Abstract: In this paper, the problem of blind separation of independent sources in nonlinear mixtures is considered. We focus our work on a new type of nonlinear mixture in which a linear mixing matrix is sandwiched between two mutually reverse nonlinearities. The demixing system culminates to a novel Weierstrass Network which is shown to successfully restore the original source signals under the nonlinear mixing conditions. The corresponding parameter learning algorithm for the proposed network is presented through formal mathematical derivation. This paper shows for the first time a new result based on the theory of Forward Series and Series Reversion which is integrated into a neural network to implement the proposed demixer. Simulations including both synthetic and recorded signals have been carried out to verify the efficacy of the proposed method. We demonstrate that the Weierstrass Network outperforms other tested ICA methods (linear ICA, RBF and MLP network) in terms of speed and accuracy.

Keywords: Independent component analysis, blind signal separation, series reversion, neural network and adaptive signal estimation.
1. INTRODUCTION

During the last decade, tremendous developments have been achieved in Independent Component Analysis (ICA), particularly in array signal processing and signal restoration techniques. The principle aim of ICA is to extract a set of signals as independent as possible from only a set of observations. It is well-known that ICA is closely related to Blind Signal Separation (BSS) and many potential exciting applications of ICA have attracted considerable amount of attention in both science and technology [1-14 and reference therein]. However, most existing ICA algorithms focus on linear distortion which may not accord with practical applications [7-14]. In biomedical cases, many physiological signals are nonlinearly distorted and thus the identification of nonlinear dynamics should be taken into consideration, e.g. the auditory nervous system is modelled as a memoryless nonlinear system. Another instance is the recording of multiple speech source signals by carbon-button microphones which introduce some form of nonlinearity [7, 15, 16]. For nonlinear mixing model, linear algorithms fail to extract original signals and become inapplicable since the assumption of linear mixtures is violated and the linear algorithm cannot compensate for the information distorted by the nonlinearity. Hence, the search for a nonlinear solution becomes urgent and paramount in both theoretical and practical levels.

In current literature, nonlinear ICA has mostly concentrated on combining with different kinds of neural networks. In general, these methods can be classified into either generative approaches or signal transformation approaches [13]. In generative approaches, the aim is to find a specific model which represents how the observations are generated and the solution consists of estimating both the source signals and the mixing mapping whereas signal transformation methods construct the separation system and estimate
the unknown source signals directly. In both cases, the implementations usually involve the use of neural networks and differ only in terms of cost functions and learning algorithms.

Pajunen et al [17] provided one of the earliest nonlinear ICA solutions by using the Self-Organizing Maps (SOM). Although theoretically the output of the SOM network can provide the statistically independent vectors, there is no guarantee that the original source signals can be recovered by the SOM. Since the theoretical foundation of the SOM algorithm is based on rectangular map, the main limitation of SOM lies in the inevitable distortion when the source signals differ considerably from the uniform distribution. To overcome the disadvantages associated with SOM, Bishop et al [18], Pajunen and Karhunen [19] propose the generative topographic mapping (GTM) approach. However, in order to apply non-uniformly distributed source signals, the GTM method requires the known probability density function (PDF) of the source signals, which may limit the applications of this method. Signal transformation methods based on Radial-Basis Function (RBF) [20] and Multilayer Perceptron (MLP) [7] neural networks have recently drawn a substantial amount of attention for their flexible nonlinear capability. Under the nonlinear condition, both methods provide acceptable performance. RBF-based system can provide fast convergence at the cost of less accuracy whereas MLP can recover the original signals more precisely but suffer from high computational complexity. Besides the structure of the network, the performance of the demixer also depends on the selection of the nonlinear activation function in the hidden neurons. Networks for nonlinear ICA such as SOM [17], GTM [18, 19], RBF [20] and MLP with sigmoidal nonlinearity [7] are intrinsically nonlinear because of the utilisation of fixed nonlinearities in the hidden neurons. However, the execution by using the fixed degree of nonlinearity will lead to the oversized network, which inevitably subjects to huge computational complexity. Also, neural network-based solutions often cause the generation of non-trivial mappings that result in
arbitrary independent components which are nonlinearly related to the original source signals. In [8], it is argued that the situation is accentuated especially when an oversized network is used which subsequently leads to ‘overfitting’. Hence, to prevent the generation of arbitrary independent outputs, one approach is to allow the neural network demixer to control its inherent capability from ‘overfitting’. Therefore, instead of using a fixed form of nonlinearity in the hidden neurons, we propose to design a demixer whereby its intrinsic nonlinearity can be flexibly controlled.

The objective of this paper is to perform nonlinear signal transformation on the observed signals such that the resulting transformed signals are mutually as independent as possible and statistically as close as possible to the source signals. We centralise our work in a new type of mixture recently proposed in [22] where a regular linear mixing matrix is slotted into the middle of two mutually inverse nonlinear functions. The motivation of using this type of mixture is elaborated in Section 2. Given such mixing model, our aim is to construct a nonlinear demixer and develop an efficient algorithm for adapting its parameters. The key feature of the proposed work is the design of a special type of neural-based demixer to equalise the mixing system and to combat the intrinsic problem of generating arbitrary independent components. This is further reinforced by using a specific form of polynomial expansion series based on the Weierstrass approximation theorem for the hidden neurons’ activation function. The proposed technique also requires the demixer to form an inverse representation of the hidden neurons’ activation function in one of the layers which subsequently leads to a type of polynomial series reversion. Taken together, this culminates to a nonlinear framework that merges the forward polynomial expansion series and the corresponding series reversion with a neural network. To the authors’ knowledge, this is the first time that the theory of forward series and series reversion is integrated into a neural network.
The organisation of the paper is as follows: Section 2 discusses a recently proposed nonlinear mixing model based on the theory of functional analysis. In Section 3, the theoretical foundation of demixing system using the Weierstrass Network is presented. This is followed by the theory of Series Reversion. By using the minimum mutual information as the cost function with some parameters constraints, a gradient-based parameter learning algorithm for the proposed demixer is subsequently derived. Finally, simulations including synthetic as well as real-life recorded signals are presented in Section 4. Three experiments are carried out to verify the efficacy of the proposed method.

2. PROBLEM FORMULATION AND ASSUMPTIONS

Conventional linear ICA approaches assume that the mixture is linear by virtue of its simplicity. However, this assumption is often violated and may not characterise real-life signals accurately. A realistic mixture needs to be nonlinear and concurrently capable of treating the linear mixture as a special case. Generally, a nonlinear ICA problem can be defined as follows: Given a set of observations \( \mathbf{x} = [x_1(t), x_2(t), \ldots, x_r(t)]^T \) which are random variables and generated as a mixture of independent components \( \mathbf{s} = [s_1(t), s_2(t), \ldots, s_q(t)]^T \) according to

\[
x_i = f_i(s_1, s_2, \ldots, s_q)
\]  

where \( f_i \) is an unknown differentiable bijective mapping, \( i = 1, 2, \ldots, r \) and \( t \) is the time or sample index, the method of ICA now consists of estimating both the mixture mappings \( f_i \)'s and the original sources \( s_i(t), i = 1, 2, \ldots, q \). In linear ICA, the mapping \( f_i \) corresponds to a linear function i.e.
\[ f_i(s_1, s_2, \ldots, s_q) = m_{i1} s_1 + m_{i2} s_2 + \cdots + m_{iq} s_q \] where the \( \{m_{ij}\}_{i,j=1}^{i=q,j=q} \) represents the set of mixing coefficients.

In nonlinear ICA, the separation problem becomes much more difficult than the linear case since the mixture is no longer pertained to the principle of linear superposition of the source signals. According to [23], one approach to examine the separating mappings belonging to a specific subspace \( \Xi(\Omega) \) parameterised by \( \Omega \) is to investigate the independence preservation equation which states that for all \( A \) within \( C_q \) where \( C_q \) is an \( \sigma \)-algebra on \( \mathbb{R}^q \), there exists

\[
\int_A dp_1(s_1)dp_2(s_2)\cdots dp_q(s_q) = \int_{H(A)} dp_1(y_1)dp_2(y_2)\cdots dp_q(y_q) \quad (2)
\]

Denoting \( T \) as the set of transforms that preserve independence and \( \Psi \) as the set

\[
\Psi = \{(p_1(s_1), p_2(s_2), \ldots, p_q(s_q)) \mid \exists H \in \Xi \setminus (T \cap \Xi), H(s) \text{ is independent}\} \quad (3)
\]

of all source signal distributions \( p(s) = \prod_{i=1}^q p_i(s_i) \) for which there exists a non-trivial mapping \( H \) belonging to the model \( \Xi \) and preserving the independence of the components of the source signals \( s \). Ideally, \( \Psi \) should be empty but this cannot be achieved. In the general case where the mapping \( H \) has no particular form which usually occurs in nonlinear model, independence preservation is a weak constraint for ensuring signal separability. Taleb and Jutten in [25] pointed out that statistical independence, which is sufficient in conventional linear ICA, is not strong enough to recover the sources without any distortion in the general nonlinear case. Hyvarinen et al [24] concluded that there always exist infinite solutions in nonlinear ICA problems if the mixing function \( f_i \) is not constrained. The sources can only be restored up to some unknown nonlinear functions but such indeterminacy may cause heavy distortion on the estimated sources.

To overcome the ill-posed nature of this problem, some form of constraints would need to be imposed. One approach, as has already been mentioned by Hyvarinen et al [24], is to impose structural constraints on to the
mixing function $f_i$. By proper design of the demixer, the structural constraint can effectively reduce the cardinality in the set $\mathcal{E} \setminus (T \cap \mathcal{E})$ so that the number of non-trivial mappings can be limited to be as small as possible. This explains the main reason as to why nonlinear mixing models with constraints are preferred over a general model in (1). The second approach is to impose some form of signal constraints to match the variation of the estimated signals statistics to be as close as possible to the original source signals statistics. In this paper, both the structural and signal constraints are jointly used to regularise the effects of the indeterminacy resulted from the independence preservation rule. The requirements of the above constraints do not necessarily limit the utilisation of the proposed algorithm since in a wide range of applications prior information in the form of signal constraints are readily available. For example, in wireless communications the transmitted signal is derived from finite alphabet and therefore has discrete PDF. In this case, the signal constraints can be directly obtained from the PDF.

The post-nonlinear model proposed by Taleb and Jutten [25] includes a linear mixing matrix followed by one layer of nonlinear distortion functions. Although the post-nonlinear structure is particular for its simplicity to analysis, it can model some systems reasonably well such as those that involve the use of nonlinear sensors. Furthermore, the separability analysis of the post-nonlinear mixture has been derived explicitly in [14, 25]. In fact, under some weak conditions the post-nonlinear demixer can estimate the original signals up to some permutation, scale and mean value ambiguities. However, the main drawback of this simple structure is that the problem becomes difficult and intractable as soon as the cross-channel nonlinear mixing distortion is introduced to the model.

A recent nonlinear mixing model with structural constraint has been proposed in [22] and is summarised in
Theorem 1. This model is originally developed from \[21\] and subsequently re-discovered in \[22\]. Its modelling capability is based on the theory of functional analysis and further culminates to a structure where a regular linear mixing matrix is slotted into the middle of two mutually inverse nonlinear functions. Instead of presenting the theorem formally, we give a paraphrase of the main theorem in \[21, 22\] with some specialisation to our proposed demixing system.

**Theorem 1:** Let \( F(f(x), f(y)) = f(x) \otimes f(y) \) where \( F \) is a functional that is continuous at least separately for two variables and satisfies the Abelian group structure, then there exists strictly monotonic continuous function \( f \) such that \( u \otimes v = f(f^{-1}(u) + f^{-1}(v)) \) and \( \alpha \otimes u = u \otimes u \otimes \cdots \otimes u = f \left( \alpha f^{-1}(u) \right) \) with \( \alpha \in \mathbb{R} \). Now, if a nonlinear system with \( q \) inputs and \( r \) outputs can be defined as \( x = [x_1 \ x_2 \ \cdots \ x_q]^T \) with \( x_i = g_i(s) = m_{i1} \otimes s_1 \oplus m_{i2} \otimes s_2 \oplus \cdots \oplus m_{iq} \otimes s_q \) where \( m_i \in \mathbb{R} \), \( s = [s_1 \ s_2 \ \cdots \ s_q]^T \) and \( s_j \) is the \( j \)th input signal, then the nonlinear system can be described as follows:

\[
\begin{bmatrix}
   g_1(s) \\
   g_2(s) \\
   \vdots \\
   g_q(s)
\end{bmatrix} = 
\begin{bmatrix}
   m_{11} \otimes s_1 \oplus m_{12} \otimes s_2 \oplus \cdots \oplus m_{q1} \otimes s_q \\
   m_{21} \otimes s_1 \oplus m_{22} \otimes s_2 \oplus \cdots \oplus m_{q2} \otimes s_q \\
   \vdots \\
   m_{q1} \otimes s_1 \oplus m_{q2} \otimes s_2 \oplus \cdots \oplus m_{qq} \otimes s_q
\end{bmatrix} = 
\begin{bmatrix}
   f_1(m_1^T f_1^{-1}(s)) \\
   f_2(m_1^T f_2^{-1}(s)) \\
   \vdots \\
   f_q(m_1^T f_q^{-1}(s))
\end{bmatrix}
\]

where \( M = [m_1 \ m_2 \ \cdots \ m_q]^T \) with dimension \( r \times q \) and \( m_i = [m_{i1} \ m_{i2} \ \cdots \ m_{iq}]^T \). In addition, if the nonlinear mapping functions \( \{f_i\}_{i=1}^q \) can assume the form of \( f_1 = f_2 = \cdots = f_q = f \), then (4) reduces to the following:

\[
x = f \left( M f^{-1}(s) \right)
\]

The model exemplified in (5) shall be known as the ‘mono-nonlinearity’ model. For simplicity, we constrain the number of sensors equal to the number of sources i.e. \( r = q = N \). From (5), we recognise that the
nonlinear mixture is fundamentally a synthesis of two nonlinear functions, one of which is the inverse of the other, and the linear mixing matrix is sandwiched between them. It is noted that the architecture of network from (5) may not have block by block physical representations. The generality in modelling by (5) is indeed stemmed from the theory of functional analysis where it is shown that (5) can represent at least 2-layer nonlinear mixing systems which are expected to provide more general description than the post-nonlinear systems [26]. In practice, the mixing system structures are not necessarily required to conform directly to the model in (5). In addition, one can prove that the maximum likelihood estimate of the source signals under the mono-nonlinearity model is given by

\[ \hat{s} = f(M^{-1}f^{-1}(x)) = f(Wf^{-1}(x)) \]  

(6)

where \( W \) is the demixing matrix. Given the observed signals only, our aim is to estimate \( f \) and \( W \) such that the estimated signals are as close as possible to the original source signals under certain conditions which are now detailed in the following section.

3. **WEIERSTRASS NETWORK FOR NONLINEAR ICA**

In this section, the Weierstrass Network using the Forward Series and Series Reversion is proposed to implement the demixer as expressed in (6). Our proposed technique uses the Weierstrass series as the control mechanism to regulate the degree of nonlinearity in the hidden neurons. Since the proposed demixer consists of two layers of mutually inverse nonlinearities, the estimation can be carried out by using the Weierstrass series instead of the hyperbolic functions (i.e. sigmoidal nature), which is commonly adopted in most conventional nonlinear ICA demixers such as the multilayer Perceptron (MLP) model. The use of the
Weierstrass series will avoid the intrinsic instability caused by the inverse hyperbolic tangent function especially in the region of the input space close to $\pm 1$. Another weakness of using the sigmoidal function is that if the nonlinear transfer functions do not fit into the parametric structure of a sigmoid, the performance of the learning rules may degrade [9]. Within the blind signal separation context, since the nonlinear mixing system is unknown, the use of the adaptive Weierstrass approximation series is expected to provide increased flexibility in matching the required implementation of the optimal demixing system in (6).

### 3.1 The Weierstrass Network as the Nonlinear ICA Demixer

In the Weierstrass Approximation Theorem [28], it is pointed out that for every continuous function $f : [c,d] \rightarrow \mathbb{R}$, there always exists a Weierstrass series $p(u,M,\{a_m\}_{m=0}^M)$ which can uniformly approximate $f$ with arbitrary accuracy i.e.

$$p(u,M,\{a_m\}_{m=0}^M) = a_0 + a_1 u + a_2 u^2 + \cdots = \sum_{m=0}^M a_m u^m$$

$$\forall \varepsilon > 0, \exists M > Q(\varepsilon), \forall u \in [c,d]: |f(u) - p(u)| < \varepsilon$$

(7)

where $M$ and $\{a_m\}_{m=0}^M$ are the order and coefficients of the series, respectively. Hence, according to the mono-nonlinear demixer in (6), a feedforward network using the Weierstrass Approximation Theorem is proposed as shown in Figure 1. This demixer structure shall be referred to as the Weierstrass Network. The hidden layer neurons perform the Weierstrass series to approximate the mixing mapping functions $f(u)$ and $f^{-1}(u)$ in (5). Accordingly, the outputs of the demixing system assume the following form for $i = 1,2,\ldots,N$: 
\[ y_{[3,i]} = f( y_{[2,i]} ) = \sum_{m=0}^{M_1} a_m y_{[2,i]}^m \]
\[ y_{[2,i]} = \sum_{j=1}^{N} w_{i,j} y_{[1,j]} \]
\[ y_{[1,i]} = f^{-1}( x_i ) = \sum_{n=1}^{M_2} b_n ( x_i - a_0 )^n \] (8)

where \( y_{[j,i]} \) denotes the \( i \)th output of the \( j \)th layer, \( \{ a_m \}_{m=0}^{M_1} \) and \( \{ b_n \}_{n=1}^{M_2} \) are the coefficients while \( M_1 \) and \( M_2 \) represents the order of the series expansion. In vector notation, (8) can be represented as
\[ y_{[3]} = f( y_{[2]} ) = \sum_{m=0}^{M_1} a_m y_{[2]}^m , \quad y_{[2]} = W y_{[1]} \quad \text{and} \quad y_{[1]} = f^{-1}( x ) = \sum_{n=1}^{M_2} b_n ( x - a_0 )^n \quad \text{with} \quad y_{[i]} = \begin{bmatrix} y_{[i,1]} & \cdots & y_{[i,N]} \end{bmatrix}^T. \]

The degree of the nonlinearity in the hidden layers can therefore be adaptively adjusted to approximate the objective nonlinear mixing functions in (5). As seen from the structure of the network, the number of the neurons in the 1st and 3rd layers is normally fixed to be equal to the number of observed signals (we assume that the matrix \( M \) is square). The accuracy of the network for approximation therefore mainly depends on the orders of the Weierstrass series, whereas in MLP it relies on the numbers of both nonlinear layers and neurons in each layer. Hence, to some degree, the proposed Weierstrass Network with \( 3N \) number of neurons is substantially simpler than a 2-hidden layer Perceptron. This is advantageous since it prevents the proposed network from saturating to an oversized network which would otherwise lead to the overfitting problem and generation of arbitrary independent components [8].

Of special interest in this line of work is [29] where a polynomial-based approach is presented and subsequently lead to the development of a polynomial neural network. However, unlike our proposed technique, the polynomial neural network uses only the forward polynomial expansion series to implement each hidden neurons’ activation function in every layer, very much similar to a MLP structure but with the polynomials replacing the sigmoidal functions. This approach, however, fails to take into account of the
asymmetry nature of the mixing model in (5) and therefore, leads to performance degradation.

### 3.2 Cost Function for the Weierstrass Network

In general, the primary goal of the demixing system is to obtain a set of signals as independent as possible. The cost function rooted in the Kullback-Leibler Divergence (KLD) [1-3] is commonly used in most blind signal separation problem:

\[
KLD\left(\left\{ p(y_{1,i})\right\} \left| \prod_{i=1}^{N} p(y_{3,i,j})\right.\right) = \int p(y_{1,i}) \log \left( \frac{p(y_{1,i})}{\prod_{i=1}^{N} p(y_{3,i,j})} \right) dy_{1,i}
\]

\[
= E(\log p(x)) - \log \left| \det \frac{dy_{1,i}}{dx} \right| - \sum_{i=1}^{N} E[\log p(y_{3,i,j})]
\]

where \(p_i(y_{1,i})\) represents the marginal PDF of the \(i\)th estimated signal at the output of the Weierstrass Network demixer. Nonetheless, in the general case where the mapping \(H\) in (3) has no particular form, the independence preservation in (9) is a weak constraint for ensuring signal separability which inadvertently results in non-uniqueness of solutions. Therefore, to reduce the indeterminacy of non-unique solution and compensate for the nonlinear distortion, a form of structural constraint has been imposed to the demixer which culminates to the design of a special type of neural-based demixer as shown in Figure 1. The solution is further reinforced by augmenting a set of signal constraints to the original KLD cost function as follows:

\[
J = -\log \left| \det \frac{dy_{1,i}}{dx} \right| - \sum_{i=1}^{N} \log \left( p_i(y_{1,i,j}) \right) + \sum_{i=1}^{N} \beta_i f^{(c)}_i (y_{3,i,j}, s_i)
\]

\[
f^{(c)}_i(y_{3,i,j}, s_i) = \sum_{j=1}^{N} \left[ \text{cum}(y_{3,i,j}, j) - \text{cum}(s_i, j) \right]^2
\]
where $\beta_i$’s are a set of constants to control the importance of the additional constraints and $\text{cum}(u, j)$ represents the $j^{th}$ order cumulant of $u$ and $D$ is the maximum order of the cumulant. In fact, these constraints imply the use of a priori information about the source distributions which is intended to match the outputs of the demixer to be as close as possible to the original source signals in terms of cumulants.

### 3.3 Series Reversion

As shown in Figure 1, due to the structure of the mono-nonlinearity, the implementation of the proposed demixer requires the inverse function of the Weierstrass series. It is possible to express the inverse function of a polynomial as a closed form when the order of the forward function is 3 or less; however, the problem becomes difficult and intractable as soon as the order increases. Since the accuracy of the approximation will affect the performance of the algorithm and the quality of the restored signals, the Weierstrass Network using order up to 3 may be not sufficient and as a result it is only limited to certain applications. The theory of the Series Reversion provides an alternative solution and further establishes the foundation for computing the inverse function of a general polynomial expansion. This allows the Weierstrass Network to generalise with arbitrary orders of the polynomial expansion for both forward and reverse series. The following theorems facilitate the tools for implementing the proposed Weierstrass Network.

**Theorem 2:** If the function $g(.)$ has a series expression as $g(u) = \sum_{m=0}^{M} a_m u^m$, then the inverse function can be given by the similar series expansion form of $g^{-1}(u) = \sum_{n=1}^{\infty} b_n (u-a_0)^n$ with the coefficients computed as...
where \( k_i + 2k_i + 3k_i + \cdots = n-1, \ k_i \geq 0, \ i = 2, 3, 4 \ldots \) and \( k_i = \left( n + \sum_{i=2}^{M} k_i \right) \). The expression in (12) is referred to as the Series Reversion\(^1\).

**Proof:** See [30].

According to Theorem 2, both the forward and the inverse functions can now be formulated as the Weierstrass series. It is not necessary that the orders of the forward and the reverse series are identical. However, in order to obtain reliable approximation, the reverse series is expected to have at least the same order as the forward one. Since the update of the reverse series is closely related to the forward series and the gradient based learning algorithm is used, the derivatives of the reverse series with respect to the coefficients in the forward series \( \frac{\partial g^{-1}}{\partial a_m} \) are necessary for deriving the learning algorithm. The following lemma achieves this by providing the relationship between \( b_n \)'s and \( a_m \)'s in terms of the first order derivatives.

**Lemma 1:** Given the coefficients of the reverse Weierstrass series as computed by (12), the differential of \( b_n \) with respect to \( a_m \)'s is given by

\[
\begin{align*}
\frac{db_n}{da_m} &= \sum_{i=1}^{M} \sum_{k_1, k_2, \ldots, k_i} (-1)^{n-1} \frac{n-1+\sum_{j=2}^{M} k_j}{n! \prod_{j=2}^{M} (k_j !)} \left( \prod_{j=3}^{M} a_j^{k_j} \right) k_m a_m^{n-1} \\
&= \left( \sum_{i=1}^{M} \sum_{k_1, k_2, \ldots, k_i} (-1)^{n-1} \frac{n-1+\sum_{j=2}^{M} k_j}{n! \prod_{j=2}^{M} (k_j !)} \left( \prod_{j=3}^{M} a_j^{k_j} \right) k_m a_m^{n-1} \right) da_m
\end{align*}
\]

**Proof:** From (12), it can be shown that

\(^1\) Occasionally, we will also refer to it as the reverse series in order to contrast it with the closed form inverse function of the forward series.
\[ b_n = \sum_{k_1, k_2, \ldots} (-1)^{\sum k_i} \frac{(n-1+\sum_{i=2}^{M_1} k_i)!}{n! \prod_{i=2}^{M_1} (k_i!)} a_1^{k_1} a_2^{k_2} a_3^{k_3} \cdots \] 

\[(14)\]

\[ = \sum_{k_1, k_2, \ldots} c(n, k_i) \cdot a_1^{k_1} a_2^{k_2} a_3^{k_3} \cdots \]

where \( c(n, k_i) = (-1)^{\sum k_i} \frac{(n-1+\sum_{i=2}^{M_1} k_i)!}{n! \prod_{i=2}^{M_1} (k_i!)} \) is an irrelevant constant. Hence the total differential of \( b_n \) depends only on \( \{a_u\}_{u=1}^{M_1} \) and is given by

\[
\begin{align*}
\frac{db_n}{da_u} &= \sum_{M_1}^{M} \sum_{k_1, k_2, \ldots} \left[ c(n, k_i) \left( \prod_{i=2}^{M_1} a_i^{k_i} \right) \right] \frac{d}{da_u} \\
&= \sum_{M_1}^{M} \sum_{k_1, k_2, \ldots} (-1)^{\sum k_i} \frac{(n-1+\sum_{i=2}^{M_1} k_i)!}{n! \prod_{i=2}^{M_1} (k_i!)} \left( \prod_{i=2}^{M_1} a_i^{k_i} \right) d a_u
\end{align*}
\]

\[(15)\]

This completes the proof.

Compared with the closed form inverse function, the analysis of the Series Reversion shows the followings:

(a) The derivation of the closed form of the inverse function becomes intractable as soon as the order of the forward series increases whereas the Series Reversion provides the solution as expressed in Theorem 2 and is independent of how large the forward order \( M_1 \).

(b) The closed form version of the inverse function cannot be formulated into one general expression as in (12). For example, when the 3\textsuperscript{rd} order Weierstrass series assumes the form of \( g(u) = u + a_ u^3 \), its inverse function can be written as
As the forward series changes to \( g(u) = a_u u^5 \) (e.g. a monomial), the inverse function is then given by

\[
g^{-1}(u) = \left( \frac{u}{a_u} \right)^{\frac{1}{5}}
\]  

(17)

It can be seen that there is no concrete link to combine (16) and (17) into a single formula when only the closed form expression is sought. As a result, for every order of the forward series, the closed form inverse function has to be re-derived independently. In the case when the order of the forward series is not assigned specifically, the closed form of the inverse function cannot be calculated in advance and therefore the entire learning algorithm (in the Section 3.4) will become futile. Furthermore, the various structures of the closed forms of the inverse function can cause an enormous obstacle to getting the general expression of the derivative \( \frac{\partial g^{-1}}{\partial a_n} \), which is necessary for the gradient based learning algorithm. In contrast, regardless of the forward order \( M_1 \), the general form of \( \frac{\partial g^{-1}}{\partial a_n} \) can be easily obtained for the Series Reversion by using 

\[
\frac{\partial g^{-1}}{\partial a_n} = \sum_{n=1}^{M} \left( \frac{\partial g^{-1}}{\partial b_n} \frac{\partial b_n}{\partial a_n} \right)
\]

based on Theorem 2 and Lemma 1.

Hence, in the learning algorithm, only \( \{a_n\}^{M_1}_{n=1} \) and \( W \) are the sets of parameters that need to be optimised since the reverse series can be updated through the forward one.

(c) We note that for the Series Reversion, the exact form of the reverse function can only be achieved if and only if the order of the reverse series is infinite. However, by using the analysis similar to that of the Taylor series, it is found that only the low-order terms in the reverse series play a central role in the estimation as the higher order terms tend to 0 within the input region of [-1, 1]. In simulations, we
show that the truncated reverse series despite its order is constrained to be identical to the forward series is able to maintain a relatively high level of performance.

### 3.4 Gradient Descent Based Parameter Learning Algorithm

In this section, the parameter learning algorithm for the proposed Weierstrass Network using the Series Reversion is derived. Starting from the total differential of $J$ in (10), our aim is to optimise the cost function by updating the weight $W$ and the coefficients of the forward Weierstrass series $\{a_n\}_{n=0}^{N}$. 

**Theorem 3:** Given the structure of the Weierstrass Network shown in Figure 1, the effective cost function for the demixer assumes the following form:

$$J = -\log \|\mathbf{W}\| - \sum_{i=1}^{N} \log \left| \sum_{n=1}^{M} m a_n y_{i(2)}^n \right| - \sum_{i=1}^{N} \log \left| \sum_{n=1}^{M} n b_n (x_i - a_0)^n \right| - \sum_{i=1}^{N} \left[ \log \left(p_i \left(y_{i(3)}\right) - \beta f_i^{-1} \left(y_{i(3)}, s_i\right)\right) \right]$$  

(18)

**Proof:** The derivative of the outputs with respect to the observations which are regarded as the inputs of the demixing system can be expressed as

$$\frac{dy_{i(3)}}{dx} = \text{diag} \left[ \frac{df \left(y_{i(2)}\right)}{dy_{i(2)}} \right] \cdot W \cdot \text{diag} \left[ \frac{df^{-1} \left(x\right)}{dx} \right]$$  

(19)

Since the forward and the inverse functions take the forms of the Weierstrass series:

$$f (u) = \sum_{n=1}^{M} a_n u^n \Rightarrow \frac{df \left(y_{i(2)}\right)}{dy_{i(2)}} = \sum_{n=1}^{M} ma_n y_{i(2)}^{n-1}$$

$$f^{-1} (u) = \sum_{n=1}^{M} b_n (u - a_0)^n \Rightarrow \frac{df^{-1} \left(x\right)}{dx} = \sum_{n=1}^{M} nb_n (x - a_0)^{n-1}$$  

(20)
where $M_1$ and $M_2$ denote the order of the forward and the reverse Weierstrass series respectively.

Therefore,

$$
\frac{dy_{[1]}}{dx} = \text{diag} \left[ \sum_{m=1}^{M_1} ma_m y_{[2]}^{m-1} \right] \cdot W \cdot \text{diag} \left[ \sum_{n=1}^{M_2} nb_n (x - a_n)^{n-1} \right]
$$

By inserting (21) into (10), the effective cost function can be rewritten as

$$
J = -\log|\det W| - \sum_{i=1}^{N} \log \left| \sum_{m=1}^{M_1} ma_m y_{[2]}^{m-1} \right| - \sum_{i=1}^{N} \log \left| \sum_{n=1}^{M_2} nb_n (x_i - a_n)^{n-1} \right| - \sum_{i=1}^{N} \log \left( p_i \left( y_{[1,i]} \right) \right) - \beta f^{i+1} \left( y_{[1,i]}, s_i \right)
$$

This completes the proof. Having derived the required cost function, we need to be equipped with the total differentials of the cost function prior to obtaining the parameter learning algorithm.

**Theorem 4:** According to the effective cost function shown in the formula (18), the total differential of $J$ is derived as follows:

$$
dJ = -\alpha \left[ dWW^{-1} \right] + \left[ \Psi + \beta \cdot \Phi \right]^T dy_{[1]} \\
\quad - \left[ \frac{1}{\sum_{m=1}^{M_1} ma_m y_{[2]}^{m-1}} \sum_{m=1}^{M_1} ma_m y_{[2]}^{m-1} \right] \left[ \sum_{m=1}^{M_1} m da_m y_{[2]}^{m-1} + \sum_{m=1}^{M_2} m(n-1) a_n \text{diag} \left( y_{[2]}^{n-2} \right) dy_{[2]} \right]
$$

where

$$
\frac{1}{\sum_{n=1}^{M_2} nb_n (x - a_n)^{n-1}} \sum_{n=1}^{M_2} nb_n (x - a_n)^{n-1} + \sum_{n=1}^{M_2} n(n-1) b_n \text{diag} \left( (x - a_n)^{n-2} \right) d(x - a_n)
$$
\[ \Psi = \left[ -\frac{d}{dy_{[1]} \log \left( p_i \left( y_{[1]} \right) \right)} \right] \ldots \left[ -\frac{d}{dy_{[N]} \log \left( p_N \left( y_{[N]} \right) \right)} \right]^T \]

\[ \beta = \left[ \beta_1 \ldots \beta_N \right]^T \]

\[ f = \left[ -\frac{d}{dy_{[1]} f_i \left( y_{[1]} \right)} \right] \ldots \left[ -\frac{d}{dy_{[N]} f_N \left( y_{[N]} \right)} \right]^T \]

and "\( \odot \)" denotes the Hadamard product.

**Proof:** For simplicity, we define that the total differential of \( J \) has the form of

\[ dJ = -\operatorname{tr} \left[ dWW^{-1} \right] - d\Omega_1 - d\Omega_2 + d\Theta \tag{25} \]

where \( \operatorname{tr}[.] \) represents the trace of a matrix, and

\[ \Omega_1 = \sum_{i=1}^N \log \left( \sum_{n=1}^N ma_n y_{[1]}^{m-2} \right) = \log \left( \det \left( \operatorname{diag} \left( \sum_{n=1}^N ma_n y_{[2]}^{n-1} \right) \right) \right) \tag{26} \]

\[ \Omega_2 = \sum_{i=1}^N \log \left( \sum_{n=1}^N nb_n \left( x - a_n \right)^{n-1} \right) = \log \left( \det \left( \operatorname{diag} \left( \sum_{n=1}^N nb_n \left( x - a_n \right)^{n-1} \right) \right) \right) \tag{27} \]

\[ \Theta = -\sum_{i=1}^N \left[ \log \left( p_i \left( y_{[1]} \right) \right) - \beta_i f_i \left( \left( y_{[1]} \right) \right) \right] \tag{28} \]

To further simplify the derivation, we define \( \varphi_1 = \operatorname{diag} \left( \sum_{n=1}^N ma_n y_{[2]}^{m-1} \right) \) and \( \varphi_2 = \operatorname{diag} \left( \sum_{n=1}^N nb_n \left( x - a_n \right)^{n-1} \right) \). It then follows that

\[ d\varphi_1 = d \left[ \sum_{n=1}^N ma_n y_{[2]}^{m-1} \right] = \sum_{n=1}^M m \, da_n y_{[2]}^{m-1} + \sum_{n=1}^M m \left( m - 1 \right) a_n \operatorname{diag} \left( y_{[2]}^{m-2} \right) dy_{[2]} \tag{29} \]

\[ d\varphi_2 = \sum_{n=1}^N n \, db_n \left( x - a_n \right)^{n-1} + \sum_{n=1}^N n \left( n - 1 \right) b_n \operatorname{diag} \left( \left( x - a_n \right)^{n-2} \right) dx - \sum_{n=1}^M n \left( n - 1 \right) b_n \left( x - a_n \right)^{n-2} da_n \tag{30} \]

\[ d\Omega_1 = d \left( \log \left| \det \varphi_1 \right| \right) = \operatorname{tr} \left[ d \varphi_1 \varphi_1^{-1} \right] \tag{31} \]

\[ d\Omega_2 = d \left( \log \left| \det \varphi_2 \right| \right) = \operatorname{tr} \left[ d \varphi_2 \varphi_2^{-1} \right] \tag{32} \]

\[ d\Theta = \Psi^T dy_{[1]} \tag{33} \]
where

$$\psi = \psi + \beta \mathbf{f} = \left[ \frac{d}{dy_{[13]}} \log \left( p_n(y_{[13]}) \right) + \beta \left( \frac{d}{dy_{[13]}} f_{[13]}(y_{[13]}, s_{[13]}) \right) \right] \ldots \left[ \frac{d}{dy_{[13]}} \log \left( p_n(y_{[13]}) \right) + \beta \left( \frac{d}{dy_{[13]}} f_{[13]}(y_{[13]}, s_{[13]}) \right) \right]^T$$

By using Lemma 1, we can obtain the total derivatives of \{b_{[13]}\}_{m=1}^M \text{ to } \{a_{[13]}\}_{m=1}^M \text{ as follows:}

$$db_m = \sum_{k_1, k_2, \ldots} \left( -1 \right)^{k_1} \sum_{n=1}^{M} \left( \frac{n-1}{n!} \prod_{i=1}^{k_1} (k_i !) \right) \sum_{i=1}^{M} a_i^k a_i^{k-1} \right) d\alpha_m = \sum_{n=1}^{M} \xi n_{[n,m]} d\alpha_m$$

Finally, inserting (31)-(33) and (35) into (25), we obtain the total differential of \(J\) expressed as

$$dJ = -tr \left[ \frac{\partial \mathbf{WW}^{-1} \mathbf{f}}{\partial y_{[13]}} \right] + \psi \mathbf{f} \mathbf{dy}_{[13]}$$

This completes the proof.

In addition, the function \(\psi = \left[ \psi_1(y_{[13]}) \ldots \psi_N(y_{[13]}) \right]^T = \left[ \frac{d}{dy_{[13]}} \log \left( p_n(y_{[13]}) \right) \right] \ldots \left[ \frac{d}{dy_{[13]}} \log \left( p_n(y_{[13]}) \right) \right]^T\) can be estimated either using probability series expansion such as the Gram-Charlier [27] and Edgeworth series [31] or non-parametrically using the kernel density estimation method [6].
**Theorem 5:** According to the minimum mutual information criteria, the stochastic gradient based parameter learning algorithm for the Weierstrass Network mono-linearity demixer can be obtained by

\[
W(t+1) = W(t) + \mu_w \left[ 1 - \sum_{m=1}^{M} m(m-1)a_m \text{diag}(y_{[2]}) \left( \phi_1^{\text{y}_{[1]}} - \sum_{m=1}^{M} m a_m \text{diag}(y_{[2]}) \right) \right] W(t) \tag{38}
\]

\[
a_m(t+1) = a_m(t) - \mu_a \left( 1 + \text{diag}^{-1}(\phi) \left[ W \sum_{n=1}^{N} n b_n (x-a_n)^{(m-1)} \right] \right) \tag{39}
\]

\[
a_m(t+1) = a_m(t) - \mu_a \left( 1 + \text{diag}^{-1}(\phi) \left[ W \sum_{n=1}^{N} n b_n (x-a_n)^{(m-1)} \right] \right) \tag{40}
\]

where \( \mu_w \) and \( \mu_a \) are the learning rate of the weights \( W \) and the coefficients \( a_m \)'s respectively;

\[
\phi_1 = \left[ \frac{1}{\sum_{m=1}^{M} m a_m y_{[2]}} \ldots \frac{1}{\sum_{m=1}^{M} m a_m y_{[2]}} \right] \tag{41}
\]

\[
\phi_2 = \left[ \frac{1}{\sum_{n=1}^{N} n b_n (x-a_n)^{(m-1)}} \ldots \frac{1}{\sum_{n=1}^{N} n b_n (x-a_n)^{(m-1)}} \right] \tag{42}
\]

\[
\hat{\xi}_{\pi[n,m]} = \sum_{k_2 \ldots k_\pi} \left( (-1)^{k_2+\ldots+k_\pi} \prod_{k=2}^{\pi} \left( \frac{1}{\prod_{m=1}^{M} a_m} \right) a_m^{k_m-1} \right) \tag{43}
\]

**Proof:** According to the structure of the Weierstrass Network, we can find

\[
dy_{[1]} = \sum_{m=1}^{M} \sum_{n=1}^{N} \hat{\xi}_{\pi[n,m]} (x-a_n)^{\pi} da_m + \sum_{m=1}^{M} \sum_{n=1}^{N} n b_n \text{diag}(y_{[2]}) dx - \sum_{n=1}^{N} n b_n (x-a_n)^{(m-1)} da_m \tag{44}
\]
Assuming that the mono-nonlinear demixer is used, i.e. \( f_1 = f_2 = \cdots = f_N = f \), the gradient can be calculated by substituting (44)-(46) for (23).

\[
\begin{align*}
\frac{\partial J}{\partial \mathbf{W}} &= \left[ I - \left( \sum_{n=1}^{M} m(n-1) a_n \text{diag}(y_{[2]}^m) \right) \sum_{n=1}^{M} m(n-1) y_{[2]}^m \right] \mathbf{W}^T
\end{align*}
\]  (47)

\[
\frac{\partial J}{\partial a_n} = \mathbf{W}^T \left[ \left( \sum_{n=1}^{M} (n-1) y_{[2]}^m \right) \mathbf{W} \right]
\]  (48)

\[
\frac{\partial J}{\partial a_m} = \mathbf{W}^T \left[ \left( \sum_{n=1}^{M} m(n-1) a_n \text{diag}(y_{[2]}^m) \right) \mathbf{W} \right]
\]  (49)

Since the gradient descent based learning algorithm is used,

\[
\begin{align*}
\mathbf{W}(t+1) &= \mathbf{W}(t) - \mu_w \frac{\partial J}{\partial \mathbf{W}} \mathbf{W}^T \\
\alpha_m(t+1) &= \alpha_m(t) - \mu_{\alpha_m} \frac{\partial J}{\partial a_m} \quad ; \quad m = 0,1,\ldots,M_1
\end{align*}
\]  (50-51)

Inserting (47)-(49) into (50)-(51), this leads to the stochastic gradient descent learning algorithm. This completes the proof.
Close inspection of the expressions in (38)-(40) reveals that the orders of both the forward and the reverse Weierstrass series can be selected \textit{a priori}. As mentioned above, the selection may affect the performance in terms of accuracy and computational complexity of the algorithm. Since the Weierstrass Network can treat the linear separation as a special case, the Weierstrass series is usually initialised to a zero-preserving linear function, i.e. $a_0 = 0$, $a_1 = 1$ and $\{a_n\}_{n=2}^{M} = 0$. So starting from the linear function, the Weierstrass series adjusts its nonlinearity towards the targeted mixing function in (5) by adaptively updating the coefficients of the series. In addition, the learning rates $\mu_w$ and $\mu_n$ need to be chosen carefully. The selection of the learning rate is the trade-off between the speed and the steady-state fluctuation of the convergence. This is especially more pronounced in the nonlinear case where $\mu_w$ and $\mu_n$ represent the learning rates of the linear demixing matrix and the nonlinear approximation, respectively, and they should match with each other in order to provide the best performance. Since the Weierstrass series is used, small $\mu_n$'s can change the nonlinearity gradually and keep the stability of the demixer; while the speed of the convergence is mainly dependent on $\mu_w$. Therefore, the selection that $\mu_w$ is larger than $\mu_n$ may be expected to provide better performance than the one based on same fixed values. Monte-Carlo experiments have been conducted on the selection of the learning rates and the results show that $\mu_w \in [0.001, 0.0001]$ and $\mu_n \in [0.00001, 0.00005]$ yield relatively good performance in terms of speed and accuracy. The weight update expressed in (38) closely resembles the natural gradient descent algorithm [32] used in linear ICA except that the present paper deals with the nonlinear demixer using the Weierstrass Network. The main motivation of using the natural gradient in the nonlinear demixer is mainly for the improvement of the computational complexity since it avoids computing the inverse matrix and increases the speed of convergence of the learning algorithm.
4. RESULTS

In this section, three experiments are carried out to verify the efficacy of the proposed method. The first two experiments have similar settings but differ in terms of the source signals being used, one of which is subgaussian signals and the other is supergaussian. In the third experiment, recorded speech signals are tested to complete the investigation.

Experiment 1

In this experiment, five subgaussian signals are generated synthetically as the original sources which are then passed through the mono-nonlinearity mixing system. In order to assist us in gaining insights of the efficacy of the mono-nonlinearity model, we start from the following study in which five observed signals are considered. We assume that the sources are nonlinearly coupled up to a $5^{th}$ order system given by

\[
x_i = \frac{s_4 - s_2 - 2s_1 + 2s_2s_3 + s_1^2s_4 - s_2s_4 + n_i}{-1 + s_2 + 2s_1s_4 - 2s_1s_2 - s_1^2 + s_2s_4}
\]

\[
x_2 = \frac{s_1s_2s_3 + s_1s_4s_5 - s_1 + s_4 - s_1s_2s_3 + s_1s_4s_5 - s_4}{-1 - s_1s_4 + s_2s_4 + s_1s_4 + s_2s_4s_5 - s_4s_5 + s_1s_4 + s_2s_5}
\]

\[
x_3 = \frac{-2s_1s_2s_3 - s_2 + s_1 - s_2s_3 + s_2s_3 + n_3}{s_3s_2 - 2s_2s_3 - 1 + 2s_1s_5 + s_2s_5s_3 - s_2s_5}
\]

\[
x_4 = \frac{-s_2 - s_1 + s_2 + s_1s_2s_3 - s_1s_2s_3 - s_1s_2s_3 + s_1s_2s_3 + n_4}{1 + s_1s_2s_3s_4 - s_1s_2s_3 - s_1s_2s_3 - s_1s_2s_3 - s_1s_2s_3}
\]

\[
x_5 = \frac{-s_2 - s_1 + s_2 + s_1s_2s_3}{-1 - s_1s_5 + s_2s_4 + s_1s_5} + n_5
\]

where $\{s_i\}_{i=1}^5 \in (-1,1)$, $\{x_i\}_{i=1}^5$ and $\{n_i\}_{i=1}^5$ are the source signals, observed signals and noise, respectively. By using the mono-nonlinearity mixing model, there exist $f(s_i) = \frac{e^{s_i} - e^{-s_i}}{e^{s_i} + e^{-s_i}}$ and $f^{-1}(s_i) = \frac{1}{2} \ln \left(1 + \frac{1}{1 - s_i}\right)$ which are both strictly monotonic continuously differentiable functions. Substituting these functions into (52), the latter can be represented as the mono-nonlinearity mixing system by
\[ x_1 = \tanh\left[2\tanh^{-1}(s_1) + \tanh^{-1}(s_2) - \tanh^{-1}(s_3)\right] + n_1 \]
\[ x_2 = \tanh\left[\tanh^{-1}(s_1) - \tanh^{-1}(s_2) + \tanh^{-1}(s_3) + \tanh^{-1}(s_4)\right] + n_2 \]
\[ x_3 = \tanh\left[-2\tanh^{-1}(s_2) + \tanh^{-1}(s_3) - \tanh^{-1}(s_4)\right] + n_3 \]
\[ x_4 = \tanh\left[\tanh^{-1}(s_1) + \tanh^{-1}(s_2) - \tanh^{-1}(s_3) - \tanh^{-1}(s_4)\right] + n_4 \]
\[ x_5 = \tanh\left[-\tanh^{-1}(s_1) + \tanh^{-1}(s_2) - \tanh^{-1}(s_3)\right] + n_5 \]

(53)

Alternatively, in the vector form,

\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
  x_3 \\
  x_4 \\
  x_5 
\end{bmatrix} = \tanh\left(\begin{bmatrix}
  2 & 1 & 0 & -1 & 0 \\
  1 & 0 & -1 & 1 & 1 \\
  0 & -2 & 1 & -1 & 0 \\
  1 & 1 & -1 & 0 & -1 \\
  -1 & 0 & 0 & 1 & -1 \\
\end{bmatrix} \begin{bmatrix}
  s_1 \\
  s_2 \\
  s_3 \\
  s_4 \\
  s_5 
\end{bmatrix}\right) + \begin{bmatrix}
  n_1 \\
  n_2 \\
  n_3 \\
  n_4 \\
  n_5 
\end{bmatrix}
\]

(54)

Therefore, the described mono-nonlinearity model can simplify the complicated nonlinear expressions and show its generalisation in modelling. The source signals, shown in Figure 2 (a), are expressed as

\[
\begin{bmatrix}
  s_1(t) \\
  s_2(t) \\
  s_3(t) \\
  s_4(t) \\
  s_5(t) 
\end{bmatrix} = \begin{bmatrix}
  \text{Binary signal} \\
  \sin(1600\pi t) \\
  \sin(600\pi t + 6\cos(120\pi t)) \\
  \sin(180\pi t) \\
  \text{Uniformly distributed signal} 
\end{bmatrix}
\]

(55)

Simulation has been carried out iteratively with 2500 samples and the sampling frequency is 1kHz. The source signals are then mixed according to the mono-nonlinearity model as \( x = \tanh(M\tanh^{-1}(s)) \) where \( M \) is a \( 5 \times 5 \) mixing matrix uniformly distributed within \([-0.5, 0.5]\). Since all source signals have subgaussian distributions, the truncated 4th order Edgeworth expansion can be used to estimate the function \( \psi \). For the proposed Weierstrass Network, the order of each Weierstrass function is selected to be 5. All weights are initialised randomly. The learning rates for the weights and the coefficients \( a_m \) are set to 0.001 and 0.00003, respectively. In order to assess the performance of the proposed approach, we compare it with the following well-known algorithms in which the parameters are selected \textit{a priori} based on the previous experience of successful separation in simulation.

- Linear method based on Extended ICA [33];
• RBF network with 5 input nodes, 20 hidden neurons with Gaussian kernel function, and 5 output nodes [20];

• Feedforward MLP (FMLP) network with 5 input nodes, 15 hidden neurons and 5 output nodes [7].

Figure 2(b) shows the mixed signals through the mono-nonlinearity mixing model. In Figure 3, the recovered signals of four different tested algorithms are displayed. As a comparison, in order to evaluate the performance of the proposed Weierstrass Network in terms of convergence and accuracy, the following normalised mean square is used as the performance index:

\[
\rho = \frac{1}{NT} \sum_{t=1}^{T} \sum_{i=1}^{S} \left[ \frac{s_i(t)}{E(s_i^2)} \frac{y_{[13,1]}(t)}{\sqrt{E(y_{[13,1]}^2)}} \right]^2
\]

(56)

where \(T\) represents the length of the source signals in term of the number of samples.

The performance index of the tested algorithms, i.e. linear Extended ICA method, RBF, FMLP network and the proposed Weierstrass Network, is shown in Figure 4. It is observed that the proposed Weierstrass algorithm outperforms the others in term of accuracy. Firstly, due to the intrinsic linear characteristic the linear method loses its capability to restore the original sources under nonlinear mixtures. In Figure 5, we show the effect of nonlinear distortion on the mixed signals. As can be seen, due to the nonlinear function, many informative linear mixing details have disappeared. Since the linear model does not take the nonlinear transformation into consideration, the linear ICA method cannot compensate for the lost of information and therefore fails to recover the source signals. Secondly, in Figure 3(b), the visual performance of the separation by the RBF network seems acceptable. According to the analysis of the performance index
illustrated in Figure 4, the speed of convergence of the RBF network is between the MLP and the proposed Weierstrass Network. However, from Figure 4, it can also be seen that the accuracy of the RBF network is the worst among all tested nonlinear ICA methods. The reason lies in the use of fixed Gaussian kernel functions in the hidden layer where the Euclidean norm between the input vector and the centre is computed, which subsequently leads to local approximation to the nonlinear input-output mappings. As a result, the performance tends to degrade for rapidly fluctuating source signals. Finally, although the MLP neural network is powerful for its ability to approximate any continuous function, it suffers from huge computational complexity and the overfitting problem which lead to the slow convergence and generation of arbitrary independent components. The tested FMLP is based on the cascade linear-nonlinear-linear model which can treat the mono-nonlinearity mixing model as a special case. As discussed above, the 1st layer nonlinearity of the FMLP network need to be updated towards the inverse hyperbolic tangent function in the experiment. However, the inverse hyperbolic tangent function $\tanh^{-1}(\cdot)$ is highly unstable in the region close to $\pm 1$ and therefore difficult to be approximated using finite number of neurons with fixed activation functions. On the contrary, with the a priori information of the relationship between the two mutually inverse function at the hidden layers, the special structure of the Weierstrass Network helps to avoid the overfitting problem and provides better estimation accuracy among the tested algorithms.

**Experiment 2**

In order to test the efficacy of the proposed Weierstrass Network, same mixing procedure is applied to five supergaussian distributed source signals expressed as
where the impulsive noise is generated by \( 2(r(t) < 0.5) \cdot \log(r(t)) \), ‘\( \odot \)’ denotes the Hadamard product, \( r_1(t) \) and \( r_2(t) \) are the uniform distributed signals, the function \( \text{rem}(u,v) \) represents the remainder of \( u \) divided by \( v \) with \( t = 0,1,2,\ldots \). Except for the sources, the experiment uses the same settings as in Experiment 1. Due to the supergaussian nature of the sources, a suitable function for \( \Psi \) would be the hyperbolic tangent function which is closely linked to the Maximum Likelihood (ML) principle [33]. The original source signals, mixed signals and corresponding recovered signals by Extended ICA, RBF, FMLP and the proposed Weierstrass Network are shown in Figure 6 and 7, respectively. Similar to the experiment 1, the linear ICA method fails to extract the original source signals from the nonlinearly mixed observations, which is in accordance with the discussion in previous sections and provides the evidence on the essentiality of using nonlinear ICA. In Figure 8, we illustrate the performance of the nonlinear estimation by the Weierstrass approximation series. Given that the order of the forward Weierstrass series is 5, it can approximate the mixing nonlinear function \( \tanh(\cdot) \) in the space of \([-1, 1]\) with relatively high accuracy. Although the inverse function \( \tanh^{-1}(\cdot) \) is highly nonlinear and non-continuous at the asymptotic points \( n = \pm 1, \pm 3, \pm 5, \ldots \), the truncated reverse series still pertains to good estimation. The converged coefficients of the forward and the reverse Weierstrass series are listed in Table 1. Starting from a linear function, the Weierstrass approximation series can be adaptively updated towards the mixing functions. Since both \( \tanh(\cdot) \) and \( \tanh^{-1}(\cdot) \) functions are odd, the converged values of the even items stay around zero and those of the odd items change the nonlinearity of the functions.
To investigate the computational complexity of the proposed Weierstrass Network, the measurement is implemented on the Intel Pentium 4 3.00GHz processor and 2GB of RAM. The time consumption of one (batch) iteration based on 2500 samples is calculated and is tabulated in Table 2 along with the linear ICA, RBF and FMLP methods. From the table, it is inferred that the computational complexity of the proposed method is the highest among all methods. Despite its high of computational complexity, the proposed method is only 12% and 20% more demanding than MLP and RBF demixers, respectively.

**Experiment 3**

In the third experiment, in order to investigate the efficacy of the proposed scheme in practical term, two recorded speech signals and one music signal playing at background are used as the original source signals. Similar mixing model is applied to the sources, i.e. $x = \sinh^{-1}(M \sinh(s))$ where $M$ is a $3 \times 3$ mixing matrix randomly sampled within [-1, 1]. The mono-nonlinearity mixing system is expected to represent the combined recording amplifier [15] and the nonlinearity due to the carbon-button microphones [16] working in the saturation region whose characteristics can be approximated by the hyperbolic function. The sampling frequency for the recorded signals is 22.05kbps. The parameter settings are identical to the first two experiments except that the learning rates are now changed to $\mu_w = 0.0001$ and $\mu_a = 0.00001$. Different orders of the Weierstrass series are applied in this experiment to investigate the influence caused by the truncation. The investigated orders are 5, 7 and 9. The original sources and the nonlinearly mixed signals are shown in Figure 9. In Figure 10, the restored signals via the linear ICA method, RBF, FMLP and the proposed Weierstrass Network are displayed. The performance of the algorithm is evaluated in term of convergence and accuracy, shown in Figure 11, using the performance index as expressed in the (56). Similar to the previous experiments, our analysis shows that the proposed method is successful in recovering real-life
recorded signals. Concurrently, we also compare the performance of the proposed Weierstrass Network with different orders. The convergences of the performance index for the 5\textsuperscript{th}, 7\textsuperscript{th} and 9\textsuperscript{th}-order Weierstrass Network in Figure 11 are almost identical but differ in terms of steady-state values and the positions of the second sharp drop. It is seen that the 9\textsuperscript{th}-order Weierstrass Network outperforms others in term of accuracy. The proposed 9\textsuperscript{th}-order Weierstrass Network improves the separation accuracy by 88.27\% compared with the linear Extended ICA method, 71.75\% with the RBF network, and 52.60\% with the FMLP method. In comparing the performance among the Weierstrass Network at different orders, the improvement achieved from the 5\textsuperscript{th}-order to the 7\textsuperscript{th} is significant as noted by decrement of 0.0886 whereas from the 7\textsuperscript{th} to the 9\textsuperscript{th} order, the decrement is only 0.0201. This result accords with the discussion that the low-order items in the Weierstrass series dominate the performance of the approximation. It is consequently implied that there exists a latent optimal order of the Weierstrass series which can balance the computational complexity and the accuracy of the Weierstrass network. Based on the results acquired from both synthetic and recorded signals, the proposed Weierstrass Network has demonstrated its efficacy in separating signals under the mono-nonlinearity mixture. The success is consecutively followed by the MLP and RBF but the separation results achieved by the linear method falls far from optimal and this indicates the crucial need for nonlinear separation techniques.

The proposed method is developed under the assumption that the mixing model is noiseless. Since the demixer requires inversion of one of the hidden layers, it is possible that the sensor noise can be enhanced. If the class of sigmoid functions is to be used, the sensor noise will be progressively amplified as the noise level gradually reaches the asymptotes. For example, if a Perceptron based on the hyperbolic tangent function is used, the sensor noise will be heavily amplified in the region close to ±1. This shows another reason as to why Perceptron model is not suitable for separating the mono-nonlinearily mixed signals in (5).
The effect of noise enhancement is less severe in the proposed demixer since only finite order is used for both forward and reverse series as shown in Figure 8. The determination of the optimal Weierstrass order when the system is embedded in noise is beyond the coverage of the present paper. Nevertheless, as a rule of thumb, the Weierstrass order should be selected to be as high as possible if computational complexity can be afforded. However, as soon as the sensor noise level increases, the order should be decreased in order to prevent noise amplification. An adaptive approach for selecting the optimal order for both the forward and reverse series is currently undertaken.

5. CONCLUSION

A new demixing scheme for separating nonlinearly mixed signals using the Weierstrass Network is proposed. The key features of the proposed approach can be summarised as follows: (a) utilisation of both the structural constraint and the signal constraint in the design of the demixer; (b) a set of adaptively adjustable nonlinear function facilitated by the Weierstrass series is performed as the hidden neurons’ activation function. This leads to the use of smaller network size and ameliorates the problem of producing arbitrary independent components; (c) the theory of Series Reversion is integrated with neural networks to compute the inverse of the forward series. The weakness of proposed method lies in its demand of high computational intensity as the order of the Weierstrass series increases. Future work will focus on the issues of reducing the complexity of the overall system and simplifying the parameter learning algorithm to render the proposed method more practical.
6. REFERENCE


29. WOO, W.L. and KHOR, L.C.: 'Blind restoration of nonlinearly mixed signals using multilayer


Figure 1  The Weierstrass Network as the nonlinear demixer

\[ y_{[1]} = f^{-1}(x) \quad y_{[2]} = Wy_{[1]} \quad y_{[3]} = f(y_{[2]}) \]

Figure 2  Signals in experiment 1. (a) Original subgaussian sources. (b) Nonlinearly mixed sources.
Figure 3  Recovered subgaussian signals by (a) Linear ICA method. (b) RBF Network.
(c) FMLP Network. (d) Proposed Weierstrass Network.
**Figure 4** The performance index of the tested algorithms for experiment 1.

**Figure 5** Linear and nonlinear distortion of the mixed signals.
Figure 6  Signals in experiment 2. (a) Original supergaussian sources. (b) Nonlinearly mixed sources.
Figure 7  Recovered supergaussian signals by (a) Extended ICA method. (b) RBF Network.

(c) FMLP Network. (d) Proposed Weierstrass Network.
Nonlinear Estimation of the mixing function $f(.)$

Hyperbolic tangent function

Estimated $\tanh(.)$ by the Weierstrass function

Inverse hyperbolic tangent function

Figure 8  Weierstrass approximation of (a) $\tanh(.)$. (b) $\tanh^{-1}(.)$

Source Signals Including Two voices and One Music

No. of Samples

Figure 9  Speech signals in experiment 3. (a) Real-life recorded sources. (b) Nonlinearly mixed sources.
Figure 10  Recovered speech signals by (a) Extended ICA method. (b) RBF Network. (c) FMLP Network. (d) Proposed Weierstrass Network.
Figure 11 The performance index of the tested algorithms for experiment 3.
<table>
<thead>
<tr>
<th>Coefficients</th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$a_3$</th>
<th>$a_4$</th>
<th>$a_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Converged Values</td>
<td>0.97908</td>
<td>0.026176</td>
<td>-0.1751</td>
<td>0.01148</td>
<td>-0.09964</td>
</tr>
</tbody>
</table>

**Table 1 (a)** The converged values of the coefficients in the forward Weierstrass series

<table>
<thead>
<tr>
<th>Coefficients</th>
<th>$b_1$</th>
<th>$b_2$</th>
<th>$b_3$</th>
<th>$b_4$</th>
<th>$b_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Converged Values</td>
<td>0.95958</td>
<td>-0.026203</td>
<td>0.18045</td>
<td>-0.036528</td>
<td>0.21125</td>
</tr>
</tbody>
</table>

**Table 1 (b)** The converged values of the coefficients in the reverse series

<table>
<thead>
<tr>
<th>Tested Algorithms</th>
<th>Time consumption of one iteration (2500 samples)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed Weierstrass Network</td>
<td>0.7133s</td>
</tr>
<tr>
<td>Linear ICA (Extended ICA)</td>
<td>0.2170s</td>
</tr>
<tr>
<td>RBF Network (Enhanced K-means Approach)</td>
<td>0.5688s</td>
</tr>
<tr>
<td>FMLP Network</td>
<td>0.6290s</td>
</tr>
</tbody>
</table>

**Table 2** Time consumption of the tested algorithms
FIGURE CAPTION

Figure 1  The Weierstrass Network as the nonlinear demixer

Figure 2  Signals in experiment 1.
(a) Original subgaussian sources.
(b) Nonlinearly mixed sources.

Figure 3  Recovered subgaussian signals by
(a) Linear ICA method.
(b) RBF Network.
(c) FMLP Network.
(d) Proposed Weierstrass Network.

Figure 4  The performance index of the tested algorithms for experiment 1.

Figure 5  Linear and nonlinear distortion of the mixed signals.

Figure 6  Signals in experiment 2.
(a) Original supergaussian sources.
(b) Nonlinearly mixed sources.

Figure 7  Recovered supergaussian signals by
(a) Extended ICA method.
(b) RBF Network.
(c) FMLP Network.
(d) Proposed Weierstrass Network.

**Figure 8** Weierstrass approximation of (a) \( \tanh(\cdot) \). (b) \( \tanh^{-1}(\cdot) \)

**Figure 9** Speech signals in experiment 3.
(a) Real-life recorded sources.
(b) Nonlinearly mixed sources.

**Figure 10** Recovered speech signals by
(a) Extended ICA method.
(b) RBF Network.
(c) FMLP Network.
(d) Proposed Weierstrass Network.

**Figure 11** The performance index of the tested algorithms for experiment 3.
TABLE CAPTION

Table 1 (a)  The converged values of the coefficients in the forward Weierstrass series.

Table 1 (b)  The converged values of the coefficients in the reverse series.

Table 2  Time consumption of the tested algorithms.