

Research Article

Complex-Valued Adaptive Signal Processing Using Nonlinear Functions

Hualiang Li and Tülay Adalı

Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD 21250, USA

Correspondence should be addressed to Tülay Adalı, adali@umbc.edu

Received 16 October 2007; Accepted 14 February 2008

Recommended by Aníbal Figueiras-Vidal

We describe a framework based on Wirtinger calculus for adaptive signal processing that enables efficient derivation of algorithms by directly working in the complex domain and taking full advantage of the power of complex-domain nonlinear processing. We establish the basic relationships for optimization in the complex domain and the real-domain equivalences for first- and second-order derivatives by extending the work of Brandwood and van den Bos. Examples in the derivation of first- and second-order update rules are given to demonstrate the versatility of the approach.

Copyright © 2008 H. Li and T. Adalı. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Most of today's challenging signal processing applications require techniques that are nonlinear, adaptive, and with on-line processing capability. Also, there is need for approaches to process *complex-valued data* as such data arises in a good number of scenarios, for example, when processing radar and magnetic resonance data as well as communications data and when working in a transform domain such as frequency. Even though complex signals play such an important role, many engineering shortcuts have typically been taken in their treatment preventing full utilization of the power of complex domain processing as well as the information in the real and imaginary parts of the signal.

The main difficulty arises due to the fact that in the complex domain, analyticity, that is, differentiability in a given open set, as described by the Cauchy-Riemann equations [1] imposes a strong structure on the function itself. Thus the analyticity condition is not satisfied for many functions of practical interest, most notably for the cost (objective) functions used as these are typically real valued and hence nonanalytic in the complex domain. Definition of pseudogradients are used—and still not through a consistent definition in the literature—and when having to deal with vector gradients, transformations $\mathbb{C}^N \mapsto \mathbb{R}^{2N}$ are commonly used. These transformations are isomorphic and allow the use of real-valued calculus in the computations, which includes well-defined gradient and Hessians that can be at the end transformed

back to the complex domain. The approach facilitates the computations but increases the dimensionality of the problem and might not be practical for functions that are nonlinear since in this case, the functional form might not be easily separable to real and imaginary parts.

Another issue that arises in the nonlinear processing of complex-valued data is due to the conflict between the boundedness and differentiability of complex functions. This result is stated by Liouville's theorem as: *a bounded entire function must be a constant in the complex domain* [1]. Hence, to use a flexible nonlinear model such as the nonlinear regression model, one cannot identify a complex nonlinear function ($\mathbb{C} \mapsto \mathbb{C}$) that is bounded everywhere on the entire complex domain. A practical solution to satisfy the boundedness requirement has been to process the real and imaginary parts (or the magnitude and phase) separately through bounded real-valued nonlinearities (see, e.g., [2–6]). The solution provides reasonable approximation ability but is an ad hoc solution not fully exploiting the efficiency of complex representations, both in terms of parameterization (number of parameters to estimate) and in terms of learning algorithms to estimate the parameters as we cannot define true gradients when working with these functions.

In this paper, we define a framework that allows taking full advantage of the power of complex-valued processing, in particular when working with nonlinear functions, and eliminates the need for either of the two common engineering practices we mentioned. The framework we develop

is based on Wirtinger calculus [7] and extends the work of Brandwood [8] and van den Bos [9] to define the basic formulations for derivation of algorithms and their analyses in the complex domain. We show how the framework also naturally admits the use of nonlinear functions that are analytic rather than the pseudocomplex nonlinear functions defined using real-valued nonlinearities. Analytic complex nonlinear functions have been shown to provide efficient representations in the complex plane [10, 11] and to be universal approximators when used as activation functions in a single-layer multilayer perceptron (MLP) network [12].

The work by Brandwood [8] and van den Bos [9] emphasize the importance of working with complex-valued gradient and Hessian operators rather than transforming the problem to the real domain. Both contributions, though not acknowledged in either of the papers, make use of Wirtinger calculus [7] that provides an elegant way to bypass the limitation imposed by the strict definition of differentiability in the complex domain. Wirtinger calculus relaxes the traditional definition of differentiability in the complex domain—which we refer to as *complex differentiability*—by defining a form that is much easier to satisfy and includes almost all functions of practical interest, including functions that are $\mathbb{C}^N \mapsto \mathbb{R}$. The attractiveness of the formulation stems from the fact that though the derivatives defined within the framework do not satisfy the Cauchy-Riemann conditions, they obey all the rules of calculus, including the chain rule, differentiation of products and quotients. Thus all computations in the derivation of an algorithm can be carried out as in the real case. We provide the connections between the gradient and Hessian formulations given in [9] described in \mathbb{C}^{2N} and \mathbb{R}^{2N} to the complex \mathbb{C}^N -dimensional space, and establish the basic relationships for optimization in the complex domain including first- and second-order Taylor-series expansions.

Three specific examples are given to demonstrate the application of the framework to complex-valued adaptive signal processing, and to show how they enable the use of the true processing power of the complex domain. The examples include a multilayer perceptron filter design and the derivation of the gradient update (backpropagation) rule, independent component analysis using maximum likelihood, and the derivation of an efficient second-order learning rule, the conjugate gradient algorithm for the complex domain.

Next section introduces the main tool, Wirtinger calculus for optimization in the complex domain and the key results given in [8, 9], which we use to establish the main theory presented in Section 3. In Section 3, we consider both vector and matrix optimization and establish the equivalences for first- and second-order derivatives for the real and complex case, and provide the fundamental results for \mathbb{C}^N and $\mathbb{C}^{N \times M}$. Section 4 presents the application examples and Section 5 gives a short discussion.

2. COMPUTATION OF GRADIENTS IN THE COMPLEX DOMAIN USING WIRTINGER CALCULUS

The fundamental result for the differentiability of a complex-valued function

$$f(z) = u(x, y) + jv(x, y), \quad (1)$$

where $z = x + jy$, is given by the Cauchy-Riemann equations [1]:

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}, \quad \frac{\partial v}{\partial x} = -\frac{\partial u}{\partial y}, \quad (2)$$

which summarize the conditions for the derivative to assume the same value regardless of the direction of approach when $\Delta z \rightarrow 0$. These conditions, when considered carefully, make it clear that the definition of complex differentiability is quite stringent and imposes a strong structure on $u(x, y)$ and $v(x, y)$, the real and imaginary parts of the function, and consequently on $f(z)$. Also, obviously most cost (objective) functions do not satisfy the Cauchy-Riemann equations as these functions are typically $f : \mathbb{C} \rightarrow \mathbb{R}$ and thus have $v(x, y) = 0$.

An elegant approach due to Wirtinger [7] relaxes this strong requirement for differentiability, and defines a less stringent form for the complex domain. More importantly, it describes how this new definition can be used for defining complex differential operators that allow computation of derivatives in a very straightforward manner in the complex domain, by simply using real differentiation results and procedures.

In the development, the commonly used definition of differentiability that leads to the Cauchy-Riemann equations is identified as *complex differentiability* and functions that satisfy the condition on a specified open set as *complex analytic* (or complex holomorphic). The more flexible form of differentiability is identified as *real differentiability*, and a function is called real differentiable when $u(x, y)$ and $v(x, y)$ are differentiable as functions of real-valued variables x and y . Then, one can write the two real-variables as $x = (z + z^*)/2$ and $y = -j(z - z^*)/2$, and use the chain rule to derive the operators for differentiation given in the theorem below. The key point in the derivation is regarding the two variables z and z^* as independent from each other, which is also the main trick that allows us to make use of the elegance of Wirtinger calculus. Hence, we consider a given function $f : \mathbb{C} \mapsto \mathbb{C}$ as $f : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{C}$ by writing it as $f(z) = f(x, y)$, and make use of the underlying \mathbb{R}^2 structure. The main result in this context is stated by Brandwood as follows [8].

Theorem 1. *Let $f : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{C}$ be a function of real variables x and y such that $g(z, z^*) = f(x, y)$, where $z = x + jy$ and that g is analytic with respect to z^* and z independently. Then,*

(i) *the partial derivatives*

$$\frac{\partial g}{\partial z} = \frac{1}{2} \left(\frac{\partial f}{\partial x} - j \frac{\partial f}{\partial y} \right), \quad \frac{\partial g}{\partial z^*} = \frac{1}{2} \left(\frac{\partial f}{\partial x} + j \frac{\partial f}{\partial y} \right) \quad (3)$$

can be computed by treating z^ as a constant in g and z as a constant, respectively;*

(ii) *a necessary and sufficient condition for f to have a stationary point is that $\partial g / \partial z = 0$. Similarly, $\partial g / \partial z^* = 0$ is also a necessary and sufficient condition.*

Therefore, when evaluating the gradient, we can directly compute the derivatives with respect to the complex

argument, rather than calculating individual real-valued gradients as typically performed in the literature (see, e.g., [2, 6, 12, 13]). The requirement for the analyticity of $g(z, z^*)$ with respect to z and z^* is independently equivalent to the condition on real differentiability of $f(x, y)$ since we can move from one form of the function to the other using the simple linear transformation given above [1, 14]. When $f(z)$ is complex analytic, that is, when the Cauchy-Riemann conditions hold, $g(\cdot)$ becomes a function of only z , and the two derivatives, the one given in the theorem and the traditional one coincide.

The case we are typically interested in the development of signal processing algorithms is given by $f : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ and is a special case of the result stated in the theorem. Hence we can employ the same procedure—taking derivatives independently with respect to z and z^* , in the optimization of a real-valued function as well. In the rest of the paper, we consider such functions as these are the costs used in machine learning, though we identify the deviation, if any, from the general $f : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{C}$ case for completeness.

As a simple example, consider the function $g(z, z^*) = zz^* = |z|^2 = x^2 + y^2 = f(x, y)$. We have $(1/2)(\partial f/\partial x + j(\partial f/\partial y)) = x + jy = z$, which we can also evaluate as $\partial g/\partial z^* = z$, that is, by treating z as a constant in g when calculating the partial derivative.

The complex gradient defined by Brandwood [8] has been extended by van den Bos to define a complex gradient and Hessian in \mathbb{C}^{2N} by defining a mapping

$$\mathbf{z} \in \mathbb{C}^N \mapsto \tilde{\mathbf{z}} = \begin{bmatrix} z_1 \\ z_1^* \\ \vdots \\ z_N \\ z_N^* \end{bmatrix} \in \mathbb{C}^{2N}. \quad (4)$$

Note that the mapping allows a direct extension of Wirtinger's result to the multidimensional space through N mappings of the form $(z_{R,k}, z_{I,k}) \mapsto (z_k, z_k^*)$, where $z = z_R + jz_I$, so that one can make use of Wirtinger derivatives. Since the transformation from \mathbb{R}^2 to \mathbb{C}^2 is a simple linear invertible mapping, one can work in either space, depending on the convenience offered by each. In [9], it is shown that such a transformation allows the definition of a Hessian, hence of a Taylor series expansion very similar to the one in the real case, and the Hessian matrix \mathbf{H} defined in this manner is naturally linked to the complex $\mathbb{C}^{N \times N}$ Hessian \mathbf{G} in that if λ is an eigenvalue of \mathbf{G} , then 2λ is the corresponding eigenvalue of \mathbf{H} . The result implies that the positivity of the eigenvalues as well as the conditioning of the Hessian matrices are shared properties of the two matrices, that is, of the two representations. For example, in [15], this property has been utilized to derive the local stability conditions of the complex-valued maximization of negentropy algorithm for performing independent component analysis. In the next section, we establish the connections of the results of [9] to \mathbb{C}^N for first- and second-order derivatives such that efficient second-order optimization algorithms can be derived by di-

rectly working in the original \mathbb{C}^N space where the problems are typically defined.

3. OPTIMIZATION IN THE COMPLEX DOMAIN

3.1. Vector case

We define $\langle \cdot, \cdot \rangle$ as the scalar inner product between two matrices \mathbf{W} and \mathbf{V} as

$$\langle \mathbf{W}, \mathbf{V} \rangle = \text{Trace}(\mathbf{V}^H \mathbf{W}), \quad (5)$$

so that $\langle \mathbf{W}, \mathbf{W} \rangle = \|\mathbf{W}\|_{\text{Fro}}^2$, where the subscript Fro denotes the Frobenius norm. For vectors, the definition simplifies to $\langle \mathbf{w}, \mathbf{v} \rangle = \mathbf{v}^H \mathbf{w}$.

We define the gradient vector $\nabla_{\mathbf{z}} = [\partial/\partial z_1, \partial/\partial z_2, \dots, \partial/\partial z_N]^T$ for vector $\mathbf{z} = [z_1, z_2, \dots, z_N]^T$ with $z_k = z_{R,k} + jz_{I,k}$ in order to write the first-order Taylor series expansion for a function $g(\mathbf{z}, \mathbf{z}^*) : \mathbb{C}^N \times \mathbb{C}^N \rightarrow \mathbb{R}$,

$$\Delta g = \langle \Delta \mathbf{z}, \nabla_{\mathbf{z}^*} g \rangle + \langle \Delta \mathbf{z}^*, \nabla_{\mathbf{z}} g \rangle = 2\text{Re}\{\langle \Delta \mathbf{z}, \nabla_{\mathbf{z}^*} g \rangle\}, \quad (6)$$

where the last equality follows because $g(\cdot, \cdot)$ is real valued. Using the Cauchy-Schwarz-Bunyakovski inequality [16], it is straightforward to show that the first-order change in $g(\cdot, \cdot)$ will be maximized when $\Delta \mathbf{z}$ and the gradient $\nabla_{\mathbf{z}^*} g$ are collinear. Hence, it is the gradient with respect to the conjugate of the variable, $\nabla_{\mathbf{z}^*} g$, that defines the direction of the maximum rate of change in $g(\cdot, \cdot)$ with respect to \mathbf{z} , not $\nabla_{\mathbf{z}} g$ as sometimes noted in the literature. Thus the gradient optimization of $g(\cdot, \cdot)$ should use the update

$$\Delta \mathbf{z} = \mathbf{z}_{t+1} - \mathbf{z}_t = -\mu \nabla_{\mathbf{z}^*} g \quad (7)$$

as this form leads to a nonpositive increment given by $\Delta g = -2\mu \|\nabla_{\mathbf{z}^*} g\|^2$, while the update using $\Delta \mathbf{z} = -\mu \nabla_{\mathbf{z}} g$ results in updates $\Delta g = -2\mu \text{Re}\{\langle \nabla_{\mathbf{z}^*} g, \nabla_{\mathbf{z}} g \rangle\}$, which are not guaranteed to be nonpositive.

Based on (6), similar to a scalar function of two real vectors, the second-order Taylor series expansion of $g(\mathbf{z}, \mathbf{z}^*)$ can be written as [17]

$$\begin{aligned} \Delta^2 g = & \frac{1}{2} \left\langle \frac{\partial g}{\partial \mathbf{z} \partial \mathbf{z}^T} \Delta \mathbf{z}, \Delta \mathbf{z}^* \right\rangle + \frac{1}{2} \left\langle \frac{\partial g}{\partial \mathbf{z}^* \partial \mathbf{z}^H} \Delta \mathbf{z}^*, \Delta \mathbf{z} \right\rangle \\ & + \left\langle \frac{\partial g}{\partial \mathbf{z} \partial \mathbf{z}^H} \Delta \mathbf{z}^*, \Delta \mathbf{z}^* \right\rangle. \end{aligned} \quad (8)$$

Next, we derive the same complex gradient update rule using another approach, which provides the connection between the real and complex domains. We first introduce the following fundamental mappings that are similar in nature to those introduced in [9].

Proposition 1. *Given a function $g(\mathbf{z}, \mathbf{z}^*) : \mathbb{C}^N \times \mathbb{C}^N \rightarrow \mathbb{R}$ that is real differentiable and $f : \mathbb{R}^{2N} \rightarrow \mathbb{R}$ such that $g(\mathbf{z}, \mathbf{z}^*) = f(\mathbf{w})$,*

where $\mathbf{z} = [z_1, z_2, \dots, z_N]^T$, $\mathbf{w} = [z_{R,1}, z_{I,1}, z_{R,2}, z_{I,2}, \dots, z_{R,N}, z_{I,N}]^T$, and $z_k = z_{R,k} + jz_{I,k}$, $k \in \{1, 2, \dots, N\}$, then

$$\begin{aligned} \frac{\partial f}{\partial \mathbf{w}} &= \mathbf{U}^H \frac{\partial g}{\partial \bar{\mathbf{z}}^*}, \\ \frac{\partial^2 f}{\partial \mathbf{w} \partial \mathbf{w}^T} &= \mathbf{U}^H \frac{\partial^2 g}{\partial \bar{\mathbf{z}}^* \partial \bar{\mathbf{z}}^T} \mathbf{U}, \end{aligned} \quad (9)$$

where \mathbf{U} is defined by $\bar{\mathbf{z}} \triangleq \begin{bmatrix} \mathbf{z} \\ \mathbf{z}^* \end{bmatrix} = \mathbf{U}\mathbf{w}$ and satisfies $\mathbf{U}^{-1} = (1/2)\mathbf{U}^H$.

Proof. Define a 2×2 matrix \mathbf{J} as

$$\mathbf{J} = \begin{bmatrix} 1 & j \\ 1 & -j \end{bmatrix} \quad (10)$$

and a vector $\tilde{\mathbf{z}} \in \mathbb{C}^{2N}$ as $\tilde{\mathbf{z}} = [z_1, z_1^*, z_2, z_2^*, \dots, z_N, z_N^*]^T$. Then

$$\tilde{\mathbf{z}} = \mathbf{U}'\mathbf{w}, \quad (11)$$

where $\mathbf{U}'_{2N \times 2N} = \text{diag}\{\mathbf{J}, \mathbf{J}, \dots, \mathbf{J}\}$ that satisfies $(\mathbf{U}')^{-1} = (1/2)(\mathbf{U}')^H$ [9]. Next, we can find a permutation matrix \mathbf{P} such that

$$\bar{\mathbf{z}} \triangleq [z_1, z_2, \dots, z_N, z_1^*, z_2^*, \dots, z_N^*]^T = \mathbf{P}\tilde{\mathbf{z}} = \mathbf{P}\mathbf{U}'\mathbf{w} = \mathbf{U}\mathbf{w}, \quad (12)$$

where $\mathbf{U} \triangleq \mathbf{P}\mathbf{U}'$ that satisfies $\mathbf{U}^{-1} = (1/2)\mathbf{U}^H$ since $\mathbf{P}^{-1} = \mathbf{P}^T$. Using the Wirtinger derivatives in (3), we obtain

$$\frac{\partial g}{\partial \bar{\mathbf{z}}} = \frac{1}{2}\mathbf{U}^* \frac{\partial f}{\partial \mathbf{w}}, \quad (13)$$

which establishes the first-order connection between the complex gradient and the real gradient. By applying the two derivatives (3) recursively to obtain the second-order derivative of g , we obtain

$$\begin{aligned} \frac{\partial^2 f}{\partial \mathbf{w} \partial \mathbf{w}^T} &\stackrel{1}{=} (\mathbf{U}')^H \frac{\partial^2 g}{\partial \tilde{\mathbf{z}}^* \partial \tilde{\mathbf{z}}^T} \mathbf{U}' \\ &\stackrel{2}{=} (\mathbf{U}')^H \mathbf{P}^T \frac{\partial^2 g}{\partial \bar{\mathbf{z}}^* \partial \bar{\mathbf{z}}^T} \mathbf{P}\mathbf{U}' = \mathbf{U}^H \frac{\partial^2 g}{\partial \bar{\mathbf{z}}^* \partial \bar{\mathbf{z}}^T} \mathbf{U}. \end{aligned} \quad (14)$$

Equality 1 is already proved in [18]. Equality 2 is obtained by simply rearranging the entries in $\partial^2 g / \partial \tilde{\mathbf{z}}^* \partial \tilde{\mathbf{z}}^T$ to form $\partial^2 g / \partial \bar{\mathbf{z}}^* \partial \bar{\mathbf{z}}^T$. Therefore, the second-order Taylor expansion given in (8) can be rewritten as

$$\Delta g = \Delta \bar{\mathbf{z}}^T \frac{\partial g}{\partial \bar{\mathbf{z}}} + \frac{1}{2} \Delta \bar{\mathbf{z}}^H \frac{\partial^2 g}{\partial \bar{\mathbf{z}}^* \partial \bar{\mathbf{z}}^T} \Delta \bar{\mathbf{z}}, \quad (15)$$

which demonstrates that the $\mathbb{C}^{2N \times 2N}$ Hessian in (15) can be decomposed into three $\mathbb{C}^{N \times N}$ Hessians in (8). \square

The mappings given in Proposition 1 are similar to those defined in [9]. However, the mappings given in [9] include redundancy since they operate in \mathbb{C}^{2N} and the dimension cannot be further reduced. This is not convenient since cost

function $g(\mathbf{z})$ is normally defined in \mathbb{C}^N and the \mathbb{C}^{2N} mapping as described by $\tilde{\mathbf{z}}$ cannot be always easily applied to define $g(\tilde{\mathbf{z}})$, as observed in [18].

In the following two propositions, we show how to use the same mappings we defined above to obtain first- and second-order derivatives, and hence algorithms, in \mathbb{C}^N in an efficient manner.

Proposition 2. Given functions g and f defined as in Proposition 1, one has the complex gradient update rule

$$\Delta \mathbf{z} = -2\mu \frac{\partial g}{\partial \bar{\mathbf{z}}^*}, \quad (16)$$

which is equivalent to the real gradient update rule

$$\Delta \mathbf{w} = -\mu \frac{\partial f}{\partial \mathbf{w}}, \quad (17)$$

where \mathbf{z} and \mathbf{w} are as defined in Proposition 1 as well.

Proof. Assuming f is known, the gradient update rule in the real domain is

$$\Delta \mathbf{w} = -\mu \frac{\partial f}{\partial \mathbf{w}}. \quad (18)$$

Mapping back into complex domain, we obtain

$$\Delta \bar{\mathbf{z}} = \mathbf{U} \Delta \mathbf{w} = -\mu \mathbf{U} \frac{\partial f}{\partial \mathbf{w}} = -2\mu \frac{\partial g}{\partial \bar{\mathbf{z}}^*}. \quad (19)$$

The dimension of the update rule can be further decreased as

$$\begin{bmatrix} \Delta \mathbf{z} \\ \Delta \mathbf{z}^* \end{bmatrix} = -2\mu \begin{bmatrix} \frac{\partial g}{\partial \bar{\mathbf{z}}^*} \\ \frac{\partial g}{\partial \bar{\mathbf{z}}} \end{bmatrix} \Rightarrow \Delta \mathbf{z} = -2\mu \frac{\partial g}{\partial \bar{\mathbf{z}}^*}. \quad (20)$$

\square

Proposition 3. Given functions g and f defined as in Proposition 1, one has the complex Newton update rule

$$\Delta \mathbf{z} = -(\mathbf{H}_2^* - \mathbf{H}_1^* \mathbf{H}_2^{-1} \mathbf{H}_1)^{-1} \left(\frac{\partial g}{\partial \bar{\mathbf{z}}^*} - \mathbf{H}_1^* \mathbf{H}_2^{-1} \frac{\partial g}{\partial \bar{\mathbf{z}}} \right), \quad (21)$$

which is equivalent to the real Newton update rule

$$\frac{\partial^2 f}{\partial \mathbf{w} \partial \mathbf{w}^T} \Delta \mathbf{w} = -\frac{\partial f}{\partial \mathbf{w}}, \quad (22)$$

where

$$\mathbf{H}_1 = \frac{\partial^2 g}{\partial \mathbf{z} \partial \mathbf{z}^T}, \quad \mathbf{H}_2 = \frac{\partial^2 g}{\partial \mathbf{z} \partial \mathbf{z}^H}. \quad (23)$$

Proof. The pure Newton method in the real domain takes the form given in (22). Using the equalities given in Proposition 1, it can be easily shown that the Newton update in (22) is equivalent to

$$\frac{\partial^2 g}{\partial \bar{\mathbf{z}}^* \partial \bar{\mathbf{z}}^T} \Delta \bar{\mathbf{z}} = -\frac{\partial g}{\partial \bar{\mathbf{z}}^*}. \quad (24)$$

Using the definitions for \mathbf{H}_1 and \mathbf{H}_2 given in (23), we can rewrite (24) as

$$\begin{bmatrix} \mathbf{H}_2^* & \mathbf{H}_1^* \\ \mathbf{H}_1 & \mathbf{H}_2 \end{bmatrix} \begin{bmatrix} \Delta \mathbf{z} \\ \Delta \mathbf{z}^* \end{bmatrix} = - \begin{bmatrix} \frac{\partial g}{\partial \mathbf{z}^*} \\ \frac{\partial g}{\partial \mathbf{z}} \end{bmatrix}. \quad (25)$$

If $\partial^2 g / \partial \bar{\mathbf{z}}^* \partial \bar{\mathbf{z}}^T$ is positive definite, we have

$$\begin{bmatrix} \Delta \mathbf{z} \\ \Delta \mathbf{z}^* \end{bmatrix} = - \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} \begin{bmatrix} \frac{\partial g}{\partial \mathbf{z}^*} \\ \frac{\partial g}{\partial \mathbf{z}} \end{bmatrix}, \quad (26)$$

where

$$\begin{aligned} \mathbf{M}_{11} &= (\mathbf{H}_2^* - \mathbf{H}_1^* \mathbf{H}_2^{-1} \mathbf{H}_1)^{-1}, \\ \mathbf{M}_{12} &= \mathbf{H}_2^{-*} \mathbf{H}_1^* (\mathbf{H}_1 \mathbf{H}_2^{-*} \mathbf{H}_1^* - \mathbf{H}_2)^{-1}, \\ \mathbf{M}_{21} &= (\mathbf{H}_1 \mathbf{H}_2^{-*} \mathbf{H}_1^* - \mathbf{H}_2)^{-1} \mathbf{H}_1 \mathbf{H}_2^{-*}, \\ \mathbf{M}_{22} &= (\mathbf{H}_2 - \mathbf{H}_1 \mathbf{H}_2^{-*} \mathbf{H}_1^*)^{-1}, \end{aligned} \quad (27)$$

and \mathbf{H}_2^{-*} denotes $(\mathbf{H}_2^*)^{-1}$. Since $\partial^2 g / \partial \bar{\mathbf{z}}^* \partial \bar{\mathbf{z}}^T$ is Hermitian, we finally obtain the complex Newton rule as

$$\Delta \mathbf{z} = -(\mathbf{H}_2^* - \mathbf{H}_1^* \mathbf{H}_2^{-1} \mathbf{H}_1)^{-1} \left(\frac{\partial g}{\partial \mathbf{z}^*} - \mathbf{H}_1^* \mathbf{H}_2^{-1} \frac{\partial g}{\partial \mathbf{z}} \right). \quad (28)$$

The expression for $\Delta \mathbf{z}^*$ is the conjugate of (28). \square

3.2. Matrix case

The extension from the vector gradient to matrix gradient is straightforward. For a real-differentiable $g(\mathbf{W}, \mathbf{W}^*) : \mathbb{C}^{N \times N} \times \mathbb{C}^{N \times N} \rightarrow \mathbb{R}$, we can write the first-order expansion as

$$\begin{aligned} \Delta g &= \left\langle \Delta \mathbf{W}, \frac{\partial g}{\partial \mathbf{W}^*} \right\rangle + \left\langle \Delta \mathbf{W}^*, \frac{\partial g}{\partial \mathbf{W}} \right\rangle \\ &= 2\text{Re} \left\{ \left\langle \Delta \mathbf{W}, \frac{\partial g}{\partial \mathbf{W}^*} \right\rangle \right\}, \end{aligned} \quad (29)$$

where $\partial g / \partial \mathbf{W}$ is an $N \times N$ matrix whose (i, j) th entry is the partial derivative of g with respect to w_{ij} . By arranging the matrix gradient into a vector and by using the Cauchy-Schwarz-Bunyakovski inequality [16], it is easy to show that the matrix gradient $\partial g / \partial \mathbf{W}^*$ defines the direction of the maximum rate of change in g with respect to \mathbf{W} .

For local stability analysis, Taylor expansions up to the second order is also frequently needed. Since the first-order matrix gradient takes a matrix form already, here we only provide the second-order expansion with respect to every entry of matrix \mathbf{W} . From (8), we obtain

$$\begin{aligned} \Delta^2 g &= \frac{1}{2} \left(\sum \frac{\partial g}{\partial w_{ij} \partial w_{kl}} dw_{ij} dw_{kl} + \sum \frac{\partial g}{\partial w_{ij}^* \partial w_{kl}^*} dw_{ij}^* dw_{kl}^* \right) \\ &\quad + \sum \frac{\partial g}{\partial w_{ij} \partial w_{kl}^*} dw_{ij} dw_{kl}^*. \end{aligned} \quad (30)$$

We can use the first-order Taylor series expansion to derive the relative gradient [19] update rule for the complex case, which is usually directly extended to the complex case without a derivation [5, 13, 20]. To write the relative gradient rule, we consider an update of the parameter matrix \mathbf{W} in the invariant form $(\Delta \mathbf{W})\mathbf{W}$ [19]. We then write the first-order Taylor series expansion for the perturbation $(\Delta \mathbf{W})\mathbf{W}$ as

$$\begin{aligned} \Delta g &= \left\langle (\Delta \mathbf{W})\mathbf{W}, \frac{\partial g}{\partial \mathbf{W}^*} \right\rangle + \left\langle (\Delta \mathbf{W}^*)\mathbf{W}^*, \frac{\partial g}{\partial \mathbf{W}} \right\rangle \\ &= 2\text{Re} \left\{ \left\langle \Delta \mathbf{W}, \frac{\partial g}{\partial \mathbf{W}^*} \mathbf{W}^H \right\rangle \right\} \end{aligned} \quad (31)$$

to determine the quantity that maximizes the rate of change in the function. The complex relative gradient of g at \mathbf{W} is then written as $(\partial g / \partial \mathbf{W}^*)\mathbf{W}^H$ to write the relative gradient update term as

$$\Delta \mathbf{W} = -\mu \frac{\partial g}{\partial \mathbf{W}^*} \mathbf{W}^H \mathbf{W}. \quad (32)$$

Upon substitution of $\Delta \mathbf{W}$ into (29), we observe that $\Delta g = -2\mu \|(\partial g / \partial \mathbf{W}^*)\mathbf{W}^H\|_{\text{Fro}}^2$ is a nonpositive quantity, thus a proper update term. The relative gradient can be regarded as a special case of natural gradient [21] in the matrix space, but provides the additional advantage that it can be easily extended to nonsquare matrices. In Section 4.2, we show how the relative gradient update rule for independent component analysis based on maximum likelihood can be derived in a very straightforward manner in the complex domain using (32) and Wirtinger calculus.

4. APPLICATION EXAMPLES

We demonstrate the application of the optimization framework introduced in Section 3 by three examples. The first two examples demonstrate the derivation of the update rules for complex-valued nonlinear signal processing. In the third example, we show how the relationship for Newton updates given by Proposition 3 can be utilized to derive efficient update rules such as the conjugate gradient algorithm for the complex domain.

4.1. Fully complex MLP for nonlinear adaptive filtering

The multilayer perceptron filter—or network—provides a good example case for the difficulties that arise in complex-valued processing as discussed in the introduction. These are due to the selection of activation functions for use in the filter structure and the optimization procedure for deriving the weight update rule.

The first issue is due to the conflict between the boundedness and differentiability of functions in the complex domain. This result is stated by Liouville's theorem as: *a bounded entire function must be a constant in the complex domain* [1], where entire refers to differentiability *everywhere*. For example the sigmoid nonlinearity, which has been the most typically used activation function for real-valued MLPs,

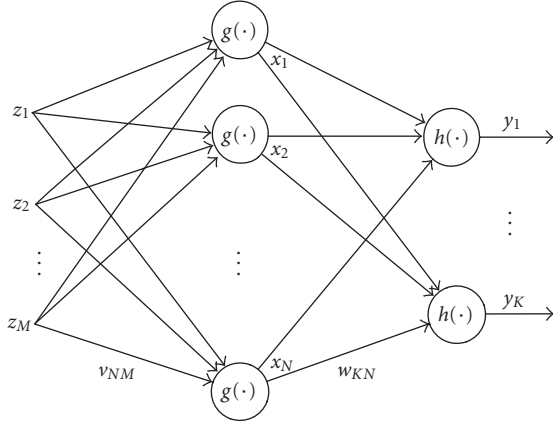


FIGURE 1: A single hidden layer MLP filter.

has periodic singular points. Since boundedness is deemed as important for the stability of algorithms, a practical solution when designing MLPs for the complex domain has been to define nonlinear functions that process the real and imaginary parts separately through bounded real-valued nonlinearities as in [2]

$$f(z) \triangleq \tilde{f}(x) + j\tilde{f}(y) \quad (33)$$

for a complex variable $z = x + jy$ using functions $\tilde{f} : \mathbb{R} \mapsto \mathbb{R}$. Another approach has been to define joint-nonlinear complex activation functions as in [3, 4], respectively,

$$f(z) \triangleq \frac{z}{c + |z|/d}, \quad f(re^{j\theta}) \triangleq \tanh\left(\frac{r}{m}\right)e^{j\theta}. \quad (34)$$

As shown in [10], these functions cannot utilize the phase information effectively, and in applications that introduce significant phase distortion such as equalization of saturating-type channels, are not effective as complex domain nonlinear filters.

The second issue that arises when designing MLPs in the complex domain has to do with the optimization of the chosen cost function to derive the parameter update rule. As an example, consider the most commonly used MLP structure with a single hidden layer as shown in Figure 1. If the cost function is chosen as the squared error at the output, we have

$$J(\mathbf{V}, \mathbf{W}) = \sum_k (d_k - y_k)(d_k^* - y_k^*), \quad (35)$$

where $y_k = h(\sum_n w_{kn}x_n)$ and $x_n = g(\sum_m v_{nm}z_m)$. Note that if both activation functions $h(\cdot)$ and $g(\cdot)$ satisfy the property $[f(z)]^* = f(z^*)$, then the cost function assumes the form $J(\mathbf{V}, \mathbf{W}) = G(z)G(z^*)$ making it clear how practical the derivation of the update rule will be using Wirtinger calculus, since then we treat the two variables z and z^* as independent in the computation of the derivatives. On the other hand, when any of the activation functions given in (33) and (34) are used, it is clear that the evaluation of the gradients will have to be performed through separate real and imaginary part evaluations as traditionally done, which can easily get quite cumbersome [2, 10].

Any function $f(z)$ that is analytic for $|z| < R$ with a Taylor series expansion with all real coefficients in $|z| < R$ satisfies the property $[f(z)]^* = f(z^*)$. Examples of such functions include polynomials and most trigonometric functions and their hyperbolic counterparts. In particular, all the elementary transcendental functions proposed in [12] satisfy the property and can be used as effective activation functions. These functions, though unbounded, provide significant performance advantages in challenging signal processing problems such as equalization of highly nonlinear channels [10] in terms of superior convergence characteristics and better generalization abilities through the efficient representation of the underlying problem structure. The nonsingularities do not pose any practical problems in the implementation, except that some care is required in the selection of their parameters when training these networks. Motivated by these examples, a fundamental result for complex nonlinear approximation is given in [12], where the result on the approximation ability of the multilayer perceptron is extended to the complex domain by classifying nonlinear functions based on their singularities. To establish the universal approximation property in the complex domain, a number of elementary transcendental functions are first classified according to the nature of their nonsingularity as those with removable, isolated, and essential singularities. Based on this classification, three types of approximation theorems are given. The approximation theorems for the first two classes of functions are very general and resemble the universal approximation theorem for the real-valued feed-forward multilayer perceptron that was shown almost concurrently by multiple authors in 1989 [22–24]. The third approximation theorem for the complex multilayer perceptron is unique and related to the power series approximation that can represent any complex number arbitrarily closely in the deleted neighborhood of a singularity. This approximation is uniform only in the analytic domain of convergence whose radius is defined by the closest singularity.

For the MLP filter shown in Figure 1, where y_k is the output and z_m the input, when the activation functions $g(\cdot)$ and $h(\cdot)$ are chosen as functions that are $\mathbb{C} \mapsto \mathbb{C}$ as in [11, 12], we can directly write the backpropagation update equations using Wirtinger derivatives.

For the output units, we have $\partial y_k / \partial w_{kn}^* = 0$, therefore

$$\begin{aligned} \frac{\partial J}{\partial w_{kn}^*} &= \frac{\partial J}{\partial y_k^*} \frac{\partial y_k^*}{\partial w_{kn}^*} \\ &= \frac{\partial [(d_k - y_k)(d_k^* - y_k^*)]}{\partial y_k^*} \frac{\partial h(\sum_n w_{kn}^* x_n^*)}{\partial w_{kn}^*} \\ &= -(d_k - y_k) h' \left(\sum_n w_{kn}^* x_n^* \right) x_n^*. \end{aligned} \quad (36)$$

We define $\delta_k = -(d_k - y_k) h'(\sum_n w_{kn}^* x_n^*)$ so that we can write $\partial J / \partial w_{kn}^* = \delta_k x_n^*$.

For the hidden layer or input layer, first we observe the fact that v_{nm} is connected to x_n for all m . Again, we have

$\partial y_k / \partial v_{nm}^* = 0$, $\partial x_n / \partial v_{nm}^* = 0$. Using the chain rule once again, we obtain

$$\begin{aligned}
\frac{\partial J}{\partial v_{nm}^*} &= \sum_k \frac{\partial J}{\partial y_k^*} \frac{\partial y_k^*}{\partial x_n^*} \frac{\partial x_n^*}{\partial v_{nm}^*} \\
&= \frac{\partial x_n^*}{\partial v_{nm}^*} \sum_k \frac{\partial J}{\partial y_k^*} \frac{\partial y_k^*}{\partial x_n^*} \\
&= g' \left(\sum_m v_{nm}^* z_m^* \right) z_m^* \sum_k \frac{\partial J}{\partial y_k^*} \frac{\partial y_k^*}{\partial x_n^*} \\
&= g' \left(\sum_m v_{nm}^* z_m^* \right) z_m^* \left(\sum_k - (d_k - y_k) h' \left(\sum_l w_{kl}^* x_l^* \right) w_{kn}^* \right) \\
&= z_m^* g' \left(\sum_m v_{nm}^* z_m^* \right) \left(\sum_k \delta_k w_{kn}^* \right).
\end{aligned} \tag{37}$$

Thus, (36) and (37) define the gradient updates for computing the hidden and the output layer coefficients, w_{kn} and v_{nm} , through backpropagation. Note that the derivations in this case are very similar to the real-valued case as opposed to what is shown in [2, 10] where separate evaluations with respect to the real and imaginary parts are carried out.

4.2. Complex maximum likelihood approach to independent component analysis

Independent component analysis (ICA) for separating complex-valued signals is needed in a number of applications such as medical image analysis, radar, and communications. In ICA, the observed data are typically expressed as a linear combination of independent latent variables such that $\mathbf{x} = \mathbf{A}\mathbf{s}$ where $\mathbf{s} = [s_1, s_2, \dots, s_N]^T$ is the vector of sources, $\mathbf{x} = [x_1, x_2, \dots, x_N]^T$ is the vector of observed random variables, and \mathbf{A} is the mixing matrix. We consider the simple case where the number of independent variables is the same as the number of observed mixtures. The main task of the ICA problem is to estimate a separating matrix \mathbf{W} that yields the independent components through $\hat{\mathbf{s}} = \mathbf{W}\mathbf{x}$. Nonlinear ICA approaches such as the maximum likelihood provide practical and efficient solutions to the problem. When deriving the update rule in the complex domain, however, the optimization is not straightforward and can easily become cumbersome [13, 25]. To alleviate the problem, the relative gradient framework of [19] has been used along with isomorphic transformations $\mathbb{C}^N \mapsto \mathbb{R}^{2N}$ to derive the update equations in [25]. As we show next, Wirtinger calculus allows a much more straightforward derivation procedure, and in addition, provides a convenient formulation for working with probabilistic descriptions such as the probability density function (pdf) in the complex domain.

We define the pdf of a complex random variable $X = X_R + jX_I$ as $p_X(x) \equiv p_{X_R X_I}(x_R, x_I)$ and the expectation of $g(X)$ is given by $E\{g(X)\} = \iint g(x_R + jx_I) p_X(x) dx_R dx_I$ for any measurable function $g : \mathbb{C} \rightarrow \mathbb{C}$. The traditional ICA problem determines a weight matrix \mathbf{W} such that $\mathbf{y} = \mathbf{W}\mathbf{x}$

approximates the source \mathbf{s} subject to the permutation and scaling ambiguity. To write the density transformation, we consider the mapping $\mathbb{C} \rightarrow \mathbb{R}^{2N}$ such that $\bar{\mathbf{y}} = \mathbf{W}\bar{\mathbf{x}} = \bar{\mathbf{s}}$, where $\bar{\mathbf{y}} = [\mathbf{y}_R^T \mathbf{y}_I^T]^T$, $\bar{\mathbf{W}} = \begin{bmatrix} \mathbf{W}_R & -\mathbf{W}_I \\ \mathbf{W}_I & \mathbf{W}_R \end{bmatrix}$, $\bar{\mathbf{x}} = [\mathbf{x}_R^T \mathbf{x}_I^T]^T$, and $\bar{\mathbf{s}} = [\mathbf{s}_R^T \mathbf{s}_I^T]^T$.

Given T independent samples $\mathbf{x}(t)$, we write the log-likelihood function as [26]

$$l'(\mathbf{y}, \mathbf{W}) = \log |\det(\bar{\mathbf{W}})| + \sum_{k=1}^N \log p_k(y_k), \tag{38}$$

where p_k is the density function for k th source. Maximization of l' is equivalent to minimization of l where $l = -l'$. Simple algebraic and differential calculus yields

$$dl = -\text{tr}(d\bar{\mathbf{W}}\bar{\mathbf{W}}^{-1}) + \bar{\psi}^T(\bar{\mathbf{y}})d\bar{\mathbf{y}}, \tag{39}$$

where $\bar{\psi}(\bar{\mathbf{y}})$ is a $2N \times 1$ column vector with components

$$\begin{aligned}
\bar{\psi}(\bar{\mathbf{y}}) &= - \begin{bmatrix} \frac{\partial \log p_1(y_1)}{\partial y_{R,1}} & \dots & \frac{\partial \log p_N(y_N)}{\partial y_{R,N}} & \frac{\partial \log p_1(y_1)}{\partial y_{I,1}} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \frac{\partial \log p_N(y_N)}{\partial y_{I,N}} & \dots \end{bmatrix}.
\end{aligned} \tag{40}$$

We write $\log p_s(y_R, y_I) = \log p_s(y, y^*)$ and using Wirtinger calculus, it is straightforward to show

$$\bar{\psi}^T(\bar{\mathbf{y}})d\bar{\mathbf{y}} = \psi^T(\mathbf{y}, \mathbf{y}^*)d\mathbf{y} + \psi^H(\mathbf{y}, \mathbf{y}^*)d\mathbf{y}^*, \tag{41}$$

where $\psi(\mathbf{y}, \mathbf{y}^*)$ is an $N \times 1$ column vector with complex components

$$\psi_k(y_k, y_k^*) = - \frac{\partial \log p_k(y_k, y_k^*)}{\partial y_k}. \tag{42}$$

Defining a $2N \times 2N$ matrix $\mathbf{P} = (1/2) \begin{bmatrix} \mathbf{I} & j\mathbf{I} \\ j\mathbf{I} & \mathbf{I} \end{bmatrix}$, we obtain

$$\begin{aligned}
\text{tr}(d\bar{\mathbf{W}}\bar{\mathbf{W}}^{-1}) &= \text{tr}(d\bar{\mathbf{W}}\mathbf{P}\mathbf{P}^{-1}\bar{\mathbf{W}}^{-1}) \\
&= \text{tr} \left\{ \begin{bmatrix} d\mathbf{W}^* & jd\mathbf{W} \\ jd\mathbf{W}^* & d\mathbf{W} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{W}^* & j\mathbf{W} \\ j\mathbf{W}^* & \mathbf{W} \end{bmatrix}^{-1} \right\} \\
&= \text{tr}(d\mathbf{W}\mathbf{W}^{-1}) + \text{tr}(d\mathbf{W}^*\mathbf{W}^{-*}).
\end{aligned} \tag{43}$$

Therefore, we can write (39) as

$$\begin{aligned}
dl &= -\text{tr}(d\mathbf{W}\mathbf{W}^{-1}) - \text{tr}(d\mathbf{W}^*\mathbf{W}^{-*}) \\
&\quad + \psi^T(\mathbf{y}, \mathbf{y}^*)d\mathbf{y} + \psi^H(\mathbf{y}, \mathbf{y}^*)d\mathbf{y}^*.
\end{aligned} \tag{44}$$

Using $\mathbf{y} = \mathbf{W}\mathbf{x}$ and defining $d\mathbf{Z} = (d\mathbf{W})\mathbf{W}^{-1}$, we obtain

$$\begin{aligned}
d\mathbf{y} &= (d\mathbf{W})\mathbf{x} = d\mathbf{W}(\mathbf{W}^{-1})\mathbf{y} = d\mathbf{Z}\mathbf{y}, \\
d\mathbf{y}^* &= d\mathbf{Z}^*\mathbf{y}^*.
\end{aligned} \tag{45}$$

By treating \mathbf{W} as a constant matrix, the differential matrix $d\mathbf{Z}$ has components dz_{ij} that are linear combinations of dw_{ij}

and is a nonintegrable differential form. However, this transformation greatly simplifies the expression for the Taylor series expansion without changing the function value. It also provides an elegant approach for the derivation of the natural gradient update for maximum likelihood ICA [26]. Using this transformation, we can write (44) as

$$dl = -\text{tr}(d\mathbf{Z}) - \text{tr}(d\mathbf{Z}^*) + \psi^T(\mathbf{y}, \mathbf{y}^*) d\mathbf{Z}\mathbf{y} + \psi^H(\mathbf{y}, \mathbf{y}^*) d\mathbf{Z}^* \mathbf{y}^*. \quad (46)$$

Therefore, the gradient update rule for \mathbf{Z} is given by

$$\Delta\mathbf{Z} = -\mu \frac{\partial l}{\partial \mathbf{Z}^*} = \mu[\mathbf{I} - \psi^*(\mathbf{y}, \mathbf{y}^*) \mathbf{y}^H], \quad (47)$$

which is equivalent to

$$\Delta\mathbf{W} = \mu[\mathbf{I} - \psi^*(\mathbf{y}, \mathbf{y}^*) \mathbf{y}^H] \mathbf{W} \quad (48)$$

by using $d\mathbf{Z} = (d\mathbf{W})\mathbf{W}^{-1}$.

Thus the complex score function is defined as $\psi^*(\mathbf{y}, \mathbf{y}^*)$, as in [27], which takes a form very similar to the real case [26], but with the difference that in the complex case the entries in the score function are defined using Wirtinger derivatives.

4.3. Complex conjugate gradient (CG) algorithm

The equivalence condition given by Proposition 3 allows for easy derivation of second-order efficient update schemes as we demonstrate next. As shown in Proposition 3, for a real differentiable function $g(\mathbf{z}, \mathbf{z}^*) : \mathbb{C}^N \times \mathbb{C}^N \rightarrow \mathbb{R}$ and $f : \mathbb{R}^{2N} \rightarrow \mathbb{R}$ such that $g(\mathbf{z}, \mathbf{z}^*) = f(\mathbf{w})$, the update for the Newton method in \mathbb{R}^{2N} is given by

$$\frac{\partial^2 f}{\partial \mathbf{w} \partial \mathbf{w}^T} \Delta \mathbf{w} = -\frac{\partial f}{\partial \mathbf{w}}, \quad (49)$$

and is equivalent to

$$\Delta \mathbf{z} = -(\mathbf{H}_2^* - \mathbf{H}_1^* \mathbf{H}_2^{-1} \mathbf{H}_1)^{-1} \left(\frac{\partial g}{\partial \mathbf{z}^*} - \mathbf{H}_1^* \mathbf{H}_2^{-1} \frac{\partial g}{\partial \mathbf{z}} \right) \quad (50)$$

in \mathbb{C}^N . To achieve convergence, we require that the search direction $\Delta \mathbf{w}$ is a descent direction when minimizing a cost function, which is the case if the Hessian $\partial^2 f / \partial \mathbf{w} \partial \mathbf{w}^T$ is positive definite. However, if the Hessian is not positive definite, $\Delta \mathbf{w}$ may be an ascent direction. The line search Newton-CG method is one of the strategies for ensuring that the update is of good quality. In this strategy, we solve (49) using the CG method, terminating the updates if $\Delta \mathbf{w}^T (\partial^2 f / \partial \mathbf{w} \partial \mathbf{w}^T) \Delta \mathbf{w} \leq 0$.

When we do not have the definition of function f but only have the knowledge of g , we can obtain the complex conjugate gradient method with straightforward algebraic manipulations of the real CG algorithm (e.g., given in [28]) by using the three equalities given in (12), (13), and (14). We let $\mathbf{s} = \partial g / \partial \mathbf{z}^*$ to write the complex CG method as shown in Algorithm 1, and the complex line search Newton-CG algorithm is given in Algorithm 2.

The complex Wolfe condition [28] can be easily obtained from the real Wolfe condition using a procedure similar to

```

Given some initial gradient  $\mathbf{s}_0$ ;
Set  $\mathbf{x}_0 = \mathbf{0}$ ,  $\mathbf{p}_0 = -\mathbf{s}_0$ ,  $k = 0$ ;
while  $|\mathbf{s}_k| \neq 0$ 
     $\alpha_k = \frac{\mathbf{s}_k^H \mathbf{s}_k}{\text{Re}(\mathbf{p}_k^T \mathbf{H}_2 \mathbf{p}_k^* + \mathbf{p}_k^T \mathbf{H}_1 \mathbf{p}_k)}$ ;
     $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$ ;
     $\mathbf{s}_{k+1} = \mathbf{s}_k + \alpha_k (\mathbf{H}_2^* \mathbf{p}_k + \mathbf{H}_1^* \mathbf{p}_k^*)$ ;
     $\beta_{k+1} = \frac{\mathbf{s}_{k+1}^H \mathbf{s}_{k+1}}{\mathbf{s}_k^H \mathbf{s}_k}$ ;
     $\mathbf{p}_{k+1} = -\mathbf{s}_{k+1} + \beta_{k+1} \mathbf{p}_k$ ;
     $k = k + 1$ ;
end(while)

```

ALGORITHM 1: Complex conjugate gradient algorithm.

```

for  $k = 0, 1, 2, \dots$ 
    Compute a search direction  $\Delta \mathbf{z}$  by applying the complex
    CG method, starting from  $\mathbf{x}_0 = \mathbf{0}$ .
    Terminating when  $\text{Re}(\mathbf{p}_k^T \mathbf{H}_2 \mathbf{p}_k^* + \mathbf{p}_k^T \mathbf{H}_1 \mathbf{p}_k) \leq 0$ ;
    Set  $\mathbf{z}_{k+1} = \mathbf{z}_k + \mu \Delta \mathbf{z}$ , where  $\mu$  satisfies a complex Wolfe
    condition.
end

```

ALGORITHM 2: Complex line search Newton-CG algorithm.

the one followed in Proposition 3. It should be noted that the complex conjugate gradient algorithm is a linear version such that the solution of a linear equation is considered. The procedure given in [28] can be used to obtain the version for a given nonlinear function.

5. DISCUSSION

We describe a framework for complex-valued adaptive signal processing based on Wirtinger calculus for the efficient computation of algorithms and their analyses. By enabling to work directly in the complex domain without the need to increase the problem dimensionality, the framework facilitates the derivation of update rules and makes efficient second-order update procedures such as the conjugate-gradient rule readily available for complex optimization. The examples we have provided demonstrate the simplicity offered by the approach in the derivation of both componentwise update rules as in the case of the backpropagation algorithm for the MLP and direct matrix updates for estimating the demixing matrix as in the case of independent component analysis using maximum likelihood. The framework can also be used to perform the analysis of nonlinear adaptive algorithms such as ICA using the relative gradient update given in (48) as shown in [29] in the derivation of local stability conditions.

ACKNOWLEDGMENT

This work is supported by the National Science Foundation through Grants NSF-CCF 0635129 and NSF-IIS 0612076.

REFERENCES

- [1] R. Remmert, *Theory of Complex Functions*, Springer, New York, NY, USA, 1991.
- [2] H. Leung and S. Haykin, "The complex backpropagation algorithm," *IEEE Transactions on Signal Processing*, vol. 39, no. 9, pp. 2101–2104, 1991.
- [3] G. M. Georgiou and C. Koutsougeras, "Complex backpropagation," *IEEE Transactions on Circuits Systems*, vol. 39, no. 5, pp. 330–334, 1992.
- [4] A. Hirose, "Continuous complex-valued backpropagation learning," *Electronics Letters*, vol. 28, no. 20, pp. 1854–1855, 1992.
- [5] J. Anemüller, T. J. Sejnowski, and S. Makeig, "Complex independent component analysis of frequency-domain electroencephalographic data," *Neural Networks*, vol. 16, no. 9, pp. 1311–1323, 2003.
- [6] P. Smaragdis, "Blind separation of convolved mixtures in the frequency domain," *Neurocomputing*, vol. 22, no. 1–3, pp. 21–34, 1998.
- [7] W. Wirtinger, "Zur formalen theorie der funktionen von mehr komplexen veränderlichen," *Mathematische Annalen*, vol. 97, no. 1, pp. 357–375, 1927.
- [8] D. H. Brandwood, "A complex gradient operator and its application in adaptive array theory," *IEE Proceedings, F: Communications, Radar and Signal Processing*, vol. 130, no. 1, pp. 11–16, 1983.
- [9] A. van den Bos, "Complex gradient and Hessian," *IEE Proceedings: Vision, Image and Signal Processing*, vol. 141, no. 6, pp. 380–382, 1994.
- [10] T. Kim and T. Adalı, "Fully complex multi-layer perceptron network for nonlinear signal processing," *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, vol. 32, no. 1–2, pp. 29–43, 2002.
- [11] A. I. Hanna and D. P. Mandic, "A fully adaptive normalized nonlinear gradient descent algorithm for complex-valued nonlinear adaptive filters," *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2540–2549, 2003.
- [12] T. Kim and T. Adalı, "Approximation by fully complex multilayer perceptrons," *Neural Computation*, vol. 15, no. 7, pp. 1641–1666, 2003.
- [13] J. Eriksson, A. Seppola, and V. Koivunen, "Complex ICA for circular and non-circular sources," in *Proceedings of the 13th European Signal Processing Conference (EUSIPCO '05)*, Antalya, Turkey, September 2005.
- [14] K. Kreutz-Delgado, "Lecture supplement on complex vector calculus," *Course notes for ECE275A: Parameter Estimation I*, 2006.
- [15] M. Novey and T. Adalı, "Stability analysis of complex-valued nonlinearities for maximization of nongaussianity," in *Proceedings of the 31th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '06)*, vol. 5, pp. 633–636, Toulouse, France, May 2006.
- [16] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia, Pa, USA, 2000.
- [17] T. J. Abatzoglou, J. M. Mendel, and G. A. Harada, "The constrained total least squares technique and its applications to harmonic superresolution," *IEEE Transactions on Signal Processing*, vol. 39, no. 5, pp. 1070–1087, 1991.
- [18] A. van den Bos, "Estimation of complex parameters," in *Proceedings of the 10th IFAC Symposium on System Identification (SYSID '94)*, vol. 3, pp. 495–499, Copenhagen, Denmark, July 1994.
- [19] J.-F. Cardoso and B. H. Laheld, "Equivariant adaptive source separation," *IEEE Transactions on Signal Processing*, vol. 44, no. 12, pp. 3017–3030, 1996.
- [20] V. Calhoun and T. Adalı, "Complex ICA for FMRI analysis: performance of several approaches," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03)*, vol. 2, pp. 717–720, Hong Kong, April 2003.
- [21] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, no. 2, pp. 251–276, 1998.
- [22] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, no. 4, pp. 303–314, 1989.
- [23] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feed-forward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [24] K. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Networks*, vol. 2, no. 3, pp. 182–192, 1989.
- [25] J.-F. Cardoso and T. Adalı, "The maximum likelihood approach to complex ICA," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '06)*, vol. 5, pp. 673–676, Toulouse, France, May 2006.
- [26] S.-I. Amari, T.-P. Chen, and A. Cichocki, "Stability analysis of learning algorithms for blind source separation," *Neural Networks*, vol. 10, no. 8, pp. 1345–1351, 1997.
- [27] T. Adalı and H. Li, "A practical formulation for computation of complex gradients and its application to maximum likelihood ICA," in *Proceedings of IEEE the International Conference on Acoustics, Speech and Signal Processing (ICASSP '07)*, vol. 2, pp. 633–636, Honolulu, Hawaii, USA, 2007.
- [28] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, NY, USA, 2000.
- [29] H. Li and T. Adalı, "Stability analysis of complex maximum likelihood ICA using Wirtinger calculus," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '08)*, Las Vegas, Nev, USA, April 2008.