TFA-A algorithm, and make a comparative performance analysis on the Alt-TFA-A and TFA-A algorithms accompanied by the experimental supports. Lastly, we draw a conclusion in Section 5.

2. General TBYY system and TFA implementation

2.1. General framework of TBYY system

In the following, we will briefly introduce the TBYY system. Readers interested in it please refer to [5,6] for details.

Papers [5,6] formulate the joint pdf of the whole series of observations $\{\mathbf{x}_t\}_{t=1}^N$ and states $\{\mathbf{y}_t\}_{t=0}^N$ into two Bayesian representation forms:

$$\begin{aligned} p_{M_1}(\mathbf{X}_N, \mathbf{Y}_N) &= p_{M_{\mathrm{y|x}}}(\mathbf{Y}_N|\mathbf{X}_N)p_{M_{\mathrm{x}}}(\mathbf{X}_N), \\ p_{M_2}(\mathbf{X}_N, \mathbf{Y}_N) &= p_{M_{\mathrm{x|y}}}(\mathbf{X}_N|\mathbf{Y}_N)p_{M_{\mathrm{y}}}(\mathbf{Y}_N), \end{aligned} \tag{4}$$

where $\mathbf{X}_N = [\mathbf{x}_N^{\mathrm{T}}, \mathbf{x}_{N-1}^{\mathrm{T}}, \ldots, \mathbf{x}_1^{\mathrm{T}}]^{\mathrm{T}}$, and $\mathbf{Y}_N = [\mathbf{y}_N^{\mathrm{T}}, \mathbf{y}_{N-1}^{\mathrm{T}}, \ldots, \mathbf{y}_0^{\mathrm{T}}]^{\mathrm{T}}$. On one hand, p_{M_1} is called Yang model, which consists of two components: $p_{M_{\mathrm{x}}}$ representing the *observation* (or called *Yang*) *space*, and $p_{M_{\mathrm{y|x}}}$ representing the *Yang* (or *forward*) *pathway* from $\mathbf{X}$ to $\mathbf{Y}$. On the other hand, we have Ying model p_{M_2} that consists of the other two components: $p_{M_{\mathrm{y}}}$ is the *invisible state* (or called *Ying*) *space*, and $p_{M_{\mathrm{x|y}}}$ is the *Ying* (or *backward*) *pathway* from $\mathbf{Y}$ to $\mathbf{X}$. Such a pair of Ying–Yang models is called *temporal Bayesian Ying–Yang* (TBYY) *learning system*.

As shown in [5–7], $p_{M_{\mathrm{x}}}, p_{M_{\mathrm{y|x}}}, p_{M_{\mathrm{x|y}}}, p_{M_{\mathrm{y}}}$ are four designable pdf components. After specifying their structures, i.e., let them be some appropriate density functions with a finite number of unknown parameters, denoted as $\boldsymbol{\theta}_{\mathbf{x}}$, $\boldsymbol{\theta}_{\mathbf{y|x}}$, $\boldsymbol{\theta}_{\mathbf{x|y}}$ and $\boldsymbol{\theta}_{\mathbf{y}}$, respectively, the remaining task is to determine two unknowns. One is the component parameter set $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_{\mathbf{x}}, \boldsymbol{\theta}_{\mathbf{y|x}}, \boldsymbol{\theta}_{\mathbf{x|y}}, \boldsymbol{\theta}_{\mathbf{y}}\}$. The other one is the complexity of $\mathbf{y}_t$, denoted as k, which is actually either the dimension of $\mathbf{y}_t$ when $\mathbf{y}_t$ is a real-value vector, or the number of values that $\mathbf{y}_t$ takes in discrete case. The TBYY system determines the parameter set $\boldsymbol{\Theta}$ and the complexity k based on the fundamental *Ying–Yang harmony* principle: *The parameter set* $\boldsymbol{\Theta}$ *and the complexity* k *are decided such that the Ying model* p_{M_2} *and the Yang model* p_{M_1} *are the best harmony in a sense that we minimize both the mismatch between the two models and the diversification of the resulted*

Ying–Yang system. In implementation, a harmony measure is proposed for specifying both K and $\mathbf{\Theta}$ [6,7], which can be implemented in parallel or in a sequential way. In the sequential implementation, we first enumerate K for certain values, and at each K value, the corresponding $\mathbf{\Theta}$ is determined. Then, we determine $K^* = \operatorname{argmin}_k J(K)$ with a criterian $J(K)$ obtained from the harmony measure. This paper only considers the problem of learning $\mathbf{\Theta}$ at a fixed K in the sequential implementation.

Specifically, we have two choices to solve this problem. One is to make learning under the harmony measure, and details are referred to [6,7]. In the following, we focus on the other choice, that is, to minimize the Kullback divergence between Yang model and Ying model:

$$KL(\mathbf{\Theta}) = \int p_{M_1}(\mathbf{X}_N, \mathbf{Y}_N) \ln \frac{p_{M_1}(\mathbf{X}_N, \mathbf{Y}_N)}{p_{M_2}(\mathbf{X}_N, \mathbf{Y}_N)} \, \mathrm{d}\mathbf{X}_N \, \mathrm{d}\mathbf{Y}_N. \tag{5}$$

To simplify Eq. (5), papers [5,6] have made two general assumptions:

- Suppose $\mathbf{x}_t$'s are causal. That is, $\mathbf{x}_t$ only depends on those past $\mathbf{x}_\tau$'s with $\tau < t$. Furthermore, the causal assumptions are also imposed on

$$\begin{aligned} p_{M_{\mathrm{y|x}}}(\mathbf{y}_t|\bar{\mathbf{X}}_N, \mathbf{Y}_{t-1}) &= p_{M_{\mathrm{y|x}}}(\mathbf{y}_t|\bar{\mathbf{X}}_t, \mathbf{Y}_{t-1}), \\ p_{M_{\mathrm{x|y}}}(\bar{\mathbf{x}}_t|\mathbf{Y}_N, \bar{\mathbf{X}}_{t-1}) &= p_{M_{\mathrm{x|y}}}(\bar{\mathbf{x}}_t|\mathbf{Y}_t, \bar{\mathbf{X}}_{t-1}). \end{aligned} \tag{6}$$

- Let $p_{M_{\mathrm{x}}}(\mathbf{X}_N)$ be modeled by

$$\begin{aligned} p_{M_{\mathrm{x}}}(\mathbf{X}_N) &= \prod_{t=2}^{N} p_{M_{\mathrm{x}}}(\mathbf{x}_t|\mathbf{X}_{t-1}), \\ p_{M_{\mathrm{x}}}(\mathbf{x}_t|\mathbf{X}_{t-1}) &= \begin{cases} \delta(\mathbf{x}_t - \bar{\mathbf{x}}_t) & \text{at } \mathbf{x}_t = \bar{\mathbf{x}}_t, \\ \text{undefined} & \text{otherwise}, \end{cases} \end{aligned} \tag{7}$$

  where δ is the Dirac delta function, and $\bar{\mathbf{x}}_t$ is a sample point of $\mathbf{x}_t$.
- In the derivation, we encounter a term

$$F_t(\mathbf{Y}_{t-1}) = \ln \frac{p_{M_{\mathrm{y|x}}}(\mathbf{y}_t|\bar{\mathbf{X}}_t, \mathbf{Y}_{t-1})}{p_{M_{\mathrm{x|y}}}(\bar{\mathbf{x}}_t|\mathbf{y}_t, \bar{\mathbf{X}}_{t-1}, \mathbf{Y}_{t-1}) p_{M_{\mathrm{y}}}(\mathbf{y}_t|\mathbf{Y}_{t-1})}, \tag{8}$$

  which is approximated by its first-order Taylor expansion at the mean point $\hat{\mathbf{Y}}_{t-1} = E(\mathbf{Y}_{t-1}|\bar{\mathbf{X}}_{t-1})$.

With some mathematical computation, papers [5,6] have shown that the learning of $\mathbf{\Theta}$ to minimize $KL(\mathbf{\Theta})$ can be stochastically implemented by tuning $\mathbf{\Theta}$ in a small step size along the direction to minimize

$$KL_t(\mathbf{\Theta}) = \int p_{M_{\mathrm{y|x}}}(\mathbf{y}_t|\bar{\mathbf{X}}_t, \hat{\mathbf{Y}}_{t-1}) F_t(\hat{\mathbf{Y}}_{t-1}) \, \mathrm{d}\mathbf{y}_t \tag{9}$$

at each time step t.

Furthermore, it follows from Eqs. (1)–(3) that we have

$$\begin{aligned} p_{M_{\mathrm{y|x}}}(\mathbf{y}_t|\bar{\mathbf{X}}_t, \mathbf{Y}_{t-1}) &= p_{M_{\mathrm{y|x}}}(\mathbf{y}_t|\bar{\mathbf{X}}_t, \mathbf{y}_{t-1}) \\ &= G(\mathbf{y}_t|\hat{\mathbf{y}}_t, \mathbf{\Sigma}_\zeta), \end{aligned} \tag{10}$$

$$
\begin{aligned}
p_{M_{x|y}}(\bar{\mathbf{x}}_t|\mathbf{y}_t,\bar{\mathbf{X}}_t,\hat{\mathbf{Y}}_{t-1}) &= p_{M_{x|y}}(\bar{\mathbf{x}}_t|\mathbf{y}_t)\\
&= G(\mathbf{x}_t|\mathbf{A}\mathbf{y}_t+\mathbf{c}_\mathbf{e},\boldsymbol{\Sigma}_\mathbf{e}),
\end{aligned} \tag{11}
$$

$$
p_{M_y}(\mathbf{y}_t|\hat{\mathbf{Y}}_{t-1}) = G(\mathbf{y}_t|\boldsymbol{\Lambda}\hat{\mathbf{y}}_{t-1}+\mathbf{c}_\varepsilon,\boldsymbol{\Sigma}_\varepsilon), \tag{12}
$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix, $G(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Gamma})$ denotes the Gaussian pdf of $\mathbf{x}$ with mean $\boldsymbol{\mu}$ and co-variance $\boldsymbol{\Gamma}$, and $\hat{\mathbf{y}}_t = E(\mathbf{y}_t|\bar{\mathbf{X}}_t,\hat{\mathbf{Y}}_{t-1})$ is a posteriori state estimate of $\mathbf{y}_t$. Putting Eqs. (10)–(12) into Eq. (9), with some mathematical computation, we then obtain a general TFA cost function:

$$
\begin{aligned}
KL_t(\boldsymbol{\Theta}) = \frac{1}{2}\Bigg\{&\ln\frac{|\boldsymbol{\Sigma}_\mathbf{e}\boldsymbol{\Sigma}_\varepsilon|}{|\boldsymbol{\Sigma}_\zeta|} + \mathrm{Tr}(\boldsymbol{\Sigma}_\mathbf{e}^{-1}\mathbf{A}\boldsymbol{\Sigma}_\zeta\mathbf{A}^\mathrm{T} + \boldsymbol{\Sigma}_\zeta\boldsymbol{\Sigma}_\varepsilon^{-1})\\
&+ \hat{\mathbf{e}}_t^\mathrm{T}\boldsymbol{\Sigma}_\mathbf{e}^{-1}\hat{\mathbf{e}}_t + \hat{\boldsymbol{\varepsilon}}_t^\mathrm{T}\boldsymbol{\Sigma}_\varepsilon^{-1}\hat{\boldsymbol{\varepsilon}}_t\Bigg\} + C
\end{aligned} \tag{13}
$$

with

$$
\begin{aligned}
\hat{\mathbf{e}}_t &= \bar{\mathbf{x}}_t - (\mathbf{A}\hat{\mathbf{y}}_t + \mathbf{c}_\mathbf{e}),\\
\hat{\boldsymbol{\varepsilon}}_t &= \hat{\mathbf{y}}_t - \hat{\mathbf{y}}_t^-,\\
\hat{\mathbf{y}}_t^- &= \boldsymbol{\Lambda}\hat{\mathbf{y}}_{t-1} + \mathbf{c}_\varepsilon,
\end{aligned} \tag{14}
$$

where C is a constant, $\hat{\mathbf{e}}_t$ represents the reconstruction error to the observation, and $\hat{\boldsymbol{\varepsilon}}_t$ is the difference between a priori state estimate $\hat{\mathbf{y}}_t^-$ and a posteriori state estimate $\hat{\mathbf{y}}_t$.

2.2. A gradient-based TFA algorithm (TFA-A)

In papers [5,6], the term $G(\mathbf{y}_t|\hat{\mathbf{y}}_t,\boldsymbol{\Sigma}_\zeta)$ in Eq. (10) is explicitly expressed as a parametric density form

$$
G(\mathbf{y}_t|\hat{\mathbf{y}}_t,\boldsymbol{\Sigma}_\zeta) = G(\mathbf{y}_t|\mathbf{K}\bar{\mathbf{x}}_t + \mathbf{H}\hat{\mathbf{y}}_{t-1} + \mathbf{c}_\zeta,\boldsymbol{\Sigma}_\zeta) \tag{15}
$$

with

$$
\hat{\mathbf{y}}_t = \mathbf{K}\bar{\mathbf{x}}_t + \mathbf{H}\hat{\mathbf{y}}_{t-1} + \mathbf{c}_\zeta, \tag{16}
$$

where $\mathbf{K}$ and $\mathbf{H}$ are two independent parameters. Papers [5,6] have proposed a gradient-based TFA algorithm (TFA-A) as follows:

1. Fix $\boldsymbol{\Theta}_{\mathrm{Yang}}$ and $\boldsymbol{\Theta}_{\mathrm{Ying}}$, calculate $\hat{\mathbf{y}}_t$ by Eq. (16) as an estimate of $\mathbf{y}_t$.
2. Update $\boldsymbol{\Theta}_{\mathrm{Yang}}$ and $\boldsymbol{\Theta}_{\mathrm{Ying}}$ by the gradient-based method with

$$
\begin{aligned}
\boldsymbol{\Theta}_{\mathrm{Yang}}^{\mathrm{new}} &= \boldsymbol{\Theta}_{\mathrm{Yang}}^{\mathrm{old}} - \eta\Delta\boldsymbol{\Theta}_{\mathrm{Yang}}|_{\boldsymbol{\Theta}_{\mathrm{Yang}}^{\mathrm{old}}},\\
\boldsymbol{\Theta}_{\mathrm{Ying}}^{\mathrm{new}} &= \boldsymbol{\Theta}_{\mathrm{Ying}}^{\mathrm{old}} - \eta\Delta\boldsymbol{\Theta}_{\mathrm{Ying}}|_{\boldsymbol{\Theta}_{\mathrm{Ying}}^{\mathrm{old}}},
\end{aligned} \tag{17}
$$

where η is a small positive learning rate, $\boldsymbol{\Theta}=\boldsymbol{\Theta}_{\mathrm{Yang}}\cup\boldsymbol{\Theta}_{\mathrm{Ying}}$ with $\boldsymbol{\Theta}_{\mathrm{Yang}}=\{\mathbf{K},\mathbf{H},\mathbf{c}_\zeta,\boldsymbol{\Sigma}_\zeta\}$ and $\boldsymbol{\Theta}_{\mathrm{Ying}}=\{\boldsymbol{\Lambda},\boldsymbol{\Sigma}_\varepsilon,\mathbf{c}_\mathbf{e},\mathbf{c}_\varepsilon\}$. The $\Delta\boldsymbol{\Theta}_{\mathrm{Yang}}$ and $\Delta\boldsymbol{\Theta}_{\mathrm{Ying}}$ are either the gradient descents $\partial KL_t(\boldsymbol{\Theta})/\partial\boldsymbol{\Theta}_{\mathrm{Yang}}$ and $\partial KL_t(\boldsymbol{\Theta})/\partial\boldsymbol{\Theta}_{\mathrm{Ying}}$ as given in Table 1 and Table 2, respectively,

Table 1
The gradient descent direction $\partial KL_t(\mathbf{\Theta})/\partial \mathbf{\Theta}_{\mathrm{Yang}}$ of $\mathbf{\Theta}_{\mathrm{Yang}}$ at time step t

$\frac{\partial KL_t(\mathbf{\Theta})}{\partial \mathbf{\Theta}_{\mathrm{Yang}}} = \left\{ \frac{\partial KL_t(\mathbf{\Theta})}{\partial \mathbf{K}}, \frac{\partial KL_t(\mathbf{\Theta})}{\partial \mathbf{H}}, \frac{\partial KL_t(\mathbf{\Theta})}{\partial \mathbf{c}_\zeta}, \frac{\partial KL_t(\mathbf{\Theta})}{\partial \mathbf{\Sigma}_\zeta} \right\}$
$\frac{\partial KL_t(\mathbf{\Theta})}{\partial \mathbf{K}} = (\mathbf{\Sigma}_\varepsilon^{-1}\hat{\boldsymbol{\varepsilon}}_t - \mathbf{A}^{\mathrm{T}}\mathbf{\Sigma}_{\mathbf{e}}^{-1}\hat{\mathbf{e}}_t)\bar{\mathbf{x}}_t^{\mathrm{T}}$
$\frac{\partial KL_t(\mathbf{\Theta})}{\partial \mathbf{H}} = (\mathbf{\Sigma}_\varepsilon^{-1}\hat{\boldsymbol{\varepsilon}}_t - \mathbf{A}^{\mathrm{T}}\mathbf{\Sigma}_{\mathbf{e}}^{-1}\hat{\mathbf{e}}_t)\hat{\mathbf{y}}_{t-1}^{\mathrm{T}}$
$\frac{\partial KL_t(\mathbf{\Theta})}{\partial \mathbf{c}_\zeta} = \mathbf{\Sigma}_{\boldsymbol{\varepsilon}}^{-1}\hat{\boldsymbol{\varepsilon}}_t - \mathbf{A}^{\mathrm{T}}\mathbf{\Sigma}_{\mathbf{e}}^{-1}\hat{\mathbf{e}}_t$
$\frac{\partial KL_t(\mathbf{\Theta})}{\partial \mathbf{\Sigma}_\zeta} = \mathbf{\Sigma}_\varepsilon^{-1} + \mathbf{A}^{\mathrm{T}}\mathbf{\Sigma}_{\mathbf{e}}^{-1}\mathbf{A} - \mathbf{\Sigma}_\zeta^{-1}$

Table 2
The gradient descent direction $\partial KL_t(\mathbf{\Theta})/\partial \mathbf{\Theta}_{\mathrm{Ying}}$ of $\mathbf{\Theta}_{\mathrm{Ying}}$ at time step t

$\frac{\partial KL_t(\mathbf{\Theta})}{\partial \mathbf{\Theta}_{\mathrm{Ying}}} = \{ \frac{\partial KL_t(\mathbf{\Theta})}{\partial \mathbf{A}}, \frac{\partial KL_t(\mathbf{\Theta})}{\partial \lambda_j}, \frac{\partial KL_t(\mathbf{\Theta})}{\partial \mathbf{\Sigma}_{\mathbf{e}}}, \frac{\partial KL_t(\mathbf{\Theta})}{\partial \mathbf{c}_{\mathbf{e}}}, \frac{\partial KL_t(\mathbf{\Theta})}{\partial \mathbf{c}_\varepsilon} \}$
$\frac{\partial KL_t(\mathbf{\Theta})}{\partial \mathbf{A}} = \mathbf{\Sigma}_{\mathbf{e}}^{-1}(\mathbf{A}\mathbf{\Sigma}_\zeta - \hat{\mathbf{e}}_t\hat{\mathbf{y}}_t^{\mathrm{T}})$
$\frac{\partial KL_t(\mathbf{\Theta})}{\partial \lambda_j} = -(\hat{\boldsymbol{\varepsilon}}_t\hat{\mathbf{y}}_{t-1}^{\mathrm{T}})_{j,j}$
$\frac{\partial KL_t(\mathbf{\Theta})}{\partial \mathbf{\Sigma}_{\mathbf{e}}} = \mathbf{\Sigma}_{\mathbf{e}}^{-1}[\mathbf{\Sigma}_{\mathbf{e}} - (\mathbf{A}\mathbf{\Sigma}_\zeta\mathbf{A}^{\mathrm{T}} + \hat{\mathbf{e}}_t\hat{\mathbf{e}}_t^{\mathrm{T}})]\mathbf{\Sigma}_{\mathbf{e}}^{-1}$
$\frac{\partial KL_t(\mathbf{\Theta})}{\partial \mathbf{c}_{\mathbf{e}}} = -\mathbf{\Sigma}_{\mathbf{e}}^{-1}\hat{\mathbf{e}}_t$
$\frac{\partial KL_t(\mathbf{\Theta})}{\partial \mathbf{c}_\varepsilon} = -\mathbf{\Sigma}_\varepsilon^{-1}\hat{\boldsymbol{\varepsilon}}_t$
where λ_j is the jth diagonal element of $\mathbf{\Lambda}$, $1 \leqslant j \leqslant k$; and $\mathbf{B}_{j,j}$ denotes the (j,j)th element of matrix $\mathbf{B}$.

or the modifications by multiplying the gradient descents and a positive-definite matrix, e.g., $\mathbf{\Sigma}_\zeta \partial KL_t(\mathbf{\Theta})/\partial \mathbf{A}$ and $\mathbf{\Sigma}_{\mathbf{e}}(\partial KL_t(\mathbf{\Theta})/\partial \mathbf{\Sigma}_{\mathbf{e}})\mathbf{\Sigma}_{\mathbf{e}}$.

Please note that the scale of the state $\mathbf{y}_t$ in Eq. (2) is not identifiable because it can be absorbed by the unknown parameter $\mathbf{A}$. Without loss of generality, we therefore set $\mathbf{\Sigma}_\varepsilon$ at the identity matrix $\mathbf{I}$ in tuning $\mathbf{\Theta}$ by Eq. (17).

Furthermore, Eq. (17) should be noted that the learning of $\mathbf{\Lambda}$ will generally lead to the non-stationary process of $\hat{\mathbf{y}}_t^-$ in the model of $p_{M_y}(\mathbf{y}_t|\hat{\mathbf{Y}}_{t-1})$. If we further assume that the process of $\mathbf{y}_t$ in Eq. (1) is stationary, i.e., each diagonal element λ_j of $\mathbf{\Lambda}$ satisfies $|\lambda_j| < 1$, we can let $\mathbf{\Lambda}$ be the function of another new diagonal matrix $\mathbf{Z}$ such that the constraints on $\mathbf{\Lambda}$ can be automatically satisfied when $\mathbf{Z}$ is tuned in the learning without any constraint. Here, we let

$$\lambda_j = f(z_j) = \frac{2}{1 + \mathrm{e}^{z_j}} - 1, \tag{18}$$

where z_j is the jth diagonal element of $\mathbf{Z}$. In implementation, we therefore learn $\mathbf{Z}$ rather than $\mathbf{\Lambda}$ by

$$\Delta z_j = \Delta\lambda_j \times \frac{\mathrm{d}\lambda_j}{\mathrm{d}z_j} \tag{19}$$

with

$$\frac{\mathrm{d}\lambda_j}{\mathrm{d}z_j} = -\frac{1}{2}\mathrm{e}^{z_j}(1 + \lambda_j)^2. \tag{20}$$

3. Comparison of the TFA-A algorithm with non-temporal one

When $\mathbf{y}_t$ is i.i.d. Gaussian variable and $\mathbf{x}_t$ is modeled by Eq. (2), it is well known that $\hat{\mathbf{y}}_t$ can be arbitrarily rotated without destroying the component-wise independence requirement, thus it is impossible to identify $\mathbf{y}_t$ up to any constant scales and permutation indices from the observations [4].

Moreover, Conclusion 3 in [6] has shown that a Gaussian process $\mathbf{y}_t$ can be still successfully identified when the observed samples are time-correlated. This implies that it is important to consider the observation internal time relationship in performing Gaussian process identification.

To illustrate this point, we demonstrate the performance of two algorithms for comparison. One is the TFA-A presented in the previous section, and the other is an algorithm for the conventional factor analysis, which is actually equivalent to a degenerate case of the TFA-A algorithm with $\mathbf{\Lambda} = \mathbf{H} = \mathbf{0}$, and thus no consideration on the internal state time dependence. We use *Dg-TFA-A* to denote this algorithm hereafter.

In the experiment, we let the observations $\{\mathbf{x}_t\}_{t=1}^N$ be generated by

$$\mathbf{y}_t = \begin{pmatrix} 0.7 & 0.0 & 0.0 \\ 0.0 & -0.3 & 0.0 \\ 0.0 & 0.0 & 0.5 \end{pmatrix} \mathbf{y}_{t-1} + \boldsymbol{\varepsilon}_t, \tag{21}$$

$$\mathbf{x}_t = \begin{pmatrix} 1.5 & 0.8 & 0.7 \\ 0.7 & -1.0 & 0.6 \\ 1.2 & 0.8 & 2.0 \end{pmatrix} \mathbf{y}_t + \mathbf{e}_t, \quad 1 \leqslant t \leqslant N.$$

Without loss of generality, we suppose $E(\mathbf{y}_0) = \mathbf{0}$, and thus have $\mathbf{c}_\zeta = \mathbf{c}_\mathbf{e} = \mathbf{c}_\varepsilon = \mathbf{0}$. We initialize $\mathbf{y}_0 = \mathbf{0}$, $\mathbf{\Sigma}_\zeta = 0.3\mathbf{I}$, $\mathbf{\Sigma}_\mathbf{e} = 0.2\mathbf{I}$, and let $\boldsymbol{\varepsilon}_t$ be distributed with pdf $G(\boldsymbol{\varepsilon}|\mathbf{0}, 0.8\mathbf{I})$ while $\mathbf{e}_t$ is Gaussian distributed with $G(\mathbf{e}|\mathbf{0}, 0.1\mathbf{I})$.

We measure the performance of the algorithms by MSE value, defined by

$$MSE(y^{(j)}, \hat{y}^{(j)}) = \frac{1}{N} \sum_{t=1}^{N} (y_t^{(j)} - \hat{y}_t^{(j)})^2, \tag{22}$$

where $y^{(j)}$ and $\hat{y}_t^{(j)}$ are the jth component of true state $\mathbf{y}_t$ and corresponding state estimate $\hat{\mathbf{y}}_t$, respectively. Since the observations are sequentially observed without repeat, we calculate the MSE values on-line once every 10,000 data points, and normalize both $y_t^{(j)}$ and $\hat{y}_t^{(j)}$ to variance 1 in order to calculate MSE values invariant to scaling.

After scanning 500,000 observation points, a snapshot of MSE values obtained from the TFA-A algorithm were 0.0906, 0.0761 and 0.1208, whereas those from the Dg-TFA-A algorithm were 0.3175, 0.3445 and 0.1317. Fig. 1 shows the performance graph of the TFA-A and Dg-TFA-A, respectively, and Fig. 2 gives a slide window of state identification by these two algorithms. It can be seen that the performance of TFA-A algorithm considerably outperforms the Dg-TFA-A, although the latter converges faster than the former because of the reduced number of free parameters.

4. New Kalman filter-based TFA algorithm from TBYY system (Alt-TFA-A)

The TFA-A algorithm presented in [5,6] explicitly expresses the posteriori state estimate $\hat{\mathbf{y}}_t$ via parameters $\mathbf{K}$ and $\mathbf{H}$ as shown in Eq. (16). Alternatively, the $\hat{\mathbf{y}}_t$ and $\mathbf{\Sigma}_\zeta$ can also be indirectly indicated through Bayesian inversion. That is, we let

$$p_{M_{\mathrm{y|x}}}(\mathbf{y}_t|\bar{\mathbf{X}}_t, \mathbf{Y}_{t-1}) = G(\mathbf{y}_t|\hat{\mathbf{y}}_t, \mathbf{\Sigma}_\zeta)$$

$$= \frac{p_{M_{\mathrm{x|y}}}(\bar{\mathbf{x}}_t|\mathbf{y}_t, \bar{\mathbf{X}}_{t-1}, \mathbf{Y}_{t-1}) p_{M_\mathrm{y}}(\mathbf{y}_t|\bar{\mathbf{X}}_{t-1}, \mathbf{Y}_{t-1})}{\int p_{M_{\mathrm{x|y}}}(\bar{\mathbf{x}}_t|\mathbf{y}_t, \bar{\mathbf{X}}_{t-1}, \mathbf{Y}_{t-1}) p_{M_\mathrm{y}}(\mathbf{y}_t|\bar{\mathbf{X}}_{t-1}, \mathbf{Y}_{t-1})\, \mathrm{d}\mathbf{y}_t}.$$

Under the circumstances, as $\mathbf{\Theta}_{\mathrm{Ying}}$ is fixed, the posteriori estimation of $\mathbf{y}_t$ becomes the traditional filtering problem. We therefore can estimate $\hat{\mathbf{y}}_t$ and $\mathbf{\Sigma}_\zeta$ via Kalman filter (see [1, pp. 36–89]).

Consequently, we can obtain an alternative TFA-A algorithm, shortly denoted as Alt-TFA-A algorithm, as follows:

1. Fixing $\mathbf{\Theta}$, we estimate

$$\hat{\mathbf{y}}_t = \hat{\mathbf{y}}_t^- + \mathbf{K}_t(\bar{\mathbf{x}}_t - \mathbf{A}\hat{\mathbf{y}}_t^-) \tag{23}$$

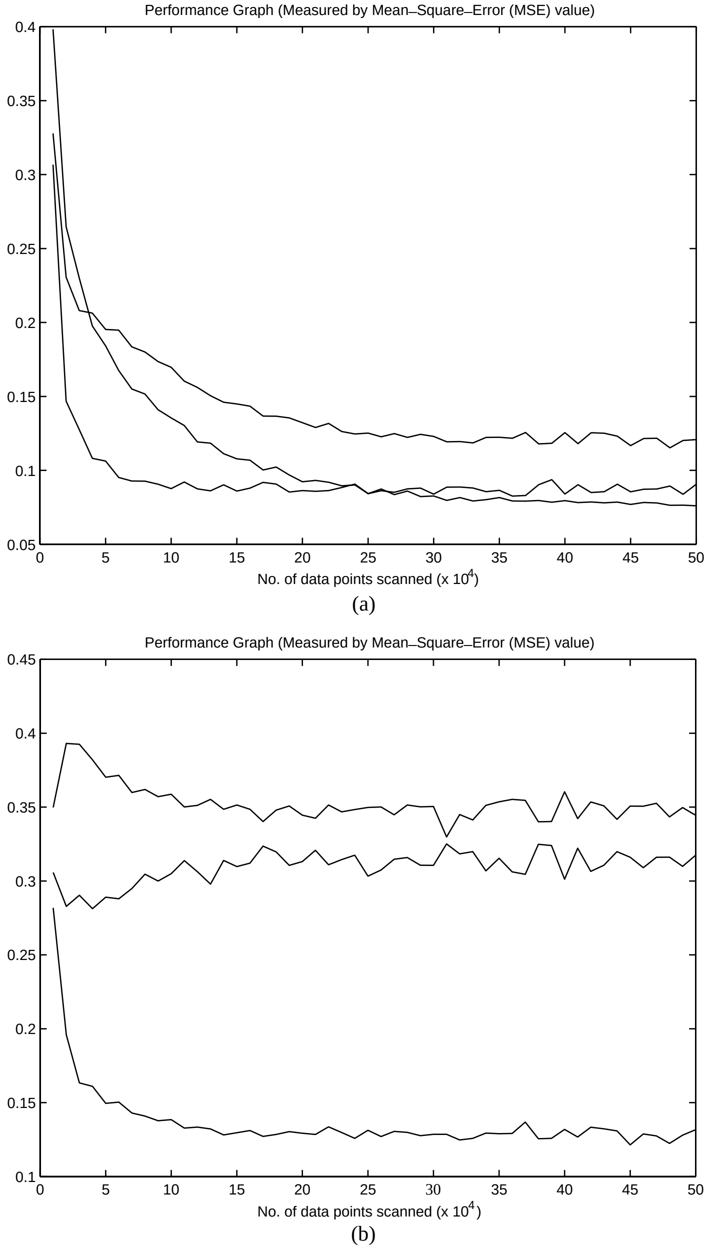
with

$$\mathbf{K}_t = \mathbf{\Sigma}_\varepsilon \mathbf{A}^\mathrm{T}(\mathbf{A}\mathbf{\Sigma}_\varepsilon \mathbf{A}^\mathrm{T} + \mathbf{\Sigma}_\mathbf{e})^{-1}, \tag{24}$$

where $\hat{\mathbf{y}}_t^-$ is given by Eq. (14).

2. Fixing $\mathbf{\Theta}_{\mathrm{ying}}$, we calculate

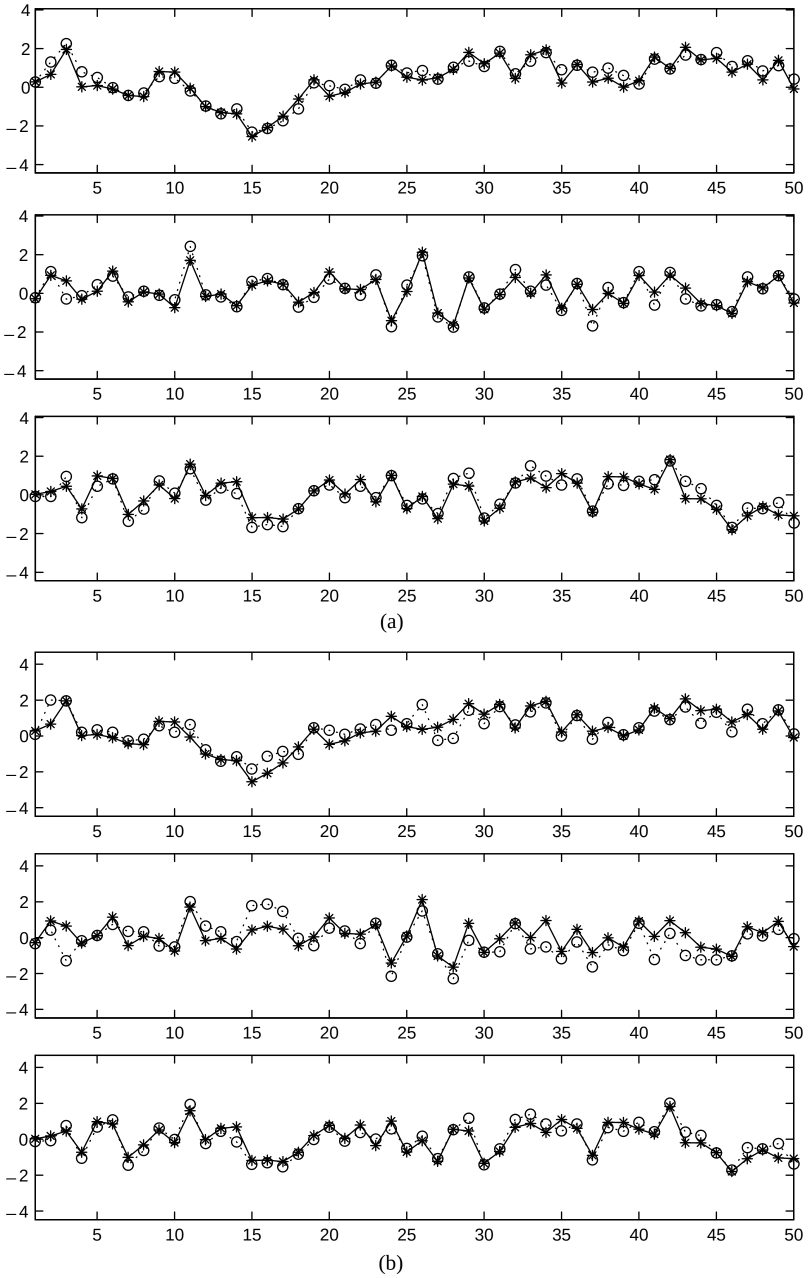
$$\mathbf{\Sigma}_\zeta = [\mathbf{I} - \mathbf{K}_t\mathbf{A}]\mathbf{\Sigma}_\varepsilon. \tag{25}$$



(a)

(b)

Fig. 1. The MSE performance graph of (a) TFA-A algorithm and (b) Dg-TFA-A algorithm.



(a)

(b)

Fig. 2. A slide window of three-dimensional state identification (a) by TFA-A algorithm and (b) by Dg-TFA-A algorithm. In each sub-figure, row j with $1 \leqslant j \leqslant 3$ shows the identifying results of $y_t^{(j)}$, where '*' denotes the value of $y_t^{(j)}$, and 'o' denotes the value of $\hat{y}_t^{(j)}$.

3. Fixing $\hat{\mathbf{y}}_t$ and $\mathbf{\Sigma}_\zeta$, we estimate $\mathbf{\Theta}_{\text{ying}}$ by Eq. (17), which is the same as the TFA-A algorithm given in Table 2.

The Alt-TFA-A algorithm estimates $\hat{\mathbf{y}}_t$ and $\mathbf{\Sigma}_\zeta$ by Kalman filter with the use of the second-order statistics. Actually, we can also tune $\mathbf{\Theta}_{\text{ying}}$ by using the second-order information, which however, involves to calculate the complicated Hessian matrix of $\mathbf{\Theta}_{\text{ying}}$. We have to make a trade-off between the state estimate error and the computing costs, and thus tune $\mathbf{\Theta}_{\text{ying}}$ by the gradient-based method with the first-order information used only.

4.1. Performance analysis

During the learning of Alt-TFA-A and TFA-A algorithms, Alt-TFA-A uses Kalman filter to analytically calculate the posteriori state estimate $\hat{\mathbf{y}}_t$ and the variance $\mathbf{\Sigma}_\zeta$ at each time step t. However, the TFA-A algorithm lets $\hat{\mathbf{y}}_t$ be the function of two independent parameters $\mathbf{K}$ and $\mathbf{H}$, and uses the gradient-based method to tune them as well as $\mathbf{\Sigma}_\zeta$ in one small step-size each time along the direction of minimizing $KL_t(\mathbf{\Theta})$. Consequently, the TFA-A algorithm is faster to reduce the MSE values than the Alt-TFA-A one.

Moreover, the Alt-TFA-A algorithm gives the same form of posteriori state estimation as that in TFA-A algorithm, where $\hat{\mathbf{y}}_t$ is the linear function of current observation $\bar{\mathbf{x}}_t$ and the priori state estimate $\hat{\mathbf{y}}_t^-$. However, in the Alt-TFA-A algorithm, the two coefficients of this linear function are all dependent on the parameter $\mathbf{K}_t$, which is optimally calculated at each time step. In comparison, the TFA-A algorithm regards that the coefficients $\mathbf{K}$ and $\mathbf{H}$ are independent parameters, and tunes them by the gradient-based method. As a result, $\mathbf{K}$ and $\mathbf{H}$ gradually converge after the convergence of Ying model parameters. Therefore, the convergent performance of Alt-TFA-A algorithm should generally be better than the TFA-A one.

4.2. Experimental demonstration

To justify the analysis in the previous sub-section, we demonstrate the performance of Alt-TFA-A algorithm as follows:

4.2.1. Experiment 1

The experimental environment is the same as that in Section 3. Fig.3 shows the performance graph of the Alt-TFA-A algorithm. Compared to the TFA-A performance in Fig. 1(a), we found that the Alt-TFA-A algorithm always keep the smaller MSE values during the algorithm learning. That is, the latter reduces the estimation error faster than the former. After performance convergence, a snapshot of Alt-TFA-A MSE values at time step t=500,000 were 0.0561, 0.0633 and 0.0681, and Fig. 4 shows a slide of the estimation errors at each individual state point, where we found that the error curve of the Alt-TFA-A algorithm is lower than that of the TFA-A one in most cases. That is, the state estimates given by the Alt-TFA-A has a smaller MSE on average. In Fig. 5, we give a performance comparison between the Alt-TFA-A algorithm and the

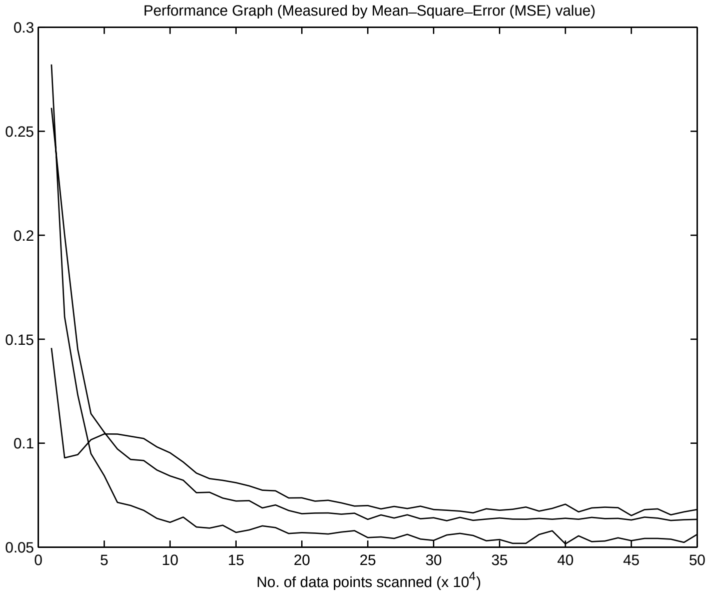


Fig. 3. The MSE performance graph of Alt-TFA-A algorithm.

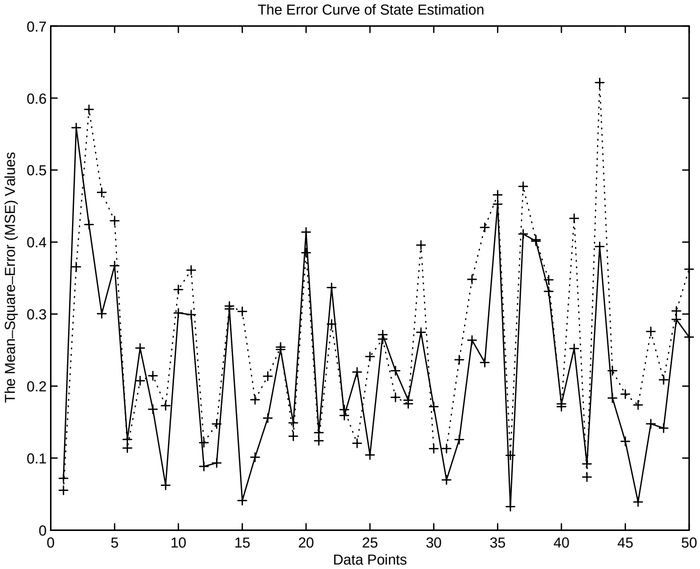
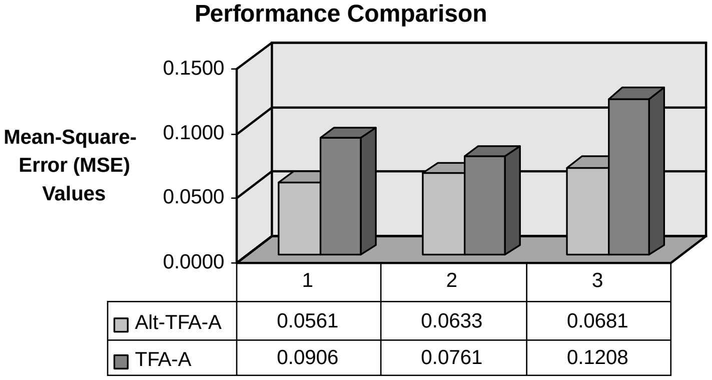


Fig. 4. The error curves of state estimation at individual points, where the solid line is obtained from the Alt-TFA-A algorithm, and the dotted line is from the TFA-A algorithm. In this figure, the value of each discrete point denoted by '+' is the average state estimation error, calculated by $\frac{1}{3}\sum_{j=1}^{3}|y_t^{(j)} - \hat{y}_t^{(j)}|$.



	1	2	3
□ Alt-TFA-A	0.0561	0.0633	0.0681
■ TFA-A	0.0906	0.0761	0.1208

Fig. 5. The performance comparison between the Alt-TFA-A and the TFA-A under the MSE measure.

TFA-A one. It can be seen that the former has significantly improved the estimation accuracy with the average of about 26% on each $y^{(j)}$.

4.2.2. Experiment 2

We further investigate the performance of the Alt-TFA-A and TFA-A algorithms in real filtering problem. We use three real-world music sounds recorded at 22 kHz sampling rate, which is noisily observed as

$$\mathbf{x}_t = \mathbf{s}_t + \mathbf{e}_t, \tag{26}$$

where $\mathbf{s}_t = [s_t^{(1)}, s_t^{(2)}, s_t^{(3)}]^{\mathrm{T}}$ with $s_t^{(j)}$ denoting the jth sound signal, and $\mathbf{e}_t$ is Gaussian white noise with $\boldsymbol{\Sigma}_{\mathbf{e}} = 0.01\mathbf{I}$. Since the actual states in this experiments are unknown, we use an alternative formula of Eq. (22):

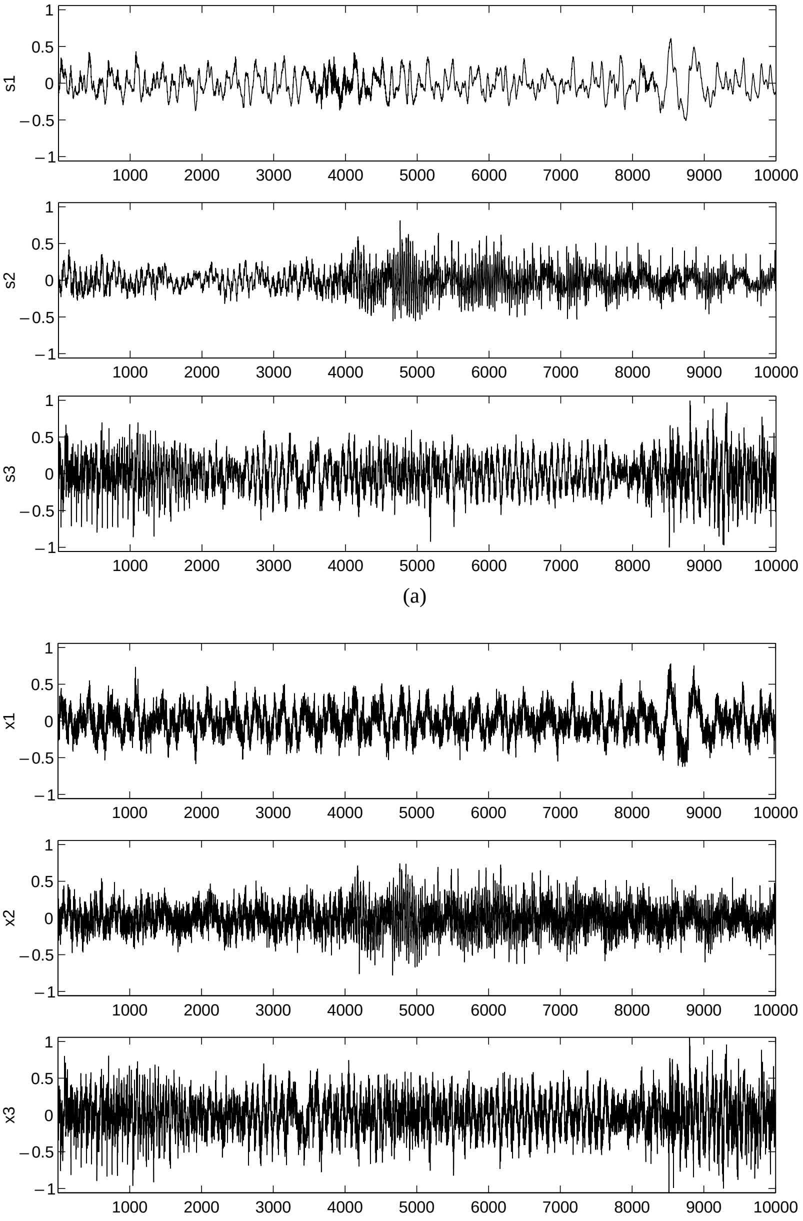
$$MSE(s^{(j)}, \hat{s}^{(j)}) = \frac{1}{N}\sum_{t=1}^{N}(s_t^{(j)} - \hat{s}_t^{(j)})^2 \tag{27}$$

with

$$\hat{\mathbf{s}}_t = \mathbf{A}\hat{\mathbf{y}}_t \tag{28}$$

to measure the performance of the Alt-TFA-A and TFA-A algorithms, respectively.

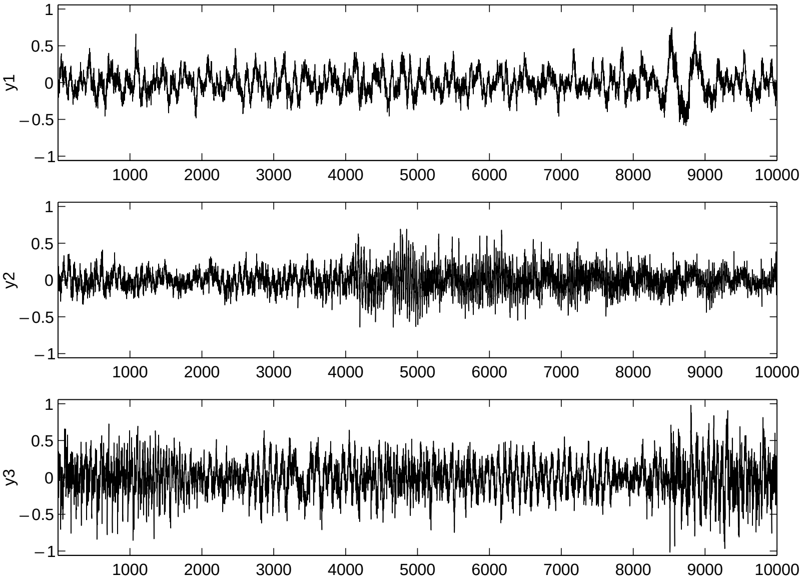
After scanning 800,000 observation points, the average MSE value of Alt-TFA-A algorithm is 0.0054. The filtering results obtained from them are shown in Fig. 6, where we found that the noise has been significantly filtered out. In comparison, the average MSE of TFA-A algorithm is 0.0065. Fig. 7 gives the comparison of their performance. It can be seen that the result is consistent with the analysis conclusions in Section 4.1. Actually, in this case, the Alt-TFA-A algorithm is better to filter out the noise with about 17% noise-reduced improvements in contrast with the TFA-A.



(a)

(b)

Fig. 6. Sub-figures (a) and (b) show a slid window of the music sounds and the noisy observations, respectively. Sub-figure (c) shows the sounds identified through the Alt-TFA-A algorithm.



(c)

Fig. 6. (*continued*).

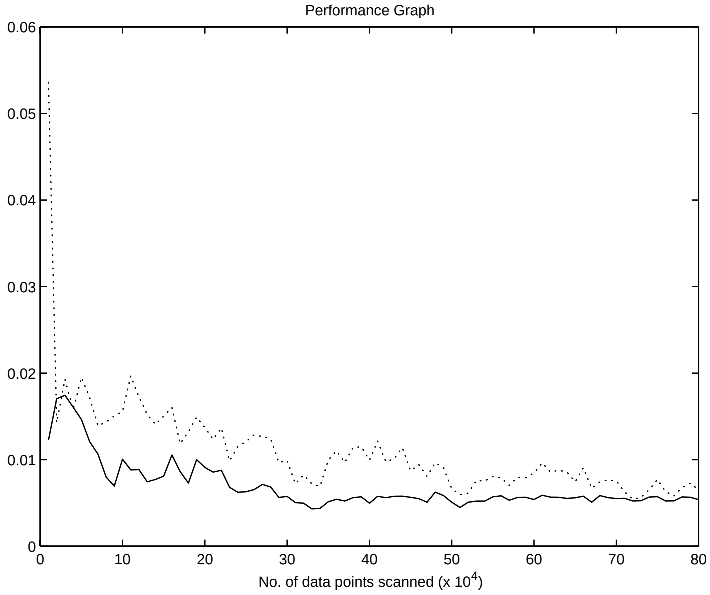


Fig. 7. The performance comparison between the Alt-TFA-A algorithm and the TFA-A, where the solid line is the average performance of the Alt-TFA-A algorithm on three different sounds, whereas the dotted line is that of the TFA-A one.

5. Conclusion

We have further studied the TFA problem. On the one hand, we have implemented the TFA algorithm presented in [5,6], and have shown that it is really useful to take into consideration the serial relations in the states. On the other hand, we have provided a new alternative algorithm, which uses the Kalman filter to make the posteriori estimation of the states and its variance matrix. The experiments have shown that the new algorithm outperforms the TFA-A algorithm.

Acknowledgements

The authors would like to thank the anonymous reviews for their valuable comments and suggestions. The work described in this paper was supported by a grant from the Research Grant Council of the Hong Kong SAR (Project No: CUHK 4169/00E).

References

[1] B.D.O. Anderson, J.B. Moore, in: T. Kailath (Ed.), Optimal Filtering, Information and System Sciences Series, Prentice-Hall, Inc., Engelwood Cliffs, NJ, 1979.

[2] T.W. Anderson, H. Rubin, Statistical Inference in Factor Analysis, Proceedings of the Berkeley Symposium on Mathematical Statistics and Probability, May 3, Berkeley, University of California, 1956, pp. 111–150.

[3] R.G. Brown, Y.C. Hwang, Introduction to Random Signals and Applied Kalman Filtering, 2nd Edition, Wiley, New York, 1992.

[4] L. Tong, Y. Inouye, R.W. Liu, Waveform-preserving blind estimation of multiple independent sources, IEEE Trans. Signal Process. 41 (7) (1993) 2461–2470.

[5] L. Xu, Bayesian Ying–Yang system and theory as a unified statistical learning approach: (V) temporal modeling for temporal perception and control, Invited Paper, Proceedings of the International Conference on Neural Information Processing (ICONIP'98), Vol. 2, Fukuoka, Japan, 1998, pp. 877–884.

[6] L. Xu, Temporal BYY learning for state space approach, hidden Markov model and blind source separation, IEEE Trans. Signal Process. 48 (7) (2000) 2132–2144. A preliminary version has been published on Proceedings of the 1999 International Joint Conference on Neural Networks, Vol. 2, Washington DC, July 1999, pp. 1071–1076.

[7] L. Xu, BYY harmony learning, independent state space and generalized APT financial analysis. IEEE Trans. Neural Networks 12 (4) (2001) 822–849.

Yiu-ming Cheung received Ph.D. degree in Computer Science and Engineering from the Chinese University of Hong Kong, Hong Kong, in 2000. Currently, he is an assistant professor of the Department of Computer Science in Hong Kong Baptist University. His research interests include machine learning, signal processing, data mining, financial modeling and portfolio management.

Lei Xu (IEEE Fellow) is currently a professor in the Department of Computer Science and Engineering at Chinese University, Hong Kong where he joined in 1993 as a senior lecturer first and then attained the current position in 1996. He has been a professor at Peking University since 1992, where he started as a postdoc in the Department of Maths in 1987 and then became one of the 10 exceptionally promoted young associate professors of the Peking University in 1988. During 1989–1993, he worked as a postdoc or a senior research associate in several universities in Finland, Canada and the USA, including Harvard and MIT. He is currently a governor on the Board of Governors, international Neural Networks Society, the chair elected of Computational Finance Technical Committee of IEEE Neural Networks Council, a past president of Asian-Pacific Neural Networks Assembly, and an associate editor for six international journals on neurocomputing, including Neural Networks, and IEEE Transactions on Neural Networks. He was an associate editor of IEEE Transactions on Neural Networks during 1994–1998. Also, served as a general chair of 1998, 2000 International Conference on Intelligent Data Engineering and Automated Learning (IDEAL, Hong Kong), a program committee chair of 1996 International Conference on Neural Information Processing and the chairs of two NIPS Workshops, as well as served as program/organizing committee members on major world conferences on Neural Networks in recent years, including NIPS, IJCNN, ICANN, IEEE WCCI and International Conference on Computational finance. He has received an 1995 international Neural Networks Society Leadership Award and several Chinese national prestigious academic awards, including Chinese National Nature Science Award, Chinese State Education Council Fok Ying Tung Award, and the second of the 10 winners of the 1988 Beijing Young Scientists Prize.