Abstract: A nonlinear approach based on Tikhonov regularised cost function is presented for blind signal separation of nonlinear mixtures. The proposed approach uses a multilayer Perceptron as the nonlinear demixer and combines both the information theoretic learning and the structural complexity learning into a single framework. In this paper, we show how this approach can be jointly used to extract independent components while constraining the overall Perceptron network to be as sparse as possible. The update algorithm for the nonlinear demixer is subsequently derived using the new cost function. We further explore how sparseness in the network connection can be utilised to determine the total number of layers required in the multilayer Perceptron and to prevent the nonlinear demixer from outputting arbitrary independent components. Experiments are meticulously conducted to study the performance of the new approach and the outcomes of these studies are critically assessed for performance comparison with existing methods.

Keywords: Nonlinear Independent Component Analysis, Tikhonov regularisation, multilayer Perceptrons and structural complexity learning.
1. INTRODUCTION

For almost a decade, blind source separation (BSS) using Independent Component Analysis (ICA) has received considerable amount of attention because of its simplicity and versatility in many signal processing applications [1-3]. The goal of ICA is to recover independent sources given only sensor observations that are unknown linear superposition of the unobserved independent source signals. Linear models are often used for both instantaneous and convolutive mixtures by virtue of their simplicity and ease of reconstruction. However, in general and for many practical problems, the mixed signals are more likely to be nonlinear or subject to some kind of nonlinear distortions due to sensory or environmental limitations which can be empirically modelled as

\[ \mathbf{x}(t) = f\left(\mathbf{s}(t)\right) = \mathbf{B}_2 \left( \mathbf{h}(\mathbf{B}_1 \mathbf{s}(t)) + \xi \right) \]  

where \( \mathbf{x}(t) = [x_1(t) \ x_2(t) \ \cdots \ x_N(t)]^\top \) (symbol ‘\( \top \)’ denotes transpose) and \( \mathbf{s}(t) = [s_1(t) \ s_2(t) \ \cdots \ s_N(t)]^\top \) are the spatially observed signals and source signals, respectively, \( \{\mathbf{B}_1, \mathbf{B}_2\} \) are the \( N \times N \) mixing matrices, \( \mathbf{h}(\cdot) = [h_1(\cdot) \ h_2(\cdot) \ \cdots \ h_N(\cdot)]^\top \) is a set of the nonlinear functions characterising the amount of nonlinearity in the mixing model and \( \xi \) is the \( N \times 1 \) bias vector which can be used to model random perturbation or some direct current (dc) components independent of \( \mathbf{s}(t) \). Figure 1 shows the proposed empirical nonlinear mixing model which is also capable of treating the linear model as a special case.

In this case, the nonlinearity elements are all set to \( h_j(u_j) = u_j \) and the bias weights \( \xi \) to zero which subsequently lead to \( \mathbf{x}(t) = f(\mathbf{s}(t)) = \mathbf{B} \mathbf{s}(t) \) with \( \mathbf{B} = \mathbf{B}_2 \mathbf{B}_1 \).

As linear BSS algorithms are not applicable in the above model, the search for nonlinear solutions to the problem becomes paramount. Therefore the need to study signal separation for nonlinear mixtures is significant at both theoretical and practical levels. Extension of existing theories and methods to
nonlinear BSS is not straightforward and so far, only a handful of initial effort has been attempted [4-19]. For a comprehensive survey on current and previous works in nonlinear BSS, readers are referred to [4 and references therein]. The first paper perhaps in a rigorous manner that deeply explores the nonlinear BSS problem is due to [5]. The paper considers the special case of two sources being mixed by a conformal mapping and the solution consists of two steps i.e. estimating a zero-preserving bijection map by using the Schwarz-Christoffel transformation and followed by estimating the rotational matrix by using the conventional linear ICA cost function. Recently, Kernel ICA [6-7] that uses a variant of nonlinear correlation analysis based on support vector machine is proposed for separating nonlinearly mixed signals. Tan et al. [8] proposed a radial basis function (RBF) network in which the hidden layer constitutes a set of Gaussian basis. The performance is directly dependent on the number of basis functions and the number of sources to be extracted. The method yields good performance when small number of sources is used accompanied by large number basis functions, normally two to fivefold the number of sources. Another type of nonlinear demixer is the self-organising map (SOM) [9]. The methodology is simple but it suffers from both network complexity and interpolation errors for continuous phase signals. Nested form of neural network models developed in [10-19] are more structured and reported to show better results than any of the previously proposed methods. Although its performance is promising, the approach suffers from a serious drawback in that it may indirectly extract independent components that are not the actual source signals but are related via some unknown maps determined by the combined mixing-demixing system. The situation is accentuated especially in cases where the complexity of neural network demixer does not match the underlying structure of the mixture.

2. MOTIVATIONS

The aim of this paper is to explore alternative approach to further enhance the performance of nested neural network-based solution by strategically controlling the structural complexity of the demixer model. This is achieved by the following objectives. Firstly, to extract independent components from
the observed signals. Secondly, to design a high-order nonlinear demixer to equalise the effects of nonlinearity embedded in the mixtures. Thirdly, to ameliorate the effects of the nonlinear demixer from outputting arbitrary independent components. The first objective can be met by using a multilayer Perceptron model whose last layer hidden neurons’ activation functions are selected or adjusted to match the derivatives of the conditional probability distribution function of the desired inputs to the Perceptron. In this case, the output from the hidden neurons will be uniformly distributed and hence results in statistical independence among the outputs. This principle closely resembles the technique of maximum entropy commonly adopted in linear ICA. On the other hand, the second and third objectives are significantly harder to meet if not impossible unless some kind of additional constraints (or/and information) are incorporated into the problem statement. The main reason stems from the fact that achieving output independence is not strong enough to grant signal separation in a nonlinear mixture and as a consequence, the outputs of the inverse system will be related to the input signals via an indeterminate nonlinear mapping which is highly undesirable. One approach to characterising the indeterminacies for a specific model \( \Xi \) is to examine the independence preservation equation [4]. Denoting \( T \) as the set of transforms that preserve independence and \( B \) as any map that transforms \( s \) to \( y \), the independence preservation equation states that for all \( A \) within \( C_N \) which is a \( \sigma \)-algebra on \( \mathbb{R}^N \), there exists

\[
\forall A \in C_N : \int_A d\tau_1(s_1) d\tau_2(s_2) \cdots d\tau_N(s_N) = \int_{B(A)} d\tau_1(y_1) d\tau_2(y_2) \cdots d\tau_N(y_N) \tag{2}
\]

and the following set

\[
\Psi = \left\{ \left( \tau_1(s_1), \tau_2(s_2), \ldots, \tau_N(s_N) \right) \mid \exists B \in \Xi \setminus (T \cap \Xi) : y = B(s) \text{ has independent components} \right\} \tag{3}
\]
of all input signal distributions \((r_1(s_1), r_2(s_2), \ldots, r_N(s_N))\) for which there exists non-trivial mapping \(B\) belonging to the model \(\Xi\) that preserves the independence of the components of the vector \(y\). Ideally, \(\Psi\) should be empty and hence \(\mathcal{T} \cap \Xi\) contains only the identity (or the permutation of the identity) as the unique element. However, in a general nonlinear mixing-demixing system where \(B\) has no particular form, this is not fulfilled and the input signals can be restored up to a non-trivial invertible nonlinear mapping belonging to the set \(\Xi \setminus (\mathcal{T} \cap \Xi)\) as denoted by (3). Basically, both equations in (2) and (3) aim to point out that in the independence preservation rule, there exists an infinite number of mappings that result in independent output signals but these signals are still mixed with respect to the source signals. This is not desirable and in this paper, this issue is being dealt with by using a nonlinear approach based on Tikhonov regularised cost function. Specifically, the proposed approach uses a multilayer Perceptron as the nonlinear demixer and combines both information theoretic and structural complexity learning into a single framework. This allows the information at the input to the Perceptron to be jointly maximised at the final output layer so that the output will be as independent as possible while limiting \(\Xi \setminus (\mathcal{T} \cap \Xi)\) to a subspace as small as possible, which is accomplished by inducing the overall Perceptron network to be as sparse as possible. This is equivalent to using a high-order nonlinear demixer for equalising the nonlinearity in the mixing system while limiting the demixer’s capability from over-equalisation, which would otherwise lead to a state where the combined mixing-demixing system becomes nonlinear and subsequently impact the demixer to output arbitrary independent components. Another major advantage of the proposed approach compared with other existing methods lies in the integration of a learning strategy whereby the complexity of the demixer can be decreased as the demixer evolves across time. This subsequently reduces the overall computational intensity once the demixer reaches the steady-state solution.
3. **EXACT INVERSE MODEL**

Prior to equalising the nonlinearity in the mixing model, it is crucial to investigate the conditions required for the existence of the inverse model to hold. In particular, when the conditions are satisfied it is vital to be able to compute the inverse model once the parameters of the mixing model are known. The following theorem characterises the inverse model:

*Theorem 1:* If the empirical nonlinear mixing function $\mathbf{x}(t) = f(\mathbf{s}(t))$ in (1) where $f : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is differentiable, then the inverse model exists and is given by $\hat{\mathbf{s}}(t) = f^{-1}(\mathbf{x}(t)) = \mathbf{B}_1^{-1} \left( \mathbf{h}^{-1} \left( \mathbf{B}_2^{-1} \mathbf{x}(t) - \xi \right) \right)$ provided that $\{\mathbf{B}_i\}_{i=1}^2$ are full rank and $\hat{h}_j(u_j) \neq 0$ for all $u_j \in \mathbb{R}$. Furthermore, the Jacobian matrix of the inverse model is given by $\mathbf{B}_1^{-1} \text{diag} \left[ \mathbf{h}^{-1} \right] \mathbf{B}_2^{-1}$ and its determinant is finite as given by

$$\prod_{i=1}^2 \prod_{j=1}^N \det \left[ \mathbf{B}_i^{-1} \right] \hat{h}_j^{-1}.$$

*Proof:* See Appendix.

It is worth noting that if the nonlinear function in (1) is continuously differentiable, this demands that $\hat{h}_j(u_j)$ be continuous at every point in the input space. Thus, a sufficient condition in choosing the nonlinearity is to constrain the derivatives $\frac{\partial^n \hat{h}_j(u_j)}{\partial u_j^n}$ to exist at least up to order $n = 2$ and be defined at every point in the open set $P$ of the input space. In addition, the empirical nonlinear model in (1) is strongly supported by the universal approximation theorem [20] which states that any continuous function can be approximated arbitrarily accurate by the empirical nonlinear model with $\{h_j(\cdot)\}_{j=1}^N$ selected from a set of non-constant, bounded and monotonically increasing functions, which satisfies the condition of $\frac{\partial^2 h_j(u_j)}{\partial u_j^2}$ and is defined at every point in the open set. Therefore, one can assume safely that with $\{h_j(\cdot)\}_{j=1}^N$ being non-constant, bounded and monotonically increasing
functions, the empirical model is general enough to approximate any continuous functions with arbitrary accuracy and the existence of the inverse model is always guaranteed (in conjunction with \( \{B_i\}_{i=1}^{2} \) having full rank).

4. TIKHONOV REGULARISED DEMIXING MODEL

Since the nonlinearity in the empirical nonlinear model is given by non-constant, bounded and monotonically increasing function, it follows that the inverse model

\[
f^{-1}(x(t)) = B^{-1}_1 \left( h^{-1} \left( B^{-1}_2 x(t) - \xi \right) \right)
\]

is also characterised by a non-constant, monotonically increasing but unbounded nonlinearity. Therefore, to effectively estimate the original signals \( s(t) \) from the nonlinearly distorted signals \( x(t) \), it is imperative that the inverse model can be accurately modelled by the neural network demixer. Analysing the inverse function, although it is an unbounded function, it is still being validated as a continuous function within the support specified by the open set \( Q \) in Theorem 1. Unfortunately, the ground in which a single-layer Perceptron is used to approximate any continuous functions is undeniably questionable within the context of nonlinear ICA. The universal approximation theorem assumes that there exists an unlimited size of neurons in the single-layer Perceptron i.e. \( N_2 \rightarrow \infty \) in

\[
y_j(t) = \sum_{j=1}^{N_j} w^{(2)}_{ij} g \left( \sum_{k=1}^{N_j} w^{(1)}_{jk} x_k(t) + \theta_j \right)
\]

so as to satisfy

\[
\left| f_i^{-1}(x(t)) - y_i(t) \right| \leq \varepsilon \quad \text{for a given threshold } \varepsilon .
\]

Clearly, it is inappropriate in practice to model any functions with an infinite number of nonlinear neurons in a single layer. To overcome this problem, we propose to use multiple hidden layers of nonlinearity where the number of hidden neurons within each layer matches the number of nodes in the input and output layers. In addition, for the case where each nonlinearity in the empirical nonlinear model is given by different functions i.e. \( h_1(\cdot) \neq h_2(\cdot) \neq \cdots \neq h_N(\cdot) \) in (1), it is then necessary to use multiple hidden layers at the demixer so as to enable the equalisation of a nonlinear channel \( h_i(\cdot) \) with high accuracy without incurring additional
deterioration at some other adjacent channels $\{h_j(\cdot)\}_{j=1,j\neq i}^N$. More importantly, within the context of inverse problems [21] (as in nonlinear ICA) the use of multiple hidden layers is further substantiated by the fact that even if the use of neuronal models with discontinuous nonlinear functions is permitted, single hidden layer Perceptron is insufficient to guarantee the solution of the inverse problems but requires a Perceptron with at least two hidden layers to induce a good-posed solution. Figure 2 shows the proposed demixing model based on a $L$-layer Perceptron (i.e. with $L-1$ layers of hidden nodes).

The input-output equation of the proposed multilayer Perceptron (MLP) can be described as follows:

$$y_i^{(l)} = \sum_{j=1}^{N} w_{ij}^{(l)} g_j^{(l-1)} (m_j^{(l-1)} y_j^{(l-1)} + \theta_j^{(l-1)}), \quad \forall i = 1, \ldots, N; \forall l = 1, \ldots, L.$$  (4)

where $L$ denotes the total number of layers in the MLP. In vector form, (4) can be represented as

$$y^{(l)} = W_l g^{(l-1)} \left( \text{diag} [m_{l-1}] y^{(l-1)} + \theta_{l-1} \right), \quad \forall l = 1, 2, \ldots, L.$$  (5)

where $y^{(l)} = [y_1^{(l)} \cdots y_N^{(l)}] \in \mathbb{R}^N$ and $W_l = [w_{ij}^{(l)}]_{j=1}^{N} \in \mathbb{R}^{N \times N}$ are the outputs and weights of the $l^{th}$ layer, respectively while $g^{(l-1)} = [g_1^{(l-1)} (m_1^{(l-1)} y_1^{(l-1)} + \theta_1^{(l-1)}) \cdots g_N^{(l-1)} (m_N^{(l-1)} y_N^{(l-1)} + \theta_N^{(l-1)})]^\top \in \mathbb{R}^N$, $\theta_{l-1} = [\theta_1^{(l-1)} \cdots \theta_N^{(l-1)}]^\top \in \mathbb{R}^N$ and $\text{diag}[m_{l-1}] = \text{diag}[m_1^{(l-1)} \cdots m_N^{(l-1)}] \in \mathbb{R}^{N \times N}$ are the nonlinearity, bias and gradient matrix associated with the $(l-1)^{th}$ layer with $y^{(0)} = x$, $\theta_0 = 0$, $\text{diag}[m_0] = \text{diag}[m_x] = I$ and $g^{(0)} = x$. Note that $g^{(l-1)}$ is a function of $m_{l-1}$, $y^{(l-1)}$ and $\theta_{l-1}$ although this is not explicitly shown for the sake of notation simplicity.
In training the $L$-layer Perceptron demixer, we need a cost function that determines not only the degree of separation in the de-mixed signals but also the strength of the synaptic connections in the demixer so as to maintain the structural complexity to be as sparse as possible. Let the set of parameters in the nonlinear demixer be denoted as $\Theta = \{W_{i=1}^L, \{m_{i=1}^{L-1}, \{\theta_{i=1}^{L-1}\}\}$, the training can be accomplished by minimising the Tikhonov regularised cost function defined as:

$$J(\Theta) = \xi_{\text{MI}}(\Theta) + \lambda \xi_{\text{SC}}(\Theta)$$

(6)

where

$$\xi_{\text{MI}}(\Theta) = -h(y^{(L)}) + \sum_i \tilde{h}(y_i^{(L)}) + \sum_i \left(\sigma_{y_i^{(L)}}^2 - \sigma_h^2\right)^2$$

(7)

$$\xi_{\text{SC}}(\Theta) = \frac{1}{2} \int \partial x \gamma(x) \left\| \partial^k G(x, \Theta) / \partial x^k \right\|^2$$

(8)

The first term $\xi_{\text{MI}}(\Theta)$ is essentially the mutual information defined at the output of the $L^{th}$ layer of the demixing model while $h(y^{(L)})$ and $\tilde{h}(y_i^{(L)})$ are the differential joint and marginal entropy, respectively. The mutual information in (7) is augmented with an additional term $\sum_i \left(\sigma_{y_i^{(L)}}^2 - \sigma_h^2\right)^2$, which is crucial in the optimisation process to limit the variance of outputs of the demixer $\sigma_{y_i^{(L)}}^2$ to be identical to the variance of the source signals $\sigma_h^2$. The series term $\tilde{q}_i(y_i^{(L)})$ represents the marginal probability density function (pdf) of the output layer and $h(x)$ is the entropy of the input signals which can simply be treated as a constant during the optimisation process. $G(x; \Theta)$ is the input-output mapping realised by the multilayer Perceptron demixer, and $\gamma(x)$ is some weighting function that ensures the integral converges and determines the region of the input space over which the mapping $G(x; \Theta)$ is required to be smooth by making the $k^{th}$ order derivative of $G(x; \Theta)$ with respect to $x$
small. The larger the value of $k$, the smoother the mapping $G(x; \Theta)$ will become. The term $\lambda$ represents the regularisation parameter that controls the amount of weighting between extracting independent components and equalising the nonlinearity in the mixtures.

Previous work [8,17] proposed to use moments or cumulants matching between the source signals and outputs of the demixer to further restrict the non-trivial mapping $B$ in the combined space as dictated by $\mathcal{R}(T \cap \mathcal{X})$. However, this method can be non-realistic since these statistics are not known \textit{a priori} in a completely blind system. Moreover, they are difficult to obtain in practice as large amount of data is required especially for estimating the higher order statistics. This situation is made worse if the source signals are non-stationary and the estimates of these statistics subsequently become unreliable under low signal-to-noise ratio (SNR). More crucially, the use of moments or cumulants matching has direct bearing on the convergence speed of the update algorithm which inadvertently depends on the order of the statistics.

To develop the update algorithm for the $L$-layer Perceptron, we may consider the first order variation of $J$ due to small perturbation on $\Theta$ i.e. $\partial \Theta$ shown as

$$
\begin{align*}
\partial J(\Theta) &= J(\Theta + \partial \Theta) - J(\Theta) \\
&= \partial \xi_{\text{MI}} + \lambda \partial \xi_{\text{SC}} \\
&= \partial \xi_{\text{MI}} + \lambda \partial \xi_{\text{SC}}
\end{align*}
$$

The differential of $\xi_{\text{MI}}$ for a $L$-layer Perceptron demixer may take the following form:
\[ \partial \xi_{MI} = -\partial \left( \log \left| \det \frac{dy^{(L)}}{dx} \right| \right) - \sum_i \partial \left( \log \tilde{q}_i(y_i^{(L)}) \right) + \sum_i \partial \left( \sigma_{yi}^2 - \sigma_i^2 \right)^2 \]
\[ = -\partial \left( \log \left| \det \left( \prod_{k=1}^{L} W_k \prod_{k=1}^{L-1} \text{diag}[m_k] \prod_{k=1}^{L-1} \text{diag}[g^{(k)}] \right) \right| \right) \]
\[ - \sum_i \partial \left( \log \tilde{q}_i(y_i^{(L)}) \right) + \sum_i \partial \left( \sigma_{yi}^2 - \sigma_i^2 \right)^2 \]
\[ = -\sum_{k=1}^{L} \text{tr} \left[ \partial W_k W_k^{-1} \right] - \sum_{k=1}^{L} \text{tr} \left[ \partial \text{diag}[m_k] \text{diag}[m_k] \right] \]
\[ + \partial \left( -\sum_{k=1}^{L} \sum_i \log \tilde{g}_i^{(k)}(m_i^{(k)} y_i^{(k)} + \theta_i^{(k)}) \right) + \partial \left( \sum_i \left( -\log \tilde{q}_i(y_i^{(L)}) + \left( \sigma_{yi}^2 - \sigma_i^2 \right)^2 \right) \right) \]

(10)

where 'tr' denotes the trace operation. Let us define the following functions:

\[ \phi^{(j)} = \left[ \phi_1^{(j)}(m_1^{(j)} y_1^{(j)} + \theta_1^{(j)}) \quad \phi_2^{(j)}(m_2^{(j)} y_2^{(j)} + \theta_2^{(j)}) \quad \cdots \quad \phi_N^{(j)}(m_N^{(j)} y_N^{(j)} + \theta_N^{(j)}) \right]^T \]

(11)

\[ \phi^{(L)} = \left[ \phi_1^{(L)}(y_1^{(L)}) \quad \phi_2^{(L)}(y_2^{(L)}) \quad \cdots \quad \phi_N^{(L)}(y_N^{(L)}) \right]^T \]

(12)

\[ v^{(L)} = \left[ v_1^{(L)}(y_1^{(L)}) \quad v_2^{(L)}(y_2^{(L)}) \quad \cdots \quad v_N^{(L)}(y_N^{(L)}) \right]^T \]

(13)

where \( \phi_i^{(j)}(y_i^{(L)}) \triangleq \partial \log \tilde{q}_i(y_i^{(L)}) / \partial y_i^{(L)} \), \( \phi_i^{(j)}(m_i^{(j)} y_i^{(j)} + \theta_i^{(j)}) \triangleq \partial \tilde{g}_i^{(j)}(m_i^{(j)} y_i^{(j)} + \theta_i^{(j)}) / \partial y_i^{(L)} \) with \( \tilde{g}_i^{(j)}(m_i^{(j)} y_i^{(j)} + \theta_i^{(j)}) \) being the first and second order derivatives of the nonlinearity with respect to the parameters and \( v_i^{(L)}(y_i^{(L)}) \triangleq \partial \left( \sigma_{yi}^2 - \sigma_i^2 \right) / \partial y_i^{(L)} \). Hence,

\[ \partial \xi_{MI} = -\sum_{k=1}^{L} \text{tr} \left[ \partial W_k W_k^{-1} \right] - \sum_{k=1}^{L-1} \text{tr} \left[ \partial \text{diag}[m_k] \text{diag}[m_k] \right] \]
\[ + \left[ \phi^{(L)} + v^{(L)} \right]^T \partial y^{(L)} + \sum_{k=1}^{L-1} \phi^{(k)} \left[ \partial \text{diag}[m_k] y^{(k)} + \text{diag}[m_k] \partial y^{(k)} + \partial \theta_k \right] \]

(14)
and the differentials of $y^{(l)}$ are recursively related to the previous outputs as

$$\partial y^{(l)} = \partial W_l g^{(l-1)} + W_l \text{diag}[g^{(l-1)}] \left( \partial \text{diag}[m_{l-1}] y^{(l-1)} + \text{diag}[m_{l-1}] \partial y^{(l-1)} + \partial \theta_{l-1} \right)$$  \hspace{1cm} (15)$$

for $l = 1, 2, \ldots, L$. Substituting (15) into (14), this culminates to

$$\partial \xi^k_{\text{ML}} = - \sum_{l=1}^L \text{tr} \left[ \partial W_k W_{k+1}^{-1} \right] - \sum_{l=1}^{L-1} \text{tr} \left[ \partial \text{diag}[m_k] \text{diag}[m_k]^{-1} \right]$$

$$+ \left[ \varphi^{(L)} + v^{(L)} \right]^T \left[ \partial W_L g^{(L-1)} + W_L \text{diag}[g^{(L-1)}] \left( \partial \text{diag}[m_{L-1}] y^{(L-1)} + \partial \theta_{L-1} \right) \right]$$

$$+ \sum_{l=1}^{L-1} \varphi^{(k)} \left[ \partial \text{diag}[m_k] y^{(k)} + \partial \theta_k \right]$$

$$+ \left[ \varphi^{(L)} + v^{(L)} \right]^T \left[ W_L \text{diag}[g^{(L-1)}] \text{diag}[m_{L-1}] \partial y^{(L-1)} \right] + \sum_{l=k}^{L-1} \varphi^{(k)} \left[ \text{diag}[m_k] \partial y^{(k)} \right]$$  \hspace{1cm} (16)$$

By considering the derivatives of $\xi^k$ with respect to the parameters, the following are obtained:

$$\frac{\partial \xi^k_{\text{ML}}}{\partial W_k} = \begin{cases} -W_L^{-1} + e^{(L)} g^{(L-1)\dagger} , & k = L \\ -W_k^{-1} + \text{diag}[m_k] e^{(k)} g^{(k-1)\dagger} , & k = 1, 2, \ldots, L-1 \end{cases}$$  \hspace{1cm} (17)$$

$$\frac{\partial \xi^k_{\text{ML}}}{\partial \text{diag}[m_k]} = -\text{diag}[m_k] \dagger + \text{diag}[e^{(k)} \circ y^{(k)}]$$  \hspace{1cm} (18)$$

$$\frac{\partial \xi^k_{\text{ML}}}{\partial \theta_k} = e^{(k)}$$  \hspace{1cm} (19)$$

where $e^{(k)}$ which is analogous to the error function in the backpropagation algorithm [22] can be recursively computed as

$$e^{(k)} = \begin{cases} \varphi^{(L)} + v^{(L)} , & k = L \\ \varphi^{(j)} + \text{diag}[g^{(k)}] W_{k+1}^\dagger \text{diag}[m_{k+1}] e^{(k+1)} , & k = 1, 2, \ldots, L-1 \end{cases}$$  \hspace{1cm} (20)$$
For a global weighting smoother, one can select a Gaussian form given by

\[ Y(x) = \frac{1}{(2\pi)^{N/2} (\det R)^{1/2}} \exp \left[ -\frac{1}{2} x^T R^{-1} x \right] \]  \hspace{1cm} (21)

For simplicity, we can use \( R = \sigma^2 I \) and it is required that \( \sigma \) be large for global smoothing effect. The Gaussian function is chosen as it simplifies the evaluation of the smoothing integral considerably since it is both separable and spherically symmetric. On the other hand, for a local weighting smoother, one can select a more general form:

\[ Y(x) = \frac{1}{Q} \sum_{k=1}^{Q} \frac{1}{(2\pi)^{N/2} (\det R)^{1/2}} \exp \left[ -\frac{1}{2} (x - x_k)^T R^{-1} (x - x_k) \right] \]  \hspace{1cm} (22)

so as to capture the local variation of the input space where \( \{x_k\}_{k=1}^{Q} \) are a set of input data points and that using \( R = \sigma^2 I \), it is required that \( \sigma \) be selected small such that

\[ \lim_{\sigma \to \infty} \frac{1}{(2\pi \sigma^2)^{N/2}} \exp \left[ -\frac{1}{2\sigma^2} \|x - x_k\|^2 \right] = \delta(x - x_k) \]  \hspace{1cm} (23)

where \( \delta(\cdot) \) is the delta function. Using Eqns. (21)-(22) and by applying the analysis in [23] for a L-layer Perceptron, it can be shown that the structural complexity function in (8) can be approximated as

\[ \xi_{SC} = \sum_i \sum_j \sum_{k=1}^{L-1} \left( w_{ij}^{(L)} \right)^2 \|w_{ij}^{(k)}\|^p \]  \hspace{1cm} (24)

with
where $w_j^{(k)} = \left[w_{j1}^{(k)} \ w_{j2}^{(k)} \ \cdots \ w_{jN}^{(k)} \right]$ is the $j^{th}$ row of matrix $W_k$, $\| \cdot \|^p$ is the $p$-norm and $q$ is the order of differentiation of $F(x, \Theta)$ with respect to $x$. The simple algebraic form of $\xi_{SC}$ in (24) enables the direct enforcement of smoothness without the need for costly Monte-Carlo integrations as required in (8). The proposed approach in (24) is more accurate than the conventional weight decay or weight elimination methods for the complexity regularisation since the former succinctly distinguishes between the roles of synaptic weights in the hidden layers and those in the output layer. This distinction is crucial in order for the demixer to maintain a close match to the underlying complexity of the input data and to avoid arbitrary generalisation that leads to non-trivial invertible nonlinear mapping due to over-specified degree of freedom in the structural complexity.

The derivatives of the structural complexity function $\xi_{SC}$ with respect to the parameters $w_j^{(k)}$ have been derived as follows:

$$\frac{\partial \xi_{SC}}{\partial w_j^{(k)}} = 2 w_j^{(k)} \sum_{k=1}^{L-1} \| w_j^{(k)} \|^p$$

(26)

and

$$\frac{\partial \xi_{SC}}{\partial w_j^{(k)}} = p \left( w_j^{(k)} \right)^{p-1} \sum_{i} \left( w_i^{(L)} \right)^2$$

(27)

where $\left( w_j^{(k)} \right)^{p-1} \equiv w_j^{(k)} \circ w_j^{(k)} \circ \cdots \circ w_j^{(k)}$ and `$\circ$' is the Hadamard product. In scalar representation, the derivative of $\xi_{c}$ with respect to matrices $W_k$ takes the following form:
Hence, the derivatives of the Tikhonov regularised cost function with respect to matrices $W_k$ may assume the form of

$$
\frac{\partial \xi}{\partial W_{mn}^{(k)}} = \begin{cases} 
2w_{mn}^{(L)} \sum_{i=1}^{L-1} \left| w_n^{(i)} \right|^p, & k = L, \\
p \left( w_{mn}^{(k)} \right)^{p-1} \sum_i \left( w_{mn}^{(L)} \right)^2, & k = 1, 2, \ldots, L - 1.
\end{cases}
$$

(28)

In (29), $[z]_{m,n}$ denotes a matrix where its elements are given by $z_{m,n}$. As the complexity-penalty function precludes any functions other than the weights matrices $W_k$, the derivatives of the combined cost function with respect to the gradient and threshold simply remain unaltered.

By considering a first order perturbation on the Tikhonov regularised cost function, one can define the relative gradient descent [1-3] with respect to the matrix parameters $\Omega = \{W_k\}_{k=1}^L, \{\text{diag}[m_k]\}_{k=1}^{L-1}$ as

$$
\Delta \Omega = -\mu \frac{\partial \xi}{\partial \Omega} \Omega^\dagger \Omega
$$

such that

$$
\Delta W_k = \begin{cases} 
\mu \left[ \left( I - e^{(L)} y^{(L)\dagger} - \lambda \left( 2w_{mn}^{(L)} \sum_{i=1}^{L-1} \left| w_n^{(i)} \right|^p \right) W_L^\dagger \right) W_L \right]^\dagger, & k = L, \\
\mu \left[ \left( I - \text{diag}[m_k] e^{(k)} y^{(k)\dagger} - \lambda \left( p w_{mn}^{(k)} \sum_i \left( w_{mn}^{(L)} \right)^2 \right) W_k^\dagger \right) W_k \right]^\dagger, & k = 1, \ldots, L - 1.
\end{cases}
$$

(30)
\[ \Delta \text{diag}[\mathbf{m}_k] = \mu \text{diag}[\mathbf{I} - \text{diag}[\mathbf{e}^{(k)} \circ \mathbf{y}^{(k)} \circ \mathbf{m}_k]] \text{diag}[\mathbf{m}_k] \]  

(31)

Hence, the updates for overall demixer parameters \( \Theta = \{\{\mathbf{W}_k\}_{k=1}^L, \{\mathbf{m}_k\}_{k=1}^{L-1}, \{\theta_k\}_{k=1}^{L-1}\} \) can be computed as follows:

\[
\mathbf{W}_k(t+1) = \mathbf{W}_k(t) + \mu_k^w \left\{ \begin{array}{l}
\mathbf{W}_L(t) - \mathbf{e}^{(L)}(t) \mathbf{y}^{(L)*}(t) \mathbf{W}_L(t) \\
-\mathbf{A} \left[ 2 \mathbf{w}_{mn}^{(L)}(t) \sum_{j=1}^{L-1} \|\mathbf{w}_n^{(j)}(t)\|^2 \right] \mathbf{W}_L^*(t) \mathbf{W}_L(t) 
\end{array} \right\}, \quad k = L
\]

\[
\mathbf{W}_k(t+1) = \mathbf{W}_k(t) + \mu_k^w \left\{ \begin{array}{l}
\mathbf{W}_k(t) - \text{diag}[\mathbf{m}_k(t)] \mathbf{e}^{(k)}(t) \mathbf{y}^{(k)*}(t) \mathbf{W}_k(t) \\
-\mathbf{A} \left[ p \left( \mathbf{w}_{mn}^{(k)}(t) \right)^{p-1} \sum_{j=1}^{L-1} \left( \mathbf{w}_m^{(j)}(t) \right)^2 \right] \mathbf{W}_k^*(t) \mathbf{W}_k(t) 
\end{array} \right\}, \quad k = 1, \ldots, L - 1.
\]

(32)

\[
\text{diag}[\mathbf{m}_k(t+1)] = \text{diag}[\mathbf{m}_k(t)] + \mu_k^m \text{diag}[\mathbf{I} - \text{diag}[\mathbf{e}^{(k)}(t) \circ \mathbf{y}^{(k)}(t)] \text{diag}[\mathbf{m}_k(t)]] \text{diag}[\mathbf{m}_k(t)] 
\]

\[
= \text{diag}[\mathbf{m}_k(t)] + \mu_k^m \text{diag}[\mathbf{m}_k(t) - \mathbf{e}^{(k)}(t) \circ \mathbf{y}^{(k)}(t) \circ (\mathbf{m}_k(t))^2]
\]

(33)

\[
\theta_k(t+1) = \theta_k(t) - \mu_k^\theta \mathbf{e}_k(t)
\]

(34)

where \( \{\mu_k^L\}_{k=1}^L \), \( \{\mu_k^m\}_{k=1}^{L-1} \) and \( \{\mu_k^\theta\}_{k=1}^{L-1} \) are the step sizes. Hence, Eqns. (32)-(34) represent the update algorithm for optimising the parameters of the \( L \)-layer Perceptron based on the Tikhonov regularised cost function epitomised in (6).

The selection of total number of layers in the nonlinear demixer is important in order to limit the overall computational complexity of the Perceptron and to prevent the demixer from outputting arbitrary independent components. Therefore, a simple rule that is both measurable and conveys
information about the structural complexity of the demixer is exactly what is needed here. The motivation stems from the fact that as the number of hidden layers increases, the nonlinear demixer is increasingly more powerful and subsequently the subspace of \( \Xi \setminus (\mathbf{T} \cap \Xi) \) grows larger which means that more non-trivial elements are nonlinearly related to the source signals and hence causing the demixer to output signals that are nonlinear transformation of the original source signals. Therefore, the need to minimise the structural complexity of the demixer while maintaining its ability to equalise the nonlinear effects in the mixture becomes inevitable. A feasible solution is to initially use a large size demixer with many hidden layers and subsequently followed by layer trimming if the number of weak synaptic connections exceeds the number of trivial weights\(^1\) in a single layer. This procedure is repeated until the number of weak synaptic connections is less than the number of trivial weights. We may formulate this scheme as follows: Let

\[
D(L) = \left\{ \{w_{ij}^{(l)}\}_{l=1}^{L} : w_{ij}^{(l)} \in [-B, B] \right\} \tag{35}
\]

denotes the set of weak synaptic connections where \( B \) is the threshold level that determines whether a connection is weak or otherwise i.e. any weights with \( |w_{ij}^{(l)}| \leq B \) will be considered as weak connections. In order for this scheme to work efficiently, the gradient of the hidden neuron nonlinear function needs to be set to unity so as to ascertain the weights in the demixer are on a common scale. Furthermore, let \( \mathcal{R}(D) \) denotes the cardinal of the set \( D \), \( N^2 \) the number of weights in a single layer and \( N_0 \) the total number of layers of the demixer. The optimum value for \( B \) is currently under investigation but for practical purpose, we propose to estimate \( B \) as the unbiased sample statistics of the scaled standard deviation of the weights distribution given by

\(^1\) For example, in the identity matrix the trivial weights are the off-diagonal elements and therefore, an \( 3 \times 3 \) identity matrix has 6 trivial elements. In general, any \( N \times N \) matrix has \( N^2 - N \) number of trivial weights.
\[ \hat{B} = \kappa \times \hat{\sigma} \]

\[ \hat{\sigma} = \sqrt{\frac{1}{N^2 N_0 - 1} \sum_{i=1}^{N} \sum_{j=1}^{N} (w_{ij}^{(t)} - \hat{\mu})^2} \]  

(36)

\[ \hat{\mu} = \frac{1}{N^2 N_0} \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{l=1}^{\lambda} w_{ij}^{(l)} \]

Monte-Carlo experiments have been conducted which suggest that for reasonably good performances, \( \kappa \) needs to be chosen within the interval of \([0,0.15]\). Assuming that the algorithm has converged to the steady state solution, the method of layer trimming works by reducing the total number of layers \( N_0 \) by one whenever the number of weak synaptic connections exceeds \( N^2 - N \), which represents the number of trivial weights in a single layer. The outline of the proposed ‘layer trimming’ scheme is illustrated in Chart 1. In the following, we present two experiments to verify the proposed work in which the first experiment uses 3 synthetic signals whereas the second uses 3 recorded speech signals.

5. RESULTS

In the first experiment, the following nonlinear mixture for the case of 3 input signals and 3 sensors is considered since such study can assist us in gaining insights into the efficacy of the proposed scheme.

The input signals are given by \([s_1(t) \ s_2(t) \ s_3(t)] = [0.4(1+\sin(60\pi t))\cos(100\pi t) \ 0.9\sin(20\pi t) \ u(t)]\)

where \( u(t) \) is a uniformly distributed random signal within the interval of \([-0.5,0.5]\) and each sensor is perturbed with independent white gaussian noise \( n_i \). The input signals are nonlinearly coupled according to the following model:
\[
\begin{bmatrix}
  v_1(t) \\
v_2(t) \\
v_3(t)
\end{bmatrix} = B_1 \times \begin{bmatrix}
  s_1(t) \\
s_2(t) \\
s_3(t)
\end{bmatrix} + \begin{bmatrix}
  \zeta_1 \\
  \zeta_2 \\
  \zeta_3
\end{bmatrix} : B_1 = \begin{bmatrix}
  1 & 1 & 1 \\
  -1 & 1 & -1 \\
  2 & 0 & 1
\end{bmatrix}
\]

\[
\begin{bmatrix}
  x_1(t) \\
x_2(t) \\
x_3(t)
\end{bmatrix} = B_2 \times \begin{bmatrix}
  \frac{1-\exp(-2v_1(t))}{1+\exp(-2v_1(t))}v_1(t) \\
  \ln(v_3(t) + \sqrt{v_3^2(t) + 1})
\end{bmatrix} + \begin{bmatrix}
  n_1(t) \\
n_2(t) \\
n_3(t)
\end{bmatrix}
\]

where elements of \( B_2 \) are randomly sampled from a uniform distribution within the interval of \([0,1]\) and \( \{\zeta_i\}_{i=1}^3 \) are set to zero. Note that all nonlinearities in the above mixing model are distinct. In particular, the first nonlinearity is a bounded function within \([-1,1]\) whereas the last two are unbounded. Figure 3(a) and (b) show the original source signals and the nonlinearly mixed signals, respectively. A multilayer Perceptron will be used as the demixing model whereas the linear demixer acts as the benchmark for comparison. The Perceptron demixer with different number of layers \( L \), order of smoothing \( p \) and amount of regularisation \( \lambda \) will be investigated in this experiment. The hidden neuron function of the demixer assumes the form of \( g_i^{(L)}(y) = \frac{1-\exp(-2y)}{1+\exp(-2y)} \). On the other hand, a linear demixing model can be directly obtained from the multilayer Perceptron by setting \( L=1 \) and \( g_i^{(1)}(y) = y \). Since the source signals are sub-gaussian, \( \Phi_i^{(L)}(y_i^{(L)}) \) can be approximated by using the truncated 4th order Edgeworth series [1]. To test the efficacy of the proposed scheme, we have used the following models:

- Linear ICA i.e. single Perceptron \( L=1 \).
- 2-layer Perceptron without structural complexity learning i.e. \( L=2, \lambda = 0 \).
- 3-layer Perceptron with global structural complexity learning i.e. \( L=3, p = 1 \).
- 3-layer Perceptron with local structural complexity learning i.e. \( L=3, p = 2 \).

Batch learning is used in case and the weights are initialised as identity matrices with step sizes \( \{\mu_k^w\}_{k=1}^L = \{\mu_k^g\}_{k=1}^{L-1} = 0.3 \). Figure 4(a)-(d) show the recovered signals attained from these Perceptrons.
which clearly demonstrate that the linear model fails to separate nonlinearly mixed signals. The 2-layer Perceptron manages to retrieve 1 source signal while the 3-layer Perceptron with global structural complexity learning is able to retrieve 3 source signals although with lesser accuracy for the third signal. However, the best performance is attained by the 3-layer Perceptron optimised using the local structural complexity learning at $\lambda = 1$ in which all the source signals are successfully retrieved with high accuracy. We hypothesise that since the source signals used in (37) are localised by virtue of its sub-gaussianity density function (as illustrated by its discontinuities in the pdf), smoothing in the outputs of the demixer must concord to the signals distribution which is achieved by using the local structural complexity. Conversely, if the source signals are non-localised by virtue of its super-gaussianity, then global structural complexity learning will result in better performance than that given by the local structural complexity learning.

For evaluation of the convergence and accuracy in signal recovery, the following performance index is used which measures the norm-2 deviation of the demixer outputs from the original source signals:

$$P = \left( 1 - \frac{1}{N} \sum_{i=1}^{N} |\rho_i| \right)$$ (38)

where

$$\rho_i = \frac{\text{E}[(s_i - E[s_i])(y_i - E[y_i])]}{\sqrt{\text{E}[|s_i - E[s_i]|^2] E[|y_i - E[y_i]|^2]}}$$ (39)

where $\rho_i$ is the normalised cross-correlation. In (38), the notations ‘*’ and ‘|.|’ denote the complex conjugate and absolute operation, respectively. The proposed performance index is essentially derived from the mean square error criterion that implicitly takes into account the scale and phase reversal ambiguities. It is desirable to have the performance index as small as possible as this indicates the degree of similarity between the solution and the actual source signals. Figure 5 shows the convergence of the performance index for each demixing model based on a Monte Carlo simulation of
100 independent trials. From the plot, it is evident that linear ICA scheme is highly unsuitable for separating nonlinear mixture with a relatively high error rate at 0.403 while using the 2-layer Perceptron without structural complexity learning, the error rate is reduced to 0.314. These results are then further improved by using the 3-layer Perceptron combined with global and local structural complexity learning. The plot also clearly identifies the superiority in performance where the best signal recovery is given by the 3-layer Perceptron with local structural complexity learning resulting in an error rate of 0.15, which is at least two and a half times better than the performance given by the linear model and twice that of the 2-layer Perceptron. To further substantiate the effects of structural complexity on signal separation, different weightings of regularisation have been applied to the Tikhonov cost function. Figures 6 and 7 show the performance indices of the 3-layer Perceptron optimised in conjunction with the structural complexity learning with varying $\lambda$ at $p = 1$ and $p = 2$, respectively. In both figures, we may identify that the selection of $\lambda$ is crucial to render good performances and from the plots, it is shown that demixer optimised with structural complexity learning is superior to that without the structural complexity learning. On the other hand, we note from the plots that placing over-emphasis on the structural complexity learning (i.e. increasing $\lambda$) may divert the demixing process from retrieving independent components from the mixtures. Also from the simulations, it is found that setting $\lambda = 1$ results in the best performance for both local and global structural complexity learning. Nonetheless, we have also investigated the case of $p = 1$ and $p = 2$ but the resulting performances are poorer than previous results by at least 15% and the computer experiments conducted so far seem to suggest that using $p = -1$ and $p = 0$ result in the update algorithm highly sensitive to the amount of noise perturbing the mixtures. Use of $p = 3$ and $p = 4$ have also been experimented but no conclusive results have been obtained as the algorithm becomes easily unstable especially when the weights’ values exceed unity.

To investigate the effects of sparseness of the weight connection due to the structural complexity learning, the histograms of the 3-layer Perceptron weights (averaged out of 100 independent trials) are
plotted in Figures 8(a)-(c). Figure 8(a) shows the histogram when the Perceptron is updated without using the structural complexity learning where a uniform weights distribution is clearly evident from the plot i.e. sparseness property is not present. On the other hand, the sparseness of the weights is perceptible in Figures 8(b)-(c) for the case of 3-layer Perceptron with structural complexity learning where at least 30% of the total weights concentrate around the region of $[-0.5,0.5]$. The standard deviations as measured by (38) have been computed to be $\sigma(\lambda = 0) = 1.233$, $\sigma(\lambda = 1, p = 1) = 1.054$ and $\sigma(\lambda = 1, p = 2) = 0.923$ for the case of without structural complexity learning, with global structural complexity learning and with local structural complexity learning, respectively. In addition, the number weights that exceed the range of $B = [-0.15\times\sigma(\lambda = 0), 0.15\times\sigma(\lambda = 0)]$ for global and local structural complexity learning are calculated to be 3 and 4 which both are less than $N^2 - N = 6$ — hence indicating that the 3-layer Perceptron is adequate for separating the above nonlinearly mixed signals. To verify this, we have simulated the performance of the 4-layer Perceptron and subsequently plotted the histograms of the converged weights in Figures 9(a)-(c). Contrasting Figure 9(a) with Figures 9(b)-(c), the histograms clearly identify that majority of the weights are almost redundant and proceeding in a similar fashion, it is calculated that up to 7 and 9 weights (corresponding to the global and local structural complexity learning, respectively) have their value falls within the critical range indicated by $\hat{B}$ — hence indicating that the 4-layer Perceptron is more than adequate for separating the signals. The overall results from Figures 4 to 9 demonstrate that an optimum solution at least locally can be obtained by using demixer as simple as the 3-layer Perceptron trained with mutual information cost function combined with weighted $\lambda = 1$ structural complexity learning at $p = 1$ and $p = 2$.

In the second set of experiments, the same nonlinear mixing model in (37) is used but the source signals are now given by the recorded speech signals as illustrated in Figure 10(a) while Figure 10(b) shows the observed nonlinearly mixed signals. Following the results obtained from experiment 1, the
3-layer Perceptron with \( \lambda = 1 \) structural complexity learning at \( p = 1 \) and \( p = 2 \) will be investigated alongside the following demixers:

- Kernel ICA using radial basis function [6].
- Polynomial Neural Network (PNN) using 9\(^{th}\) order polynomial [17].

The use of 9\(^{th}\) order polynomial has been previously shown to be adequate for equalising mild to strong nonlinearity embedded in the mixture while maintaining a reasonable architectural size to prevent it from ‘overfitting’ [17]. For comparison purpose, the above demixers are initialised in such a way that they are identical to the linear demixing model. The 3-layer Perceptron optimises its parameters by using the gradient algorithm derived in (32)-(34) and the step sizes are set to \( \{\mu_k^w\}_{k=1,2,3} = \{\mu_k^0\}_{k=1,2} = 0.2 \). As speech signals are used, the marginal pdf can be approximated by 
\[
\tilde{q}_i(y_i) = p_G(y_i) \text{sech}^2(y_i) \quad \text{where} \quad p_G(y_i) = N(0,1)
\]
the zero mean unit variance normal distribution [3].

The recovered signals from each demixer are plotted in Figures 11(a)-(d) where it is seen that the performances of both Kernel ICA and PNN demixers are inferior to the 3-layer Perceptron with structural complexity learning. In particular, the first two source signals are successfully retrieved with high accuracy by the Perceptron model as illustrated in Figure 11(c)-(d). The performance index of each demixer is further plotted in Figure 12 (averaged out of 100 realisations) where it is observed that the Kernel ICA gives rise to a relatively high error rate of 0.31 while PNN is comparatively better with an error rate of 0.18. The reason for such poor performance is that the Kernel ICA can successfully separate signals when the mixed signals undergo a linear-nonlinear transformation i.e. nonlinearity is embedded after the mixing as in the case of post-nonlinear mixture [13-14]. However, in this experiment, the mixed signals undergo a linear-nonlinear-linear transformation and therefore this poses a more difficult problem as the demixing models do not match the underlying structure of the mixture. The PNN structure matches the mixing model and succeeds in removing some degree of the nonlinearity in the mixture. However, careful inspection on Figure 11(b) reveals that some portions of the noise have actually been amplified at the PNN outputs due to the high order polynomial used by the PNN. The performance would worsen considerably if high level of noise is to perturb the sensors.
Also, it is seen that the convergence of the performance index is relatively slow within the first 35 iterations, which reflects the slow dynamical changes in the coefficients of the polynomial when they are updated. After this transition, a sharp drop in the performance index occurs where the weights of the PNN are adapted until it reaches the steady state at 0.18. On the other hand, substantial improvement is achieved by using the 3-layer Perceptron especially in the case where the demixer is trained with the global structural complexity learning with $p = 1$. This is in line with our hypothesis that the global structural complexity learning tends to result in better performance for super-gaussianity signals such as speech than the local structural complexity learning. Based on the above findings, we conclude that the 3-layer Perceptron trained with the global structural complexity learning gives the best speech separation for the nonlinear mixture in (37).

6. SUMMARY

A novel approach to blind signal separation of nonlinear mixtures based on Tikhonov regularised cost function is presented. An empirical model where a layer of nonlinearity is sandwiched between two matrices is used as the nonlinear mixing system. This paper derives a specified inverse model for equalising the nonlinearity in the mixtures. The proposed approach uses the multilayer Perceptron to approximate the inverse model and exploits both the information theoretic learning and structural complexity learning to retrieve independent components from the mixtures where the structural complexity of the demixer is constraint to be as sparse as possible. It is shown that the sparseness property can be intelligently utilised to determine the adequate number of layers required for nonlinear signal separation and to prevent the demixer from outputting arbitrary independent components. Experimental studies have been conducted on the performance of the new approach by using experiments derived from an empirical mixing model and speech signals. These studies have demonstrated that the proposed approach is highly effectiveness in terms of performance and in reducing the overall computational complexity as compared with other existing methods.
7. REFERENCES


8. APPENDIX

Proof of theorem 1.

Since the nonlinear mixing function \( x = f(s) \) in (1) where \( f : \mathbb{R}^N \rightarrow \mathbb{R}^N \) is differentiable with respect to its argument and assuming that \( \frac{dx}{ds} \neq 0 \) at an arbitrary point \( p \), then there exist open sets \( P, Q \subseteq \mathbb{R}^N \) where \( p \in P, \; q = f(p) \in Q \) such that \( f \) is a diffeomorphism of \( P \) onto \( Q \). In addition, when \( f \) is smooth, the inverse mapping \( f^{-1} : \mathbb{R}^N \rightarrow \mathbb{R}^N \) is also a smooth diffeomorphism such that \( f \) and \( f^{-1} \) comprise a unique one-to-one local bijective on \( P \) and \( Q \). Now, on differentiating (1) this leads to \( \frac{dx}{ds} = B_2 \times \text{diag}[\hat{h}] \times B_1 \) where \( \hat{h} \) is a vector that contains the element-by-element derivative of the nonlinearity \( h_j(\cdot) \) i.e. \( \hat{h} = [\hat{h}_1 \cdots \hat{h}_N] \) where \( \hat{h}_j(u_j) \equiv \frac{\partial h_j(u_j)}{\partial u_j} \). Since the Jacobian determinant of the derivative matrix must be strictly non-zero as in

\[
\det \frac{dx}{ds} = \prod_{i=1}^{2} \prod_{j=1}^{N} \det \left[ B_i \right] \hat{h}_j \neq 0 \text{ for a unique inverse solution to exist},
\]

it then follows that the matrices \( B_i \) must contain \( N \) basis vectors and that \( \hat{h}_j(u_j) \neq 0 \) for all \( u_j \in \mathbb{R} \). Hence, the inverse to the empirical nonlinear model in (1) is given by \( \hat{s}(t) = f^{-1}(x(t)) = B_1^{-1} \left( h^{-1}(B_2^{-1}x(t) - \zeta) \right) \) provided that

\[
\prod_{i=1}^{2} \prod_{j=1}^{N} \det \left[ B_i \right] \hat{h}_j(u_j) \text{ is non-zero for all } u_j \in \mathbb{R}.
\]

Let \( D \) denotes the derivative operator and therefore the Jacobian matrix of the inverse model is related to the forward model as

\[
Df^{-1}(x(t)) = Df^{-1}(f(s(t))) = [D(f(s(t)))]^{-1} = B_1^{-1} \text{diag}[\hat{h}^{-1}]B_2^{-1}
\]
where $\hat{h}^{-1}$ denotes the element-by-element inverse of $\hat{h}$ and the determinant is given by

$$
\det \frac{\partial \hat{s}}{\partial \hat{x}} = \det \left[ \frac{\partial x}{\partial \hat{s}} \right]^{-1} = \prod_{i=1}^{2} \prod_{j=1}^{N} \det [B_i^{-1}] \hat{h}_j^{-1}
$$

(41)

which is strictly bounded if and only if $\prod_{i=1}^{2} \prod_{j=1}^{N} \det [B_i] \hat{h}_j(u_j)$ is not zero. □
Figure 1

Figure 2
Figure 3
Figure 4
Figure 5
Performance Index for MLP (L=3) at $p = 1$ with varying lambda

**Figure 6**

Performance Index for MLP (L=3) at $p = 2$ with varying lambda

**Figure 7**
Figure 8
Figure 9
Figure 10
Recovered signals using Kernel ICA

Recovered signals using PNN

Recovered signals using MLP (L=3) with lambda = 1 & p = 1

Recovered signals using MLP (L=3) with lambda = 1 & p = 2

Figure 11
Figure 12
$N_o = L$

Initialise $\{w_{ij}^{(l)}\}_{i=1}^{N_o}$ & $\{\theta_{ij}^{(l)}\}_{i=1}^{N_o-1}$

Parameter updates using Eqns. (32) & (34)

Converge?

Yes

Is $\mathbb{R}(D(N_o)) > N^2 - N$?

Yes

No

Stop

$N_o \leftarrow N_o - 1$
FIGURE/CHART CAPTIONS

**Figure 1:** Empirical nonlinear mixing model.

**Figure 2:** Proposed demixing model based on the L-layer Perceptron.

**Figure 3:** Signals in experiment 1.
- (a) Original.
- (b) Observed.

**Figure 4:** Recovered signals in experiment 1 under SNR=20dB.
- (a) Linear ICA.
- (b) 2-layer Perceptron without structural complexity learning.
- (c) 3-layer Perceptron with global structural complexity learning.
- (d) 3-layer Perceptron with local structural complexity learning.

**Figure 5:** Performance index of each demixer in experiment 1.

**Figure 6:** Performance index of the 3-layer Perceptron using $p = 1$.

**Figure 7:** Performance index of the 3-layer Perceptron using $p = 2$.

**Figure 8:** Histogram of the 3-layer Perceptron weights.
- (a) Without structural complexity learning.
- (b) With global structural complexity learning.
- (c) With local structural complexity learning.
Figure 9: Histogram of the 4-layer Perceptron weights.
   (a) Without structural complexity learning.
   (b) With global structural complexity learning.
   (c) With local structural complexity learning.

Figure 10: Speech signals in experiment 2.
   (a) Original.
   (b) Observed.

Figure 11: Recovered signals in experiment 2 under SNR=30dB.
   (a) Kernel ICA.
   (b) PNN.
   (c) 3-layer Perceptron with global structural complexity learning.
   (d) 3-layer Perceptron with local structural complexity learning.

Figure 12: Performance index of each demixer in experiment 2.

Chart 1: Layer-Trimming flowchart.