Available online at www.sciencedirect.com
ScienceDirect
Neurocomputing 70 (2006) 206–218
NEUROCOMPUTING
www.elsevier.com/locate/neucom

Balanced parameterization of multichannel blind deconvolutive systems: A continuous time realization

Liangsuo Ma[a,*], Ah Chung Tsoi[b]

[a]Department of Radiology, University of Iowa, Iowa City, IA 52246, USA
[b]Monash University, Clayton Campus, Wellington Road, Clayton, Vic. 3168, Australia

Received 3 December 2004; received in revised form 31 December 2005; accepted 24 March 2006
Communicated by S. Hochreiter
Available online 7 July 2006

Abstract

A key problem in multichannel blind deconvolution (MDB) is how to properly model the mixer so that the demixer can be properly modelled correspondingly. This problem naturally triggers one question, viz. how to determine some key model orders, e.g. the number of states, if the problem is analyzed using a state space model. In this paper, to answer this question, we will apply a balanced parameterization approach to a MBD problem with the mixer being modelled as a continuous time linear time invariant (LTI) system. Besides allowing the determination of the number of states required in the demixer, compared with the controller canonical form representation or the observer canonical form representation of the LTI continuous time system, our approach has also the advantages of numerical robustness. The proposed method is validated through practical examples using speech signals and electroencephalographic (EEG) data.

Keywords: Blind source separation; Multichannel blind deconvolution; Independent component analysis; State space; Balanced parameterization

1. Introduction

The multichannel blind deconvolution (MBD) problem can be formulated as follows: given a set of observation data: the sensor outputs of an unknown multi-input and multi-output dynamical system, the objective of MBD is to recover the latent input signals (sources) based on the output observations. Generally, the inputs are generated by a number of unknown statistically independent signals. In other words, the observation data are the convolutive mixture of the latent sources and the unknown dynamical system. In the context of MBD, the unknown dynamical system is usually called the mixer. Further, the mixer is assumed to be a linear time invariant (LTI) dynamical system in this paper.[1]

The MBD problem is usually solved by devising an inverse system of the mixer, which is generally referred as a demixer. A key problem in MBD is how to properly model the mixer so that the demixer can be modelled correspondingly. This problem naturally triggers one question, viz. how to determine some key model orders of the system models. First, we need to decide if the problem is in the continuous time domain or the discrete time domain. Dependent on the situation, one may assume that the mixer to be either a continuous time dynamical system, or a discrete time dynamical system. In this paper, we will assume that the mixer is modelled by a continuous time dynamical system. Accordingly the demixer is also modelled by a continuous time dynamical system.[2]

*Corresponding author.
E-mail addresses: liangsuo-ma@uiowa.edu (L. Ma), ahchung.tsoi@adm.monash.edu.au (A.C. Tsoi).

[1]In general, the mixer can be a nonlinear time varying system. In this paper, we will only consider a LTI mixer. This may be considered as the simplest possible dynamical system representing a dynamic mixer situation.

[2]Note that this is not necessarily the case all the time. It may be possible to have a discrete time demixer to a continuous time mixer, if the sampling frequency is sufficiently high. On the other hand, it is also possible to have a continuous demixer to a discrete time mixer, provided that the sampling

0925-2312/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2006.03.011

In system theory, it is known that a LTI system is invariant under a coordinate transformation [7]. There are a number of possible canonical forms [7], which include: controller canonical form, observer canonical form, and balanced realized canonical form. Identifying a canonical form from input–output data, it is known that the observer canonical form or the controller canonical form are non-robust computationally [6], as sometimes the parameter estimation algorithm becomes unstable (often due to pole zero cancellations). It is generally acknowledged that the balanced realized canonical form offers a computationally more robust algorithm [4]. Moreover, a balanced realized canonical form allows to estimate the number of states of the mixer by considering the dominant Hankel singular values. Since the balanced realized canonical forms are only available in continuous time domain, this motivates us to study the balanced parameterization of the demixer in the continuous time domain.

However, it is also acknowledged that if the system is unknown, it is quite difficult to recover a balanced realized canonical form [4] directly from input–output measurements. Towards this end, there have been some work [11] in obtaining a balanced parameterization of an unknown linear system[3] in the discrete time domain. To the best of our knowledge, there has not been any investigation of obtaining a balanced parameterization when the mixer is modelled in the continuous time domain. This is what we wish to investigate in this paper, viz., to consider a balanced parameterization approach to MBD problem when the mixer is modelled in the continuous time domain.

This paper is organized as follows. In Section 2, we will briefly review a general continuous time state space MBD algorithm derived in [5]. In Section 3, we will give the balanced parameterization of linear systems. In Section 4, we will derive a continuous time version of balanced parameterization algorithm based on the contents in Sections 2 and 3. In Section 5, we will evaluate the proposed continuous time balanced algorithm through applications to speech signals and EEG signals. We will conclude this paper in Section 6.

2. General continuous time MBD algorithm

The general continuous time state space MBD algorithm was discussed in [5]. In this section, we will briefly review this algorithm, which will form the basis of our continuous time-balanced parameterization algorithm.

2.1. Continuous time state space models

For the sake of simplicity, we will focus on an ideal MBD problem, where both the number of sources and the number of sensors are assumed to be n. Using continuous time state space model as shown in [5], the mixer, which is assumed to be an asymptotically stable, LTI, causal dynamical system, can be modelled as follows:

$$\dot{\bar{\mathbf{x}}}_t = \overline{A}\bar{\mathbf{x}}_t + \overline{B}\mathbf{s}_t, \tag{1a}$$

$$\mathbf{u}_t = \overline{C}\bar{\mathbf{x}}_t + \overline{D}\mathbf{s}_t, \tag{1b}$$

where $\mathbf{s} \in \mathscr{R}^n$ is the source signal vector, $\mathbf{u} \in \mathscr{R}^n$ is the vector of the sensor observations, $\bar{\mathbf{x}} \in \mathscr{R}^{\overline{N}}$ is the vector of the internal states, $\overline{N}$ is the dimension of the states. Generally speaking, $\overline{N}$ is unknown. The observations $\mathbf{u}_t$ is available in the duration $t \in [0, T]$, $T > 0$.

Correspondingly, the demixer is also assumed to be an asymptotically stable, LTI, and causal dynamical system. Using continuous time state space approach, the demixer can be modelled as follows:

$$\dot{\mathbf{x}}_t = A\mathbf{x}_t + B\mathbf{u}_t, \tag{2a}$$

$$\mathbf{y}_t = C\mathbf{x}_t + D\mathbf{u}_t, \tag{2b}$$

where $\mathbf{u} \in \mathscr{R}^n$ is the vector of the observations, which also serves as the inputs of the demixer; $\mathbf{y} \in \mathscr{R}^n$ is the vector of the recovered source signals; $\mathbf{x} \in \mathscr{R}^N$ is the vector of the internal states of the demixer, N is the dimension of the states. For success of separation and deconvolution, we need $N \geqslant \overline{N}$. We will initially assume that $\overline{N}$ is known, though in a later section, we will relax this assumption, and show how this value can be estimated using the balanced parameterization technique to be discussed in this paper. Hence, the parameter set of the demixer is defined as $\Omega \equiv \{A, B, C, D\}$.

The differences between the discrete time state space models and the continuous time state space models can be delineated as follows:

(I) in the discrete time case, the data are the set of discrete samples, while in the continuous time case, the data are given in a continuous form;
(II) in the discrete time case, the dynamics is described using a forward shift operator, while in the continuous time case, the dynamics is described using a dot operator; and
(III) for an asymptotically stable LTI dynamical system, in the discrete time case, the eigenvalues of the state mixing matrix are all located within the unit circle in the complex plane, while in the continuous time case, the eigenvalues of the state mixing matrix are all located on the left half of the complex plane.

(*footnote continued*)
frequency is high. In this paper, we have chosen to consider a continuous time demixer, as it is more convenient. Secondly, once a continuous time demixer is obtained, it is possible to obtain a discrete time equivalent provided that some care is exercised in sampling the continuous time implementation of the demixer.

[3]In this paper, we note the difference between balanced realized canonical form and a balanced parameterization. A balanced realized canonical form is obtained from a known linear system, while a balanced parameterization is to obtain a balanced realized canonical form from input–output measurements.

2.2. Parameter estimation algorithm

The general continuous time state space MBD problem can be tackled by minimizing the mutual information (MI)[4] among the output signals of the demixer, which is defined by using the Kullback–Leibler divergence criterion as follows [5]:

$$I(\Omega) \equiv \int P_{\mathbf{y}}(\mathbf{y}) \log\left(\frac{P_{\mathbf{y}}(\mathbf{y})}{\prod_{k=1}^{n} P_k(y_k)}\right) \mathrm{d}\mathbf{y}, \tag{3}$$

where $P_{\mathbf{y}}(\mathbf{y})$ is the joint probability density function of the signal vector, $\mathbf{y}$; and $P_k(y_k)$ is the marginal probability density function of the kth output. Based on (3), the general continuous time MBD algorithm can be derived through maximizing the following cost function [5]:

$$l(\Omega) = \log|\det(D)| + \sum_{k=1}^{n} \log P_k(y_k) \tag{4}$$

subject to the dynamical system constraints. In [5], such an optimization problem is tackled by introducing the Lagrange multipliers as the adjoint states. In this way, the problem is converted into an unconstrained optimization problem.

The constrained optimization problem can be formulated as follows:

To optimize

$$J_o(\Omega) = \int_0^{\mathrm{T}} \mathrm{d}t\, l(\Omega)$$

subject to the state equation (2a), *with the initial condition* $\mathbf{x}_0 = \mathbf{0}$[5], *where* $\mathbf{0}$ *is an N column null vector.*

This kind of constrained optimization problem can be solved by defining the following performance index [2] through the introduction of a Lagrange multiplier:

$$J_c(\Omega) \equiv \int_0^{\mathrm{T}} \mathrm{d}t\, L(t; \mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}, \Omega), \tag{5}$$

where L is the Lagrangian function defined as follows:

$$L(t; \mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}, \Omega) \equiv l(\Omega) + \boldsymbol{\lambda}^{\mathrm{T}}(A\mathbf{x} + B\mathbf{u} - \dot{\mathbf{x}}), \tag{6}$$

where the auxiliary Lagrange multipliers $\boldsymbol{\lambda}$ are also known as the adjoint states.

Now the dynamics is described by the following Euler–Lagrange variational equations:

$$\frac{\partial L}{\partial \boldsymbol{\lambda}} - \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\boldsymbol{\lambda}}}\right) = \mathbf{0}, \tag{7}$$

$$\frac{\partial L}{\partial \mathbf{x}} - \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial \dot{\mathbf{x}}}\right) = \mathbf{0}, \tag{8}$$

with the boundary conditions $\mathbf{x}_0 = \mathbf{0}$ and $\boldsymbol{\lambda}_T = \mathbf{0}$. As a result, from (7) and (8) we obtain the following state equation and the adjoint state equation:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}, \tag{9}$$

$$\dot{\boldsymbol{\lambda}} = -A^{\mathrm{T}}\boldsymbol{\lambda} - C^{\mathrm{T}}\frac{\partial l}{\partial \mathbf{y}} \tag{10}$$

$$= -A^{\mathrm{T}}\boldsymbol{\lambda} - C^{\mathrm{T}}\varphi(\mathbf{y}), \tag{11}$$

where $\varphi(\mathbf{y})$ is an n column vector of nonlinear activation functions.

Gradient methods can be used to solve the optimization problem. From (6) we can easily obtain the following derivatives with respect to the members of the parameter set Ω:

$$\frac{\partial L}{\partial A} = -\boldsymbol{\lambda}\mathbf{x}^{\mathrm{T}}, \tag{12}$$

$$\frac{\partial L}{\partial B} = -\boldsymbol{\lambda}\mathbf{u}^{\mathrm{T}}, \tag{13}$$

$$\frac{\partial L}{\partial C} = -\varphi(\mathbf{y})\mathbf{x}^{\mathrm{T}}, \tag{14}$$

$$\frac{\partial L}{\partial D} = D^{-T} - \varphi(\mathbf{y})\mathbf{u}^{\mathrm{T}}. \tag{15}$$

From above derivatives, we can obtain the following parameter estimation algorithm with respect to the members of the parameter set Ω:

$$\dot{A} = \eta_A \frac{\partial L}{\partial A} = -\eta_A \boldsymbol{\lambda}\mathbf{x}^{\mathrm{T}}, \tag{16}$$

$$\dot{B} = \eta_B \frac{\partial L}{\partial B} = -\eta_B \boldsymbol{\lambda}\mathbf{u}^{\mathrm{T}}, \tag{17}$$

$$\dot{C} = \eta_C \frac{\partial L}{\partial C} = -\eta_C \varphi(\mathbf{y})\mathbf{x}^{\mathrm{T}}, \tag{18}$$

$$\dot{D} = \eta_D \frac{\partial L}{\partial D} D^{\mathrm{T}} D = \eta_D (I - \varphi(\mathbf{y})\mathbf{u}^{\mathrm{T}} D^{\mathrm{T}}) D, \tag{19}$$

where η_X is a time-dependent learning rate with respect to X, X may be A, B, C or D. I is an $n \times n$ identity matrix. Note, to improve the performance of the parameter estimation algorithm, the technique of natural gradient [1] was employed in the parameter estimation algorithm of matrix D.

Note that in this method, we will need to obtain the solution of the state, and the adjoint state first. We need to solve (2a) for $\mathbf{x}$ with initial condition $\mathbf{x}_0 = 0$ from 0 to T, and obtain $\mathbf{y}$ from (2b). Then, we will need to solve the corresponding adjoint equation (11) with initial condition $\boldsymbol{\lambda}_T = 0$ from T to 0. Once $\boldsymbol{\lambda}$, $\mathbf{x}$ and $\mathbf{y}$ are obtained, the values of A, B, C, and D can be obtained. Using these new values of A, B, C, and D, we can enter another iteration of

[4]When the definition of MI (3) is used, the separate MI in each time step is added up. This addition does not take into account of possible correlation among the original source signals, and may overestimate the MI. An alternative criterion could be the coherence between different signals defined as in [8]. Since this measure requires the sampling frequencies of the estimated signals (which sometimes is not possible to obtain after an experiment is performed especially if the experiment is performed sometime in the past), we still propose to use (3) as the criterion in this paper.

[5]Note that for convenience we have assumed that the initial condition is $\mathbf{0}$. In general, $\mathbf{x}_0 \neq \mathbf{0}$. However, since we are studying a LTI system, this may be transformed to the case when $\mathbf{x}_0 = \mathbf{0}$.

solving for $\mathbf{x}$, $\mathbf{y}$, and $\boldsymbol{\lambda}$, and evaluate the values of A, B, C, and D. This will continue until the error tolerance on the parameters A, B, C, and D are smaller than a prescribed threshold or when the algorithm converges.

3. Balanced parameterization of linear systems

So far, we have assumed that the number of states $\overline{N}$ of the mixer is known, unfortunately, this is not necessarily true in practice. In this paper we will tackle this problem using a balanced parameterization of the demixer [11]. The balanced parameterization approach essentially transformed the state space model using a linear transformation such that the input "energy" and the output "energy" are the same. This is equivalent to considering the energy levels of the internal modes. Thus, if the assumed number of states is larger than the true number of states of the demixer, then the corresponding "energy" levels of the spurious internal modes will be close to zero and hence can be discarded.

Consider the continuous dynamical system (2), the system is balanced if there exist two identical $N \times N$ diagonal matrices P, Q, satisfying the following dual Lyapunov equations:

$$AP + PA^{\mathrm{T}} = -BB^{\mathrm{T}}, \tag{20}$$

$$A^{\mathrm{T}}Q + QA = -C^{\mathrm{T}}C, \tag{21}$$

where matrices P, Q are known as controllability Grammian and observability Grammian, respectively. The system is balanced when $P = Q$. In this case, their diagonal elements $\{\sigma_i\}_{i=1}^{N}$ are known as Hankel singular values. If we assume that the conditions $\sigma_1 \geqslant \cdots \geqslant \sigma_N > 0$ and $\sigma_{r+1} \ll \sigma_r$ ($r \in [1, N-1]$) are satisfied simultaneously, then it is possible to show that the order-reduced model of dimension r is asymptotically stable and minimal [6]. In addition, the behavior of the order-reduced model "approximates" that of the original LTI system [6].

The solution of the dual Lyapunov equation can be obtained if Ω is known. However, if these matrices are not known, or if they have to be estimated from observation data, then it would be very difficult to find Ω such that the dual Lyapunov equations are satisfied. This gives rise to research which attempts to find the parameterization of Ω such that the dual Lyapunov equations will be automatically satisfied. We will use a particular balanced parameterization, the one presented in [4] to model the continuous system (2). In this case, the parameters $\Theta \equiv \{\Sigma, B, D, \Phi\}$ can be obtained as follows:

$\Sigma : \{\sigma_1 \quad \sigma_2 \quad \cdots \quad \sigma_N\}$, the set of singular values,
satisfying $\sigma_1 > \sigma_2 > \cdots > \sigma_N$, and $\sigma_N > 0$;

$\Phi : \{\boldsymbol{\phi}_1 \quad \boldsymbol{\phi}_2 \quad \cdots \quad \boldsymbol{\phi}_N\}$, and $\boldsymbol{\phi}_j$ is an $(n-1)$ column vector,
its element $\phi_{pj} \in (-\pi/2, \pi/2)$, for $p = 1, 2, \ldots, n-1$;

$B : \{B_1 \quad B_2 \quad \cdots \quad B_N\}$, and B_j is n row vector with real values, the first element in the row is positive;

$D : n \times n$ real matrix.

From the balanced parameter set Θ, we can obtain the members (A, B, C, D) of the state space parameter set Ω as follows:

$$B : B^{\mathrm{T}} = \begin{bmatrix} B_1^{\mathrm{T}} & B_2^{\mathrm{T}} & \cdots & B_N^{\mathrm{T}} \end{bmatrix},$$
$$C : C = \begin{bmatrix} C_1 & C_2 & \cdots & C_N \end{bmatrix},$$
$$A : A = [A_{ij}] \text{ for } i,j = 1, 2, \ldots, N,$$
$$D : D = D,$$

where the jth column of C is constructed as

$$C_j = V(\boldsymbol{\phi}_j)\sqrt{B_j B_j^{\mathrm{T}}} \tag{22}$$

and the element n column vector of V is given as follows:

$$V(\boldsymbol{\phi}_j) = \begin{bmatrix} v_{1j} & v_{2j} & \cdots & v_{nj} \end{bmatrix}^{\mathrm{T}}, \tag{23}$$

where the details of v_{pj}, for $p = 1, 2, \ldots, n$, are given as

$$v_{pj} = \begin{cases} \cos\,\phi_{1j}\prod_{k=1}^{n-2}\cos\,\phi_{n-k,j} & \text{for } p = 1, \\ \sin\,\phi_{p-1,j}\prod_{k=1}^{n-p}\cos\,\phi_{n-k,j} & \text{for } 1 < p < n, \\ \sin\,\phi_{n-1,j} & \text{for } p = n. \end{cases} \tag{24}$$

The elements A_{ij} of the matrix A is given by the following:

$$A_{ij} = \begin{cases} -\dfrac{B_j B_j^{\mathrm{T}}}{2\sigma_j} & \text{for } i = j, \\ \dfrac{\sigma_j B_i B_j^{\mathrm{T}} - \sigma_i C_i^{\mathrm{T}} C_j}{\sigma_i^2 - \sigma_j^2} & \text{for } i \neq j. \end{cases} \tag{25a,b}$$

Under the above parameterization, the continuous time LTI system with parameter set Ω is balanced with Grammian $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_N)$, where diag is the operator of diagonalization.

4. Parameter estimation algorithm

In this section we will derive a balanced MBD parameter estimation algorithm with respect to the balanced parameter set Θ through optimizing the cost function (4). Treating the parameters Ω as the intermediate variables, it is possible to compute the gradients of the cost function with respect to the parameters Θ through the chain rule of differentiation. However, since the learning algorithms with respect to the parameters Ω have been given in Section 2, if we can obtain the relationship between Θ and Ω, the continuous time balanced MBD algorithm can be derived through combining these two parts of results.

4.1. The relationship between Θ and Ω

Since the matrices B and D are shared by the parameter sets Θ and Ω, we do not need to consider them here. What we need to do is to express the derivative of ϕ_{pj} and σ_j in terms of the derivative of the elements of Ω, for $p = 1, 2, \ldots, n-1$ and $j = 1, 2, \ldots, N$.

Consider (22) and (23), if we define $R_{Bj} \equiv \sqrt{B_j B_j^{\mathrm{T}}}$, we obtain:

$$\dot{C}_{p+1,j} = R_{Bj} \sum_{\ell=p}^{n-1} \xi_{kj}^{p+1} \dot{\phi}_{\ell j} + R_{Bj}^{-1} v_{p+1,j} B_j \dot{B}_j, \tag{26}$$

where we have used the definition $\xi_{kj}^{p+1} \equiv \partial C_{p+1,j} / \partial \phi_{kj}$, for $k = p, \ldots, n-1$, this gives:

$$\xi_{kj}^{p+1} = \begin{cases} -\sin \phi_{kj} \prod_{\ell=k+1}^{n-1} \cos \phi_{\ell j} \prod_{\ell=p}^{k-1} \cos \phi_{\ell j}, & k > p, \\ \prod_{\ell=p}^{n-1} \cos \phi_{\ell j}, & k = p. \end{cases}$$

From (26), we have the following relation:

$$\dot{C}_j^{n-1} = R_{Bj} \Delta_{(j)} \dot{\boldsymbol{\phi}}_j + R_{Bj}^{-1} B_j \dot{B}_j \mathbf{v}_j^{n-1}, \tag{27}$$

where $\dot{C}_j^{n-1} \equiv \begin{bmatrix} \dot{C}_{2j} & \dot{C}_{3j} & \cdots & \dot{C}_{nj} \end{bmatrix}^{\mathrm{T}}$, $\mathbf{v}_j^{n-1} \equiv [v_{2j} \ \cdots \ v_{nj}]^{\mathrm{T}}$, $\dot{\boldsymbol{\phi}}_j \equiv \begin{bmatrix} \dot{\phi}_{1j} & \cdots & \dot{\phi}_{n-1,j} \end{bmatrix}^{\mathrm{T}}$ and

$$\Delta_{(j)} \equiv \begin{bmatrix} \xi_{1j}^2 & \cdots & \xi_{n-1,j}^2 \\ & \ddots & \vdots \\ & & \xi_{n-1,j}^n \end{bmatrix}.$$

To express $\dot{\boldsymbol{\phi}}_j$ with $\dot{C}_j^{n-1}$ and $\dot{B}_j$ from (27), we require $\det(\Delta_{(j)}) \neq 0$. Observe $\det(\Delta_{(j)}) = \prod_{k=1}^{n-1} \xi_{kj}^{k+1}$, hence we need to satisfy $\xi_{k-1,j}^k \neq 0$ for all k. From the definition of $\xi_{k-1,j}^k$, we know that this is equivalent to satisfying $\xi_{1j}^2 \neq 0$. In other words, we need $\prod_{k=1}^{n-1} \cos \phi_{kj} \neq 0$. In Section 3, we have defined $\phi_{pj} \in (-\pi/2, \pi/2)$, this guarantees that the condition $\prod_{k=1}^{n-1} \cos \phi_{kj} \neq 0$ is satisfied.

From (27), we have the following relationship:

$$\dot{\boldsymbol{\phi}}_j = R_{Bj}^{-1} \Delta_{(j)}^{-1} \left(\dot{C}_j^{n-1} - R_{Bj}^{-1} B_j \dot{B}_j \mathbf{v}_j^{n-1} \right). \tag{28}$$

Note that in (28), we need to compute an inverse matrix $\Delta_{(j)}^{-1}$, however, it is trivial because $\Delta_{(j)}$ is an upper-triangular matrix.

For the parameter set Σ, the derivation is relatively simple. Consider the diagonal elements of matrix A in (25), we can easily obtain $\partial A_{jj} / \partial \sigma_j = R_{Bj}^2 / 2\sigma_j^2$. Combining the chain rule $\dot{A}_{jj} = (\partial A_{jj} / \partial \sigma_j) \dot{\sigma}_j$, we obtain:

$$\dot{\sigma}_j = 2 R_{Bj}^{-2} \sigma_j^2 \dot{A}_{jj}. \tag{29}$$

4.2. The parameter estimation algorithm

Once the relationship between Θ and Ω has been obtained, it is easy to derive continuous time balanced MBD algorithms based on the general state space algorithms shown in Section 2. Since the matrices B and D are shared by the parameter sets Θ and Ω, their learning algorithms are still those derived in Section 2. Now we will derive the learning algorithms with respect to the parameters Φ and Σ.

Substituting (18) and (17) into (28), we obtain the following learning algorithm with respect to the parameters Φ:

$$\dot{\boldsymbol{\phi}}_j = \frac{1}{\sqrt{B_j B_j^{\mathrm{T}}}} \Delta_{(j)}^{-1} \left(\dot{C}_j^{n-1} - \frac{B_j \dot{B}_j}{\sqrt{B_j B_j^{\mathrm{T}}}} \mathbf{v}_j^{n-1} \right) \tag{30}$$

$$= -\eta_\phi \frac{\Delta_{(j)}^{-1}}{\sqrt{B_j B_j^{\mathrm{T}}}} \left[\varphi(\mathbf{y}_{n-1}) x_j^{\mathrm{T}} + \eta_{\mathrm{r}} \frac{B_j \lambda_j \mathbf{u}^{\mathrm{T}}}{\sqrt{B_j B_j^{\mathrm{T}}}} \mathbf{v}_j^{n-1} \right], \tag{31}$$

where η_{r} is the relative learning rate, which is used due to the fact that the learning rates for B and C in (18) and (17) are not necessarily identical. Substituting (16) into (29), we obtain the following learning algorithm with respect to the parameters Σ:

$$\dot{\sigma}_j = -\eta_\sigma \frac{\sigma_j^2}{B_j B_j^{\mathrm{T}}} \lambda_j x_j^{\mathrm{T}}. \tag{32}$$

Finally, we have the following continuous time MBD parameter estimation algorithm:

$$\dot{D} = \eta_D \left(I - \varphi(\mathbf{y}) \mathbf{u}^{\mathrm{T}} D^{\mathrm{T}} \right) D, \tag{33}$$

$$\dot{B} = -\eta_B \boldsymbol{\lambda} \mathbf{u}^{\mathrm{T}}, \tag{34}$$

$$\dot{\boldsymbol{\phi}}_j = -\eta_\phi \frac{\Delta_{(j)}^{-1}}{\sqrt{B_j B_j^{\mathrm{T}}}} \left[\varphi(\mathbf{y}_{n-1}) x_j^{\mathrm{T}} + \eta_r \frac{B_j \lambda_j \mathbf{u}^{\mathrm{T}}}{\sqrt{B_j B_j^{\mathrm{T}}}} \mathbf{v}_j^{n-1} \right], \tag{35}$$

$$\dot{\sigma}_j = -\eta_\sigma \frac{\sigma_j^2}{B_j B_j^{\mathrm{T}}} \lambda_j x_j^{\mathrm{T}}, \tag{36}$$

where η_X, X may be D, B, ϕ, or σ, is a time-dependent learning rate with respect to X.

Similar to the general continuous time state space MBD algorithm described in Section 2.2, in the proposed continuous time balanced algorithm, to learn B, Φ, and Σ, we need the bidirectional updates for the state $\mathbf{x}$ and the adjoint state $\boldsymbol{\lambda}$ described in (9) and (11). The implementation of the proposed continuous time balanced MBD algorithm is summarized in Algorithm 1 for ease of reference.

Algorithm 1. The continuous time balanced MBD algorithm.

1: Initialize the balanced parameter set Θ
2: **while** $\Delta l >$ tolerance OR $\Delta\lambda_0 \equiv \sum_{k=1}^{N} \lambda_k(0) >$ tolerance **do**
3: Construct the parameter set Ω from the balanced parameter set Θ
4: Obtain the outputs $\mathbf{y}_T$ by running the follows from 0 to T: $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$ AND $\mathbf{y} = C\mathbf{x} + D\mathbf{u}$
5: Obtain adjoint states $\boldsymbol{\lambda}$ by running the follows from T to 0: $\dot{\boldsymbol{\lambda}} = -A^{\mathrm{T}}\boldsymbol{\lambda} - C^{\mathrm{T}}\varphi(\mathbf{y})$
6: Learn the parameter set Θ using (33)–(36)
7: Calculate Δl, and$\Delta\lambda_0$
8: **end while**

Table 1
The comparison of number of parameters needed to learn between the general continuous time MBD algorithm [5] and the proposed continuous time balanced algorithm

	General [5]		Algorithm 1	
Parameters	A	C	Σ	Φ
No. of parameters	$N \times N$	$n \times N$	$N \times 1$	$(n-1) \times N$
Total	$(n+N) \times N$		$n \times N$	

4.3. Remarks

- The general continuous time MBD algorithm [5] used in this paper and the proposed continuous time balanced MBD algorithm are both derived through constrained optimization method, thus we can call them continuous time constrained MBD algorithm and continuous time balanced constrained MBD algorithm, respectively.
- Compared to the general continuous time MBD algorithm [5], besides possessing the ability of estimating the number of states, the proposed continuous time balanced algorithm has another advantage: the parameters needed to be estimated are less than those in its general state space counterparts. The comparison of numbers of parameters to be estimated between these two approaches are listed in Table 1. We observe that the difference of number of parameters in these two approaches are N^2. This reduced number of parameters may be significant when N is relatively large. Note we do not include D and B in the table as they are shared by both the balanced and the general approaches.

5. Experiments

In this section, we will evaluate the proposed continuous time balanced MBD algorithm (Algorithm 1) through a number of experiments using practical signals. In Section 5.1 we will give three performance measurement indices which will be used in this paper. In Section 5.2 we will examine the proposed algorithm using a set of artificially mixed speech observations. In particular, we will compare the performance of the proposed algorithm with the general continuous time state space algorithm [5]; then we will examine the balanced algorithm's ability of estimating the number of states of the mixer. In Section 5.3, we will apply the proposed algorithm to a set of EEG data. We will compare the performance of the balanced algorithm with a public domain EEG oriented package; using the estimated the number of states of the mixer ($\overline{N}$), we will show that, after $\overline{N}$ is obtained, the performance of the general continuous time state space algorithm [5] is superior to that of the EEG package.

5.1. Performance measurements

In this paper, we compare different MBD algorithms to evaluate their relative performances using some performance indices. There exist many possible performance indices [10,12] in the literature, we choose two particular ones: the mean square error (MSE) criterion and the cross-talk (X_{talk}) criterion from such set of possible candidates. Both the mean square error criterion and the cross-talk criterion evaluate the similarity between the waveforms of the source signals and their recovered version.

Given T samples of n-channel source signals $\mathbf{s}$ and their recovered signals $\mathbf{y}$, the MSE criterion and the cross-talk (X_{talk}) criterion are defined, respectively, as follows [3]:

$$\text{MSE}_i \equiv \frac{1}{T}\sum_{t=0}^{T-1}\left[y_i(t) - s_i(t)\right]^2 \quad \text{for } i = 1, \ldots, n,$$

$$X_{\text{talk}} \equiv \frac{1}{T(n^2-n)}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n}\sum_{t=0}^{T-1}\left[|y_i(t) - s_i(t)|s_j(t)\right].$$

The above two criteria require the source signals to be available, however, generally speaking, this cannot be satisfied in practice. Since the objective of MBD is usually accomplished through minimizing the MI between the outputs of the demixer, therefore, MI is a valid criterion for measuring the performance of MBD algorithms. Normally MI is difficult to compute. Moddemeijer [9] provided a practical approach to compute the MI between two signals. In this paper, we also adopt MI as our performance index. Moddemeijer's approach [9] only allows to compute the MI between two waveforms, and in our case we need to compute the average mutual information (AMI) for the situation where the source number is $\geqslant 2$. Given the number of sources $n>2$, the AMI can be defined as follows:

$$\text{AMI} \equiv \frac{1}{2(n^2-n)}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}|\text{MI}_{ij}|,$$

where $|\text{MI}_{ij}|$ is the absolute value of the MI between the ith signal and the jth signal.

5.2. Speech data

Ten thousand samples of two-column observation data (Fig. 1(b)) are obtained by passing two digital speech sources (Fig. 1(a)) through a stable discrete time LTI causal dynamical system with 2 states, thus in this mixing environment, $n = 2$, $\overline{N} = 2$. This discrete time mixer can be described as follows:

$$\mathbf{x}_{t+1} = \overline{A}_d\mathbf{x}_t + \overline{B}_d\mathbf{u}_t, \tag{37a}$$

$$\mathbf{y}_t = \overline{C}_d\mathbf{x}_t + D_d\mathbf{u}_t. \tag{37b}$$

The system matrices of the mixer are chosen such that (I) $\overline{D}_d^{-1}$ exists, (II) the eigenvalues of the $\overline{A}_d$ are all within the unit circle, and (III) $\overline{B}_d$ and $\overline{C}_d$ are randomly selected. In details, the four system matrices of the mixer are:

$$\overline{A}_d = \begin{bmatrix} 0.6128 & -0.3591 \\ -0.2523 & 0.4515 \end{bmatrix}, \quad \overline{B}_d = \begin{bmatrix} 0.5194 & 0.1519 \\ 0.1161 & 0.2722 \end{bmatrix},$$

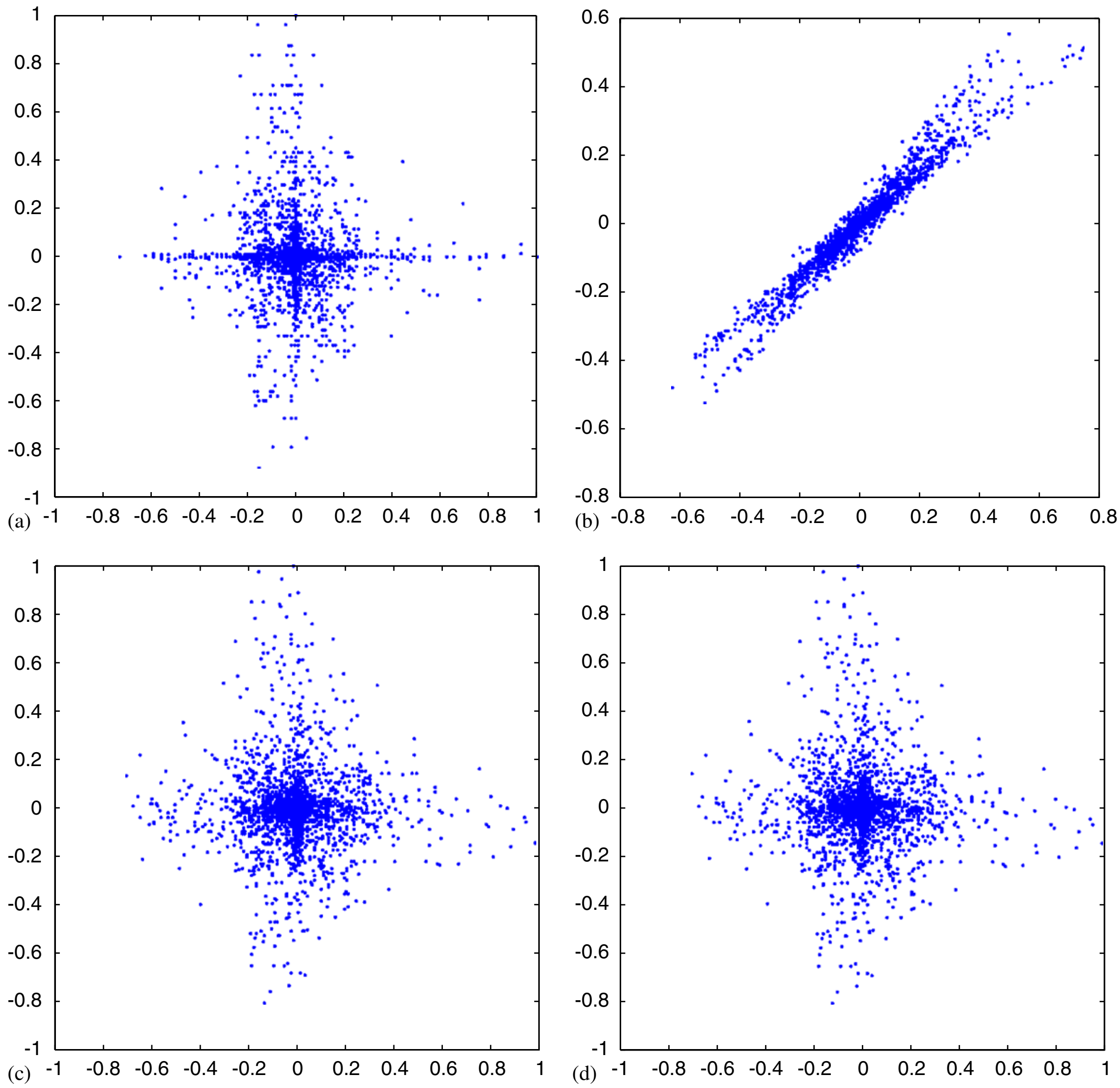

Fig. 1. The scatter diagrams of: (a) the sources; (b) the observations; (c) the separated signals obtained from general continuous time algorithm [5]; (d) the separated signals obtained from the proposed continuous time balanced algorithm.

$$\overline{C}_d = \begin{bmatrix} 0.4166 & 0.2009 \\ 0.3593 & 0.1209 \end{bmatrix}, \quad \overline{D}_d = \begin{bmatrix} 0.6757 & 0.4015 \\ 0.4441 & 0.4765 \end{bmatrix}.$$

Since the observations are given in discrete time domain, we need to discretize the continuous time demixer. In this paper, we employ the Runge–Kutta algorithm to realize this. The sampling frequency is set to 8000 Hz. The way of discretizing a continuous time LTI causal dynamical system using Runge–Kutta algorithm is specified in the Appendix.

Note that we could have used a continuous time dynamic mixer to mix the source signals, and then to pass such continuous time observations to a continuous time demixer. However, in practice, most modern practical observations are discrete in time even though the mixer is a continuous time system. In such case, we could have used interpolation techniques to find the intermediate points of the observations. But in practice because of the advances in modern electronics, it is far more prevalent to use a discrete time demixer rather than a continuous time demixer (as analog devices suffer notoriously from drifts). It would be much easier to approximate a continuous time demixer using standard methods of integration, e.g., Runge–Kutta method. Hence, we have chosen to use a discrete time system to evaluate our proposed algorithms.

Performance comparison: First we compare the proposed continuous time balanced MBD algorithm with the general continuous time MBD algorithm [5]. The purpose of this experiment is to investigate if the proposed algorithm has similar performance as that of the one proposed using general state space formulations [5]. Since the proposed algorithm is based on the estimation of the A, B, C, and D parameters from the measurements, it does not necessarily yield similar

Table 2
The mean and the variance of the mean square errors, cross-talks, and mutual information given by the general continuous time MBD algorithm [5] and the proposed balanced MBD algorithm

	General [5]		Algorithm 1	
Channel	I	II	I	II
MSE(M)	0.0012	0.0020	0.0010	0.0015
MSE(V)	9×10^{-7}	3×10^{-7}	5×10^{-7}	8×10^{-7}
$X_{\text{talk}}(M)$	0.0004		0.0004	
$X_{\text{talk}}(V)$	1×10^{-7}		2×10^{-7}	
MI(M)	0.0593		0.0443	
MI(V)	3×10^{-4}		6×10^{-4}	

Note: $X(M)$ denotes the mean, and $X(V)$ denotes the variance.

results compared with those obtained using a general state space formulation [5].

In both cases, we use a demixer with a high enough dimension of states ($N = 6$), just like suggested in [13], to separate the signals. With 10 sets of randomly chosen initial parameters, we run the two algorithms 10 times. The mean and the variance of the MSE, residual cross-talk (X_{talk}), and MI are shown in Table 2. As an example, the recovered signals from a particular run of the two approaches are plotted in Fig. 1(c) and (d), respectively. It is noted that from these scatter diagrams that the proposed balanced parameterization algorithm works well. From Table 2 we can observe that, in the sense of MSE, cross-talk and MI, the performance of the algorithms are comparable for this simple MBD problem. This indicates that the proposed algorithm is working well, yielding comparable performance to that obtained from the general state space formulation [5]. This is reassuring as it indicates that the proposed algorithm comparatively performs well.

Determining the dimension of states: Next, we will examine the ability of selecting state number of the proposed continuous time balanced algorithm. We assume that the number of states $\overline{N}$ of the mixer is unknown, but we believe that the current number of states N of the demixer is greater than $\overline{N}$. We study two situations: $N = 4$ and 6. The evolution of the singular values for the two cases is plotted in Fig. 2. We observe that in both cases there is a big gap between the second largest singular values and the remaining singular values, which are all of similar order. The results show that the continuous time balanced approaches can identify the correct number of states quite well. This can be further verified by evaluating the relative weight of the first two principal singular values as a function of the total "amount of energy" in the system, which are respectively 94% for $N = 4$ case, and 91% for $N = 6$ case. From this experiment, it may be concluded that the number of states is 2. This indicates that the proposed algorithm can obtain the number of states.

Thus from this set of experiments, we can conclude that (i) the proposed algorithm has similar performance to the one proposed in [5], and (ii) in addition, it can determine the number of states in the demixer. What is not apparent from this set of experiments is that the proposed algorithm is numerically more robust than the one proposed in [5]. Obviously once the number of states is determined, using our proposed algorithm, the two algorithms will have comparatively similar results using the correct number of states. It would be expected that if we compare the two algorithms, the one proposed in [5] using the wrong number of states, say, $N = 6$, and our proposed algorithm using the correctly determined number of states $N = 2$, then one would expect to observe some difference in performance. This thought experiment simulates the situation when we use both algorithms on observations which we do not know a priori the number of states. Using the algorithm proposed in [5], if the wrong number of states is guessed, say, $N = 6$, and in our proposed algorithm, where we can determine the number of states, in this case $N = 2$, then one would expect to observe some differences in the performance. We have carried out the experiments but we will not present the results here as it is as expected, and secondly in the next section in the EEG experiments, we have carried out the same type of experiments and they confirm the anticipated results.[6]

5.3. Electroencephalograph data

In this experiment, we will apply the proposed algorithm to a set of EEG signals. Our objective is to show how to use the proposed balanced algorithm to estimate the number of states in a mixing system whose dynamics is unknown.

The 2500 samples (10*s*) of the seven-channel EEG data (see Fig. 3) were recorded by Z. Keirn at Purdue University using a sampling frequency of 250 Hz.[7] To realize our objective, we assume that the given set of observation data is generated by a mixer with seven inputs and seven outputs. The number of states $\overline{N}$ of the mixer is unknown which needs to be estimated. We assume that $\overline{N}$ is less than 6, with 10 sets of randomly chosen initial parameters, we run the proposed balanced algorithm 10 times with the number of states $N = 6$. The mean and the variance of the AMI is shown in Table 3. As an example, the recovered signals from a particular run are plotted in Fig. 6, and the evolution of the singular values is plotted in Fig. 5. Note the recovered signals in Fig. 6 is re-ordered for the convenience of comparison. This is also applied to Figs. 4, 7, and 8 (Figs.Figs. 4–8).

[6] Note that in the EEG experiments, we carried out a version of this thought experiment, and found that indeed the proposed algorithm performs better than the algorithm proposed in [5]. Please compare the AMI result using $N = 6$ for the algorithm proposed in [5], and the AMI results using our proposed algorithm with the correctly determined number of states $N = 3$ in Table 3. However, to be sure, we performed the additional experiment reported in this section just to show that our proposed algorithm indeed performs better.

[7] The EEG recordings and the corresponding information can be found at the website of http://www.cs.colostate.edu/~anderson/res/eeg/.

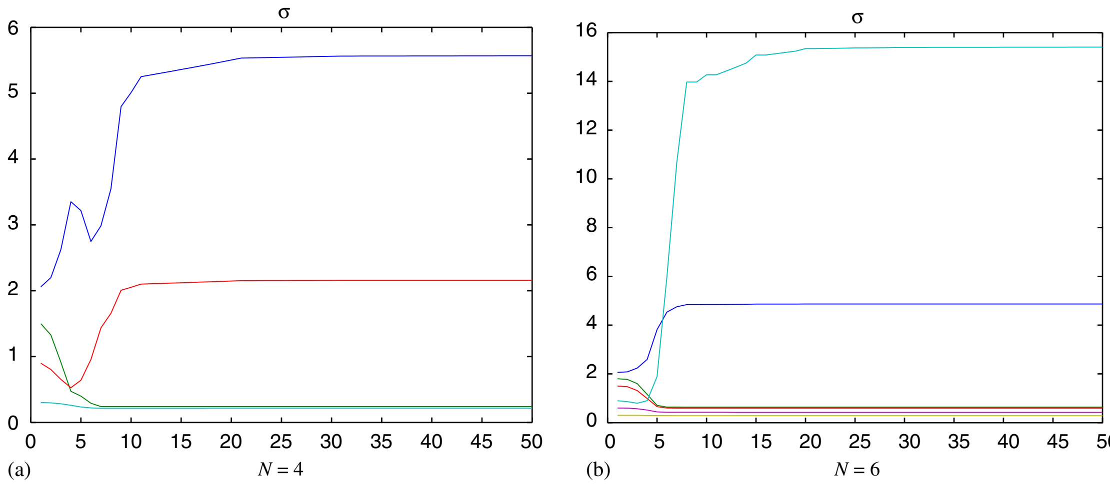

Fig. 2. The evolution of singular values in the proposed continuous time balanced algorithm (speech case), with two different assumed number of states. In both figures, the x-axis represents the number of iterations of learning, the y-axis represents the value of the Hankel singular value, each line describes the change of each singular value with the number of iterations of learning. When the proposed algorithm converges, only two of the singular values remain dominant.

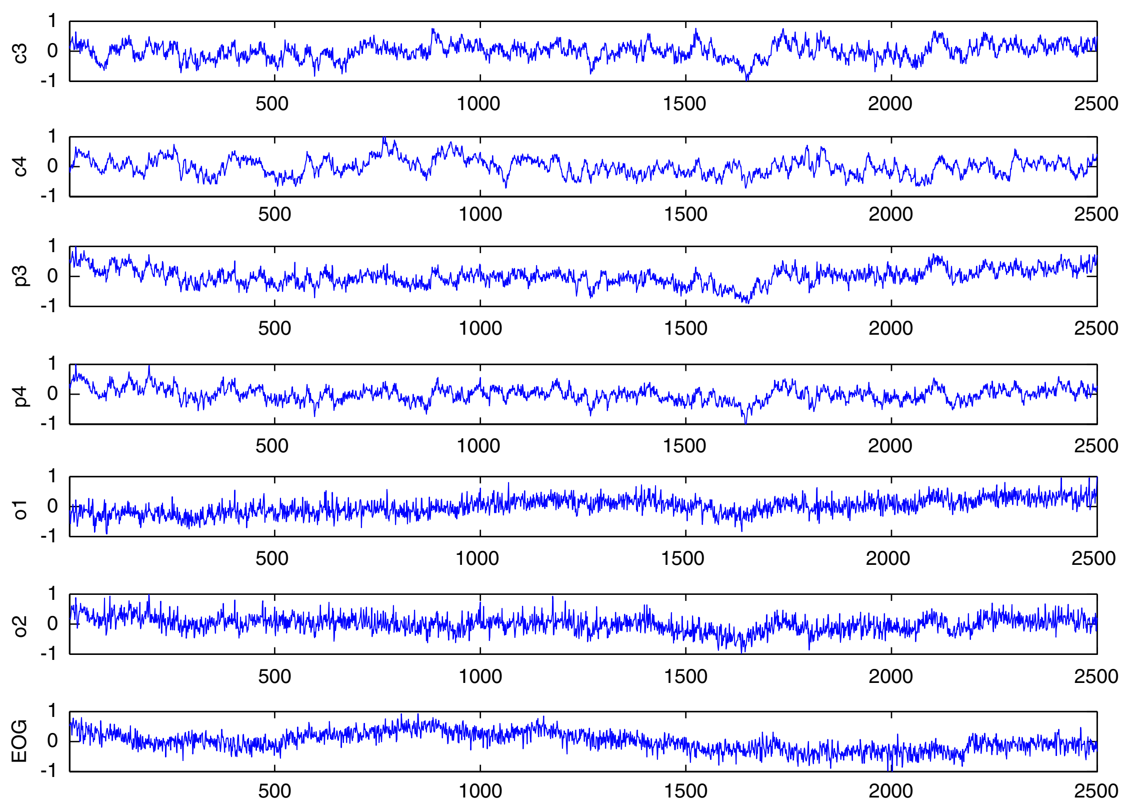

Fig. 3. The original seven channels of EEG recordings.

Table 3
The mean and the variance of the average mutual information of the recovered signals obtained from the various algorithms

	EEG toolbox	Algorithm 1 ($N = 6$)	Algorithm [5] ($N = 3$)	Algorithm 1 ($N = 3$)
AMI(M)	0.0078	0.0063	0.0042	0.0038
AMI(V)	4×10^{-8}	11×10^{-8}	14×10^{-8}	8×10^{-8}

Note: $X(M)$ denotes mean and $X(V)$ denotes variance.

To evaluate the performance of the proposed balanced algorithm, we employ a publicly available software package: the EEG toolbox 3.61[8] as a basis of comparison. This package does not include a dynamic mixer, but assumes a static mixer. Again we run the package 10 times.

[8]The EEG toolbox was released by S. Makeig et al. in CNL/Salk Institute, U.S. It can be downloaded from the world wide web at http://www.cnl.salk.edu/~scott/icabib.html.

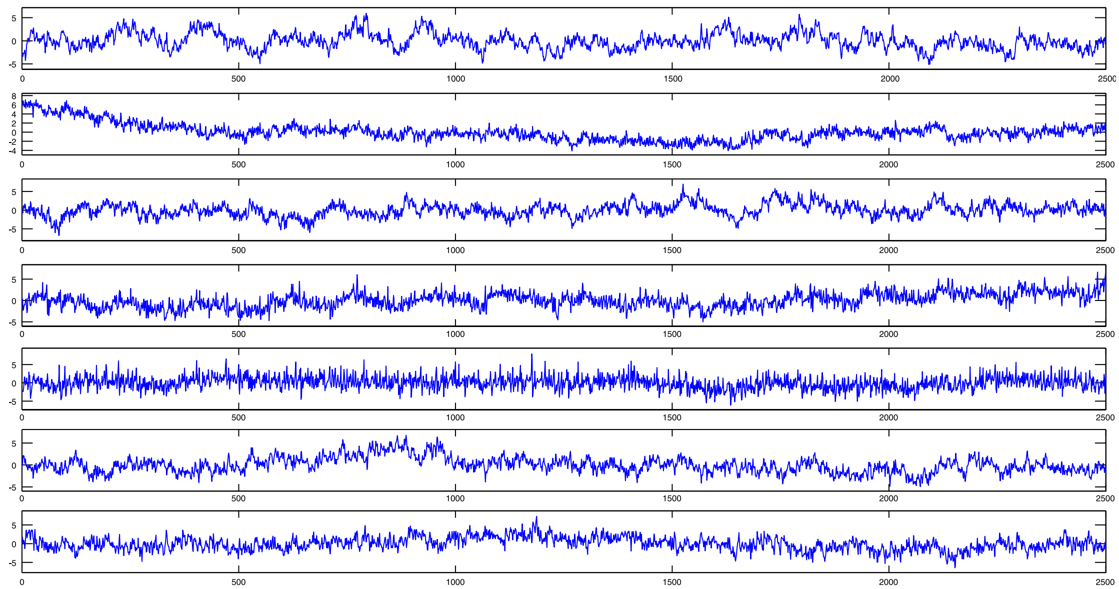

Fig. 4. The recovered EEG signals obtained by using the extended infomax ICA algorithm in EEG toolbox 3.61.

The mean and the variance of the AMI of the recovered signals are shown in Table 3. As an example, the recovered signals obtained from the EEG toolbox in a particular run are plotted in Fig. 4. From Table 3, we observe that the performance of the proposed algorithm is better than the EEG toolbox. From Fig. 5, we observe that the proposed balanced algorithm draws the conclusion that the number of states $\overline{N}$ is 3.

To examine the validity of above number of states estimation, we run the general continuous time algorithm [5] and Algorithm 1 with the number of states $N = 3$. Similarly, we run both the algorithms 10 times with random chosen initial parameters. The mean and the variance of the AMI of the recovered signals is shown in Table 3. In a particular run, the recovered signals are plotted in Figs. 7 and 8, respectively. From Table 3, we observe that, compared to the EEG toolbox and the balanced algorithm with a higher number of states, the performance of either the general continuous time algorithm [5] or Algorithm 1 with the estimated "correct" number of states[9] is superior.

This implies the following: we can use our proposed algorithm to find the number of states. Once we know the number of states, then we can use Algorithm 1 or the algorithm as proposed by [5] to recover the unknown source signals. Note that if we apply the algorithm as proposed by [5] directly without knowing the true number of states, the performance would be worse. However, if the correct number of states is known, for example, as determined by our proposed algorithm, then the performance of the algorithm as proposed in [5] is improved. To verify this conclusion, we run the algorithm in [5] with all

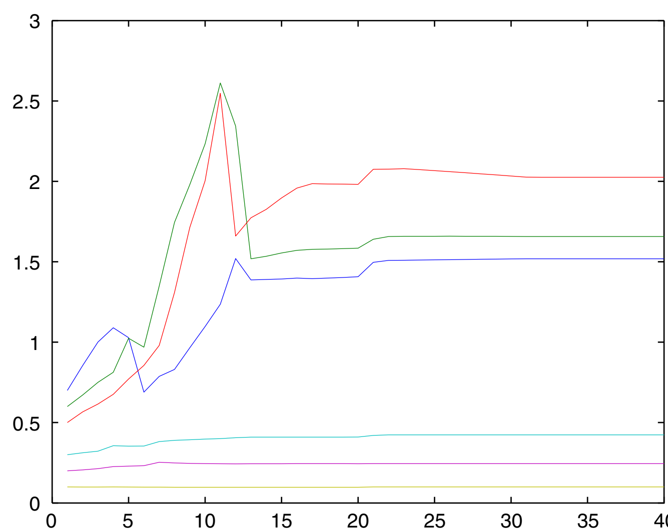

Fig. 5. The evolution of singular values in the proposed balanced algorithm (EEG case), with assumed state number $N = 6$. The x-axis represents the number of iterations of learning, the y-axis represents the value of the Hankel singular value, each line describes the change of each singular value with the number of iterations of learning. When the proposed algorithm converges, among the six singular values, three are principal.

number of states from $N = 1$ to 6, in each case, we run the algorithm 10 times with random chosen initial parameters. The mean and the variance of the AMI of the recovered signals is shown in Table 4. It shows that the minimum of the AMI appears when the number of states is 3. This somehow surprising result may be partially explained as follows: the mixer is assumed to be an asymptotically stable, LTI, causal dynamical system, the ideal demixer, which is the inverse filter of the mixer, should possess the

[9]Note that in this case, we do not know the number of states. The "correct" number of states is as determined by our algorithm.

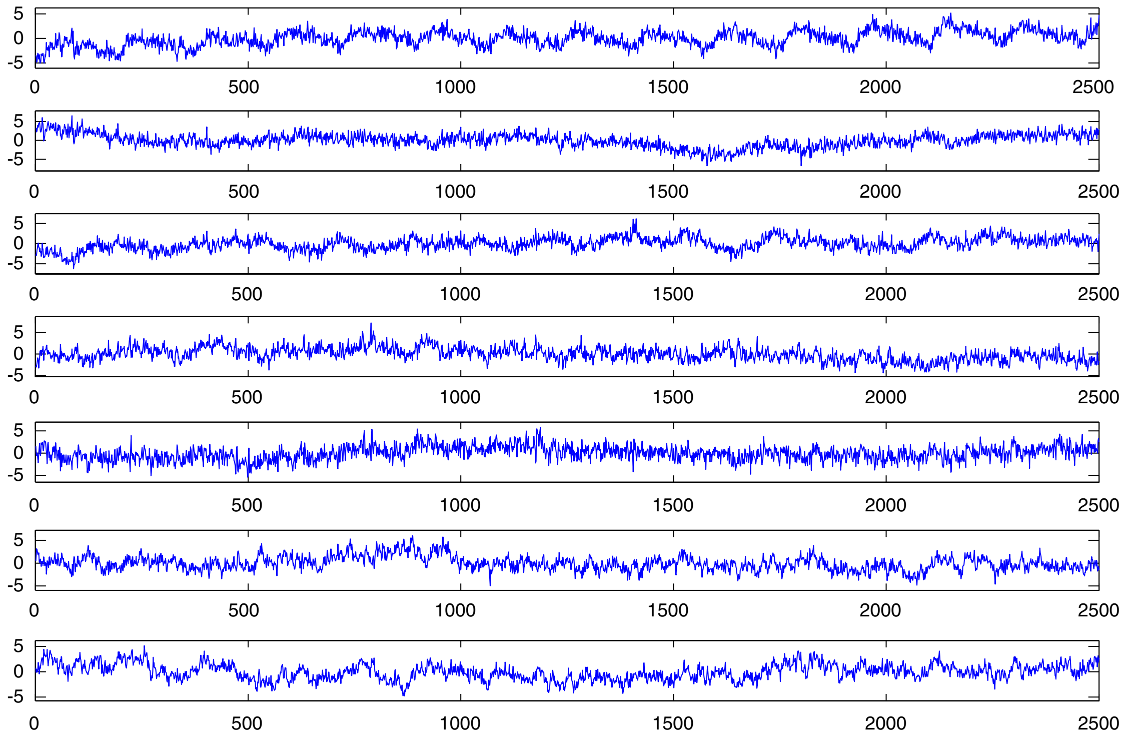

Fig. 6. The recovered EEG signals obtained from the proposed balanced algorithm, $N = 6$ case.

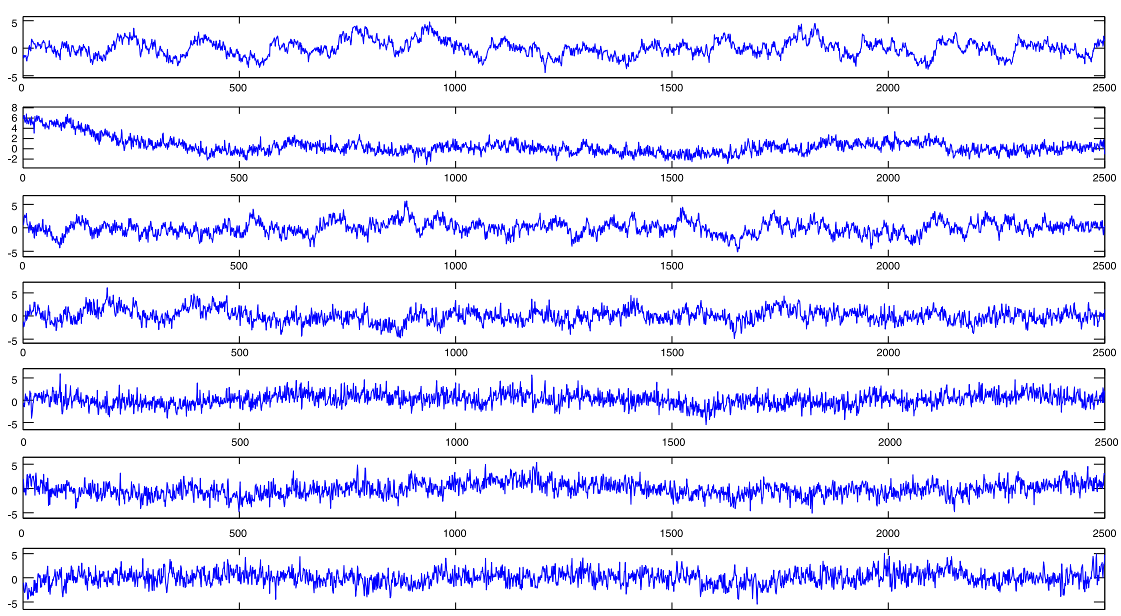

Fig. 7. The recovered EEG signals obtained from the general continuous time algorithm [5], $N = 3$ case.

same dimension of states as the mixer. Since there exist some ‘‘distances’’ between the proposed demixers and the ideal demixer, as the proposed demixers are only an estimate of the ideal demixer, which is an exact inverse of the mixer, the performance of the demixers with the ‘‘wrong’’ number of states is inevitably worse than the ideal demixer. To avoid the influence of the characteristic of a special set of measurement data, in a similar manner, we run the algorithms on another four sets of EEG data recorded by the same researcher (Z. Keirn), similar results are observed. These results will not be reported here as they do not provide additional information or added insights into the proposed algorithm not already provided by the above experiments.

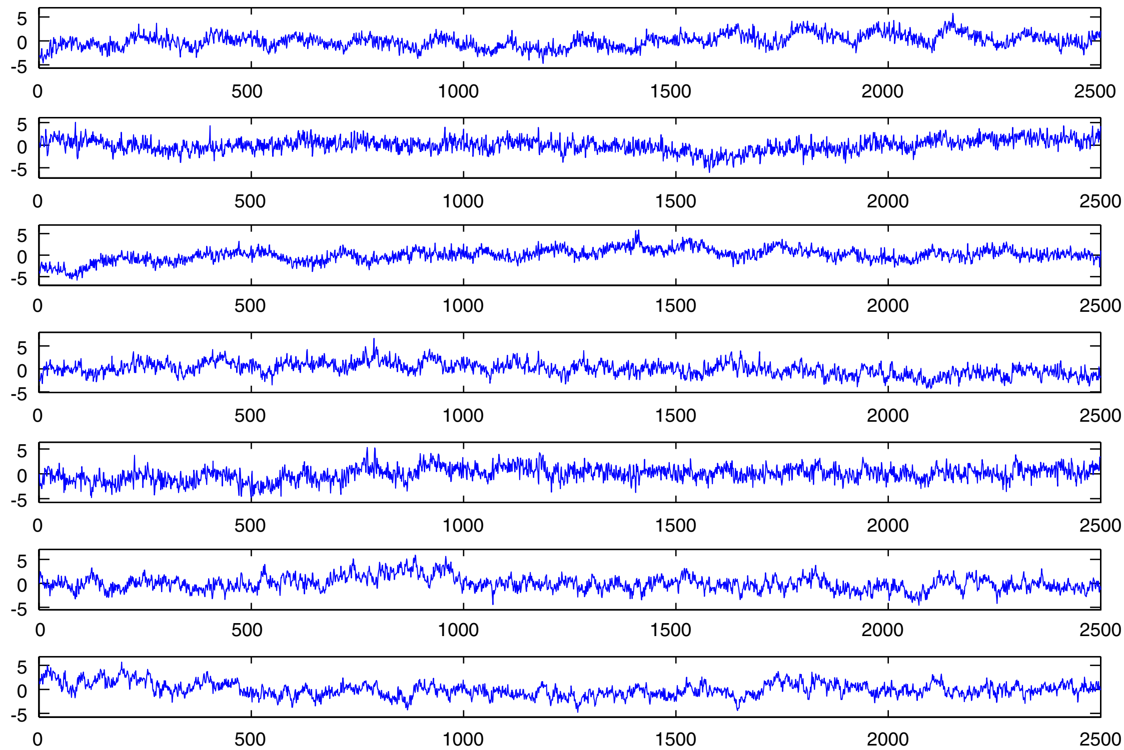

Fig. 8. The recovered EEG signals obtained from the proposed balanced algorithm, $N = 3$ case.

Table 4
The mean and the variance of the average mutual information of the recovered signals obtained from the continuous time algorithm [5] with different assumed number of states

	$N = 1$	$N = 2$	$N = 3$	$N = 4$	$N = 5$	$N = 6$
AMI(M)	0.0057	0.0054	0.0042	0.0058	0.0060	0.0063
AMI(V)	12×10^{-8}	16×10^{-8}	14×10^{-8}	17×10^{-8}	13×10^{-8}	10×10^{-8}

Note: $X(M)$ denotes mean and $X(V)$ denotes variance.

6. Conclusion

In this paper we have studied the problem of MBD in the continuous time domain. We first reviewed the general continuous time state space algorithm presented in [5], which was derived using a constrained optimization method after introducing the Lagrange multipliers as adjoint state vectors. Then using similar techniques, we have derived a continuous time balanced MBD algorithm.

The advantages of the proposed continuous time balanced MBD algorithm are as follows: compared to the discrete time balanced algorithm derived in [11], it does not require to perform a bilinear transformation, thus it is more efficient;[10] moreover, it can estimate the number of states; also the number of parameters which needs to be estimated is less.

We have used the standard Runge–Kutta integration method to approximate the continuous time demixer. In some instances this approximation may not be necessary, if the demixer is implemented by analog devices, or by a digital device with a high sampling rate relative to the underlying natural frequencies of the system.

There remains some questions, which would be interesting to consider in future research. One such question is: would the separation of the source signals mixed using a continuous time mixer be better than the one formulated in discrete time domain. For this purpose an experimental comparison between both domains with synthetical quasi-continuous data (high sampling rate compared to the output rate) would have been of interest. A second question is: if both the source signals and the mixer are continuous time ones, would a continuous time demixer perform better than a discrete time one. These questions are reserved as directions of future research.

[10]Bilinear transformations are required to compute an inverse matrix with the dimension of N, therefore, the computational load becomes large if N is relatively high. It would be interesting to compare the measured computation times between Algorithm 1 and the discrete time balanced algorithm [11], however, it is not within the scope of this paper and hence this is proposed as a matter for future research.

Acknowledgement

The authors wish to acknowledge constructive comments made by the anonymous reviewers which lead to improvement of an earlier version of this paper. In addition, the first author wishes to acknowledge the partial financial support from the Australian Research Council through a Linkage Project grant. The authors further acknowledges the partial financial support provided by Motorola Australia Research Centre. This work was performed while the second author was with University of Wollongong.

Appendix

Fourth order Runge–Kutta algorithm is one of the most powerful predictor-corrector algorithms which allows us to numerically solve differential equations. Given a differential equation $\dot{y} = f(t, y)$ with the initial condition $y(t_0) = y_0$. Suppose h is a small interval with respect to t, at the point of $t_{k+1} = t_k + h$, the Runge–Kutta algorithm gives:

$$y_{k+1} = y_k + 1/6(r_1 + 2r_2 + 2r_3 + r_4), \tag{38}$$

where

$$r_1 = hf(t_k, y_k), \quad r_2 = hf(t_k + h/2, y_k + k_1/2), \tag{39}$$

$$r_3 = hf(t_k + h/2, y_k + k_2/2), \quad r_4 = hf(t_k + h, y_k + k_3). \tag{40}$$

References

[1] S. Amari, Natural gradient working efficiently in learning, Neural Comput. 10 (2) (1998) 251–276.

[2] A.E. Bryson, Y.-C. Ho, Applied Optimal Control: Optimization, Estimation, and Control, Hemisphere Pub. Corp, Washington, 1975.

[3] R.A. Choudrey, Variational Methods for Bayesian independent component analysis, Ph.D. Thesis, University of Oxford, 2002.

[4] C.T. Chou, J.M. Maciejowski, System identification using balanced parametrizations, IEEE Trans. Automation Control 42 (7) (1997) 956–974.

[5] G. Erten, F. Salam, Voice extraction by on-line signal separation and recovery, IEEE Trans. Circuits Systems II: Analog Digital Signal Process. 46 (7) (1999) 915–922.

[6] K. Glover, All optimal Hankel norm approximations of linear multivariable systems and L^∞ error bounds, Int. J. Control 39 (1984) 1115–1193.

[7] T. Kailath, Linear Systems, Prentice-Hall, Englewood Cliffs, NJ, 1980.

[8] P.P. Mitra, B. Pesaran, Analysis of dynamic brain imaging data, Biophys. J. 76 (1999) 691–708.

[9] R. Moddemeijer, On estimation of entropy and mutual information of continuous distributions, Signal Process. 16 (3) (1989) 233–246.

[10] A.K. Nandi, (Ed.), Blind Estimation Using Higher-Order Statistics, Kluwer Academic Publishers, Dordrecht, 1999.

[11] A.C. Tsoi, L.S. Ma, Blind deconvolution of dynamical system using a balanced parameterised state space approach, in: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 4, Hongkong, 2003, pp. 309–312.

[12] K. Waheed, F.M. Salam, A data-derived quadratic independence measure for adaptive blind source recovery in practical applications, in: 45th IEEE International Midwest Symposium on Circuits and Systems, vol. 3, 2002, pp. 473–476.

[13] L. Zhang, A. Cichocki, Blind deconvolution of dynamical systems: a state space approach, J. Signal Process. 4 (2) (2000) 111–130.

Liangsuo Ma was born in China. He received the B.E. degree in electrical engineering from Tsinghua University, China, in 1995. He received the M.E. and Ph.D. degrees in electrical, computer & telecommunication engineering from University of Wollongong, Australia in 2000 and 2005, respectively. Since December 2004, he has been working as a postdoctoral research fellow at the University of Iowa, United States.

Ah Chung Tsoi was born in Hong Kong. He received the Higher Diploma in Electronic Engineering from Hong Kong Technical College in 1969, and the M.Sc. degree in electronic control engineering, and Ph.D. degree in control engineering from University of Salford in 1970 and 1972, respectively. He also received a B.D. degree from University of Otago in 1980.

From 1972 to 1974 he was a Senior Research Fellow at the Inter-University Institute of Engineering Control, University College of North Wales, Bangor, Wales. From 1974 to 1977 he was a Lecturer at the Paisley College of Technology, Paisley Refrewshire, Scotland. From 1977 to 1985 he was a Senior Lecturer at the University of Auckland, New Zealand. From 1985 to 1990, he was a Senior Lecturer at the University College, University of New South Wales. From 1990 to 1996, he was an Associate Professor, and then a Professor in Electrical Engineering at the University of Queensland, Australia. While at the University of Queensland, Professor Tsoi led a large R & D project on speaker verification. From July 1996 to February 2004 he was at the University of Wollongong, where he had been Dean, Faculty of Informatics (1996–2001), and Director of Information Technology Services (1999–2001). From February 2001 to 2004, he was the foundation Pro Vice Chancellor (Information Technology and Communications) at the University of Wollongong. In the role of Pro Vice-Chancellor (Information Technology and Communications) he initiated a number of infrastructure projects: gigabit ethernet for the campus, voice over IP telephony, wireless access for the entire campus, implementation of a student management package. His research interests include aspects of neural networks and fuzzy systems and their application to practical problems, adaptive signal processing, speech processing, adaptive control. He applies these techniques to many practical application situations, including data mining, internet search engine designs. From February 2004 to December 2005, he was the Executive Director, Mathematics, Information and Communications Sciences, Australian Research Council (ARC). In this role, he oversaw grant administration as well as formulation of policies pertaining to Mathematics, Information and Communication Sciences research in Australia. He also initiated a funding scheme, e-Research Support under the Special Research Initiatives to support e-Research activities in Australia. Since December 2005, he has been the Director, e-Research Centre, at Monash University.

He publishes widely in the area of adaptive signal processing, neural networks, fuzzy systems.

He received an Outstanding Alumni Award, Hong Kong Polytechnic University in 2001.