1

COMPLEX-VALUED ADAPTIVE SIGNAL PROCESSING

Tülay Adalı and Hualiang Li
University of Maryland Baltimore County, Baltimore, MD

1.1 INTRODUCTION

Complex-valued signals arise frequently in applications as diverse as communications, radar, and biomedicine, as most practical modulation formats are of complex type and applications such as radar and magnetic resonance imaging (MRI) lead to data that are inherently complex valued. When the processing has to be done in a transform domain such as Fourier or complex wavelet, again the data are complex valued. The complex domain not only provides a convenient representation for these signals but also a natural way to preserve the physical characteristics of the signals and the transformations they go through, such as the phase and magnitude distortion a communications signal experiences. In all these cases, the processing also needs to be carried out in the complex domain in such a way that the complete information—represented by the interrelationship of the real and imaginary parts or the magnitude and phase of the signal—can be fully exploited.

In this chapter, we introduce a framework based on Wirtinger calculus that enables working completely in the complex domain for the derivation and analysis of signal processing algorithms, and in such a way that all of the computations can be performed in a straightforward manner, very similarly to the real-valued case. In the derivation of

Adaptive Signal Processing: Next Generation Solutions. Edited by Tülay Adalı and Simon Haykin

adaptive algorithms, we need to evaluate the derivative of a cost function. Since the cost functions are real valued, hence not *differentiable* in the complex domain, traditionally we evaluate derivatives separately for the real and imaginary parts of the function and then combine them to form the derivative. We show that using Wirtinger calculus, we can directly evaluate the derivatives without the need to evaluate the real and imaginary parts separately. Beyond offering simple convenience, this approach makes many signal processing tools developed for the real-valued domain readily available for complex-valued signal processing as the evaluations become very similar to the real-valued case and most results from real-valued calculus do hold and can be directly used. In addition, by keeping the expressions simple, the approach eliminates the need to make simplifying assumptions in the derivations and analyses that have become common place for many signal processing algorithms derived for the complex domain.

It is important to emphasize that the regularity condition for the applicability of Wirtinger calculus in the evaluations is quite mild, making it a very powerful tool, and also widely applicable. To reiterate the two points we have made regarding the main advantages of the approach, first, algorithm derivation and analysis become much shorter and compact compared to the traditional splitting approach. In this chapter, this advantage is demonstrated in the derivation of update rules for the multilayer perceptron and the widely linear filter, and of algorithms for independent component analysis.

However, the real advantage of the Wirtinger approach is beyond simple convenience in the derivations. Because the traditional splitting approach for the real and imaginary parts leads to long and complicated expressions, especially when working with nonlinear functions and/or second-order derivatives, one is often forced to make certain assumptions to render the evaluations more manageable. One such assumption that is commonly made is the circularity of signals, which limits the usefulness of the solutions developed since many practical signals have noncircular distributions as we discuss in Section 1.2.5. Since with Wirtinger calculus, the expressions are kept simple, we can avoid such and many other simplifying assumptions allowing one to fully exploit the power of complex processing, for example, in the derivation of independent component analysis (ICA) algorithms as discussed in Section 1.6.

Besides developing the main results for the application of Wirtinger calculus, in this chapter, we demonstrate the application of the framework to a number of powerful solutions proposed recently for the complex-valued domain, and emphasize how the Wirtinger framework enables taking full advantage of the power of complex-valued processing and of these solutions in particular. We show that the widely linear filter is to be preferred when the commonly invoked circularity assumptions on the signal do not hold, and that the fully complex nonlinear filter allows efficient use of the available information, and more importantly, show how both solutions can take full advantage of the power of Wirtinger calculus. We also show that the framework enables the development and study of a powerful set of algorithms for independent component analysis of complex-valued data.

1.1.1 Why Complex-Valued Signal Processing?

Complex domain is the natural home for the representation and processing of many signals we encounter in practice. There are four main scenarios in which complex processing is needed.

- The signal can be natively complex, where an in-phase and a quadrature component is the natural representation and enables one to fully take the relationship between the two components into account. Examples include radar and MRI signal [2] as well as many communication signals such as those using binary phase shift keying (BPSK), quadrature phase shift keying (QPSK), and quadrature amplitude modulation (QAM) as shown in Figure 1.1. The MRI signal is acquired as a quadrature signal using two orthogonal detectors as shown in Figure 1.2 [17]. Hence, the complex k-space representation is the natural one for the MRI signal, which is typically inverse Fourier-transformed into the complex image space in reconstruction resulting in complex-valued spatial domain signal.
- Harmonic analysis, in particular Fourier analysis, has been one of the most widely used tools in signal processing. More recently, complex wavelet transforms have emerged as attractive tools for signal processing as well, and in all these instances where the processing has to be performed in a transform domain, one needs to perform complex-valued signal processing.
- Analytic representation of a real-valued bandpass signal using its complex envelope is commonly used in signal processing, in particular in communications. The complex envelope representation facilitates the derivation of modulation and demodulation techniques, and the analysis of certain properties of the signal.
- There are also cases where complex domain is used to capture the relationship between the magnitude and phase or two channels of real-valued signals. Examples include wind data where a complex-valued signal is constructed using the strength and direction of wind data [37] and the magnitude of structural MRI data where the white and gray matter are combined to form a complex number to make use of their interdependence in the processing of data [116].

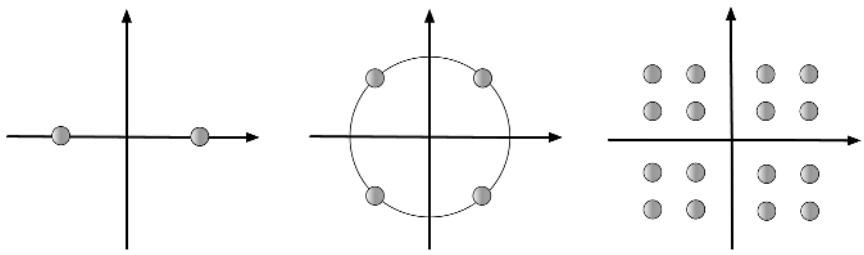

Figure 1.1 Signal constellations for BPSK, QPSK, and QAM signals.

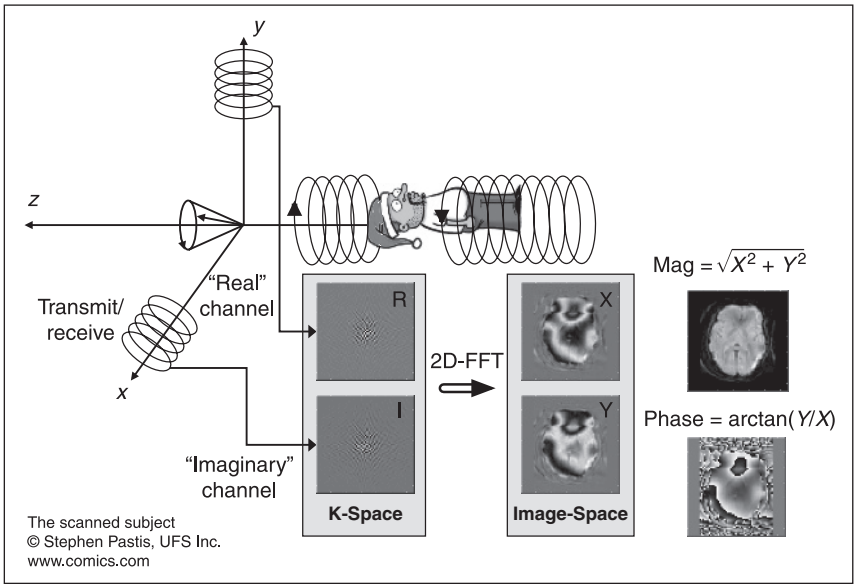

Figure 1.2 MRI signal is acquired as a quadrature signal using two orthogonal detectors, hence is inherently complex.

In all these instances, and in many similar ones, complex domain allows one to fully take advantage of the complete information in the real and imaginary channels of a given signal and thus is the natural home for the development of signal processing algorithms.

In this chapter, our focus is the description of an efficient framework such that all (or most) of the processing can be performed in the complex domain without performing transformations to and from the real domain. This point has long been a topic of debate since equivalent transformations between the two domains can be easily established, and since the real domain is the one with which we are more familiar, the question arises as to why not transform the problem into the real domain and perform all of the evaluations and analyses there. There are a number of reasons for keeping the computations and analysis in the complex domain rather than using complex-to-real transformations.

(1) Most typically, when the signal in question is complex, the cost function is also defined in the complex domain where the signal as well as the transformations the signal goes through are easily represented. It is thus desirable to keep all of the computations in the original domain rather than working with transformations to and from the real-valued domain, that is, transformations of the type: $\mathbb{C}^N \mapsto \mathbb{R}^{2N}$.

(2) Even though real-to-complex transformations are always possible using Jacobians, they are not always very straightforward to obtain, especially when the function is not invertible. In addition, when nonlinear functions are

involved, in order to transform the solution back to the complex domain, we usually have to make additional assumptions such as analyticity of the function. We give a simple example (Example 1.3) to highlight this point in Section 1.2.2.

(3) When working in the real-dimensional space with the double dimension, many quantities assume special forms. Matrices in this space usually have special block structures which can make further analysis and manipulations more complicated. In fact, these structures have been the primary motivation for invoking certain simplifying assumptions in the analysis, such as the circularity of signals. For example, this assumption is made in [13] in the derivation of an independent component analysis algorithm when computing the Hessian primarily for this reason. Circularity, which implies that the phase of the signal is uniformly distributed and hence is noninformative, is in most cases an unrealistic assumption limiting the usefulness of algorithms. The communications signals shown in Figure 1.1 as well as a number of other real-world signals can be shown not to satisfy this property, and are discussed in more detail in Section 1.2.5.

Thus, even though we can define a transformation $\mathbb{C}^N \mapsto \mathbb{R}^{2N}$, which is isomorphic, we have to remember that mathematical equivalence does not imply that the optimization, analysis, and numerical and computational properties of the algorithms will be similar in these two domains. We argue that $\mathbb{C}^N$ defines a much more desirable domain for adaptive signal processing in general and give examples to support our point. Using Wirtinger calculus, most of the processing and analysis in the complex domain can be performed in a manner very similar to the real-valued case as we describe in this chapter, thus eliminating the need to consider such transformations in the first place.

The theory and algorithms using the widely linear and the fully complex filter can be easily developed using Wirtinger calculus. Both of these filters are powerful tools for complex-valued signal processing that allow taking advantage of the full processing power of the complex domain and without having to make limiting assumptions on the nature of the signal, such as circularity.

1.1.2 Outline of the Chapter

To present the development, we first present preliminaries including a review of basic results for derivatives and Taylor series expansions, and introduce the main idea behind Wirtinger calculus that describes an effective approach for complex-valued signal processing. We define first- and second-order Taylor series expansions in the complex domain, establish the key relationships that enable efficient derivation of first- and second-order adaptive algorithms as well as performing analyses such as local stability using a quadratic approximation within a neighborhood of a local optimum. We also provide a review of complex-valued statistics, again a topic that has been, for the most part, treated in a limited form in the literature for complex signals. We carefully define circularity of a signal, the associated properties and complete

statistical characterization of a complex signal, which play an important role in the subsequent discussions on widely linear filters and independent component analysis.

Next, we show how Wirtinger calculus enables derivation of effective algorithms using two filter structures that have been shown to effectively use the complete statistical information in the complex signal and discuss the properties of these filters. These are the *widely linear* and the *fully complex* nonlinear filters, two attractive solutions for the next generation signal processing systems. Even though the widely linear filter is introduced in 1995 [94], its importance in practice has not been noted until recently. Similarly, the idea of fully complex nonlinear filters is not entirely new, but the theory that justifies their use has been developed more recently [63], and both solutions hold much promise for complex-valued signal processing. In Sections 1.4 and 1.5, we present the basic theory of widely linear filters and nonlinear filters—in particular multilayer perceptrons—with fully complex activation functions using Wirtinger calculus. Finally in Section 1.6, we show how Wirtinger calculus together with fully complex nonlinear functions enables derivation of a unified framework for independent component analysis, a statistical analysis tool that has found wide application in many signal processing problems.

1.2 PRELIMINARIES

1.2.1 Notation

A complex number $z \in \mathbb{C}$ is written as $z = z_r + jz_i$ where $j = \sqrt{-1}$ and z_r and z_i refer to the real and imaginary parts. In our discussions, when concentrating on a single variable, we use the notation without subscripts as in $z = x + jy$ to keep the expressions simple. The complex conjugate is written as $z^* = z_r - jz_i$, and vectors are always assumed to be column vectors, hence $\mathbf{z} \in \mathbb{C}^N$ implies $\mathbf{z} \in \mathbb{C}^{N\times 1}$.

In Table 1.1 we show the six types of derivatives of interest that result in matrix forms along with our convention for the form of the resulting expression depending on whether the vector/matrix is in the numerator or the denominator. Our discussions in the chapter will mostly focus on the derivatives given on the top row of the table, that is, functions that are scalar valued. The extension to the other three cases given in

Table 1.1 Functions of interest and their derivatives

	Scalar Variable: $z \in \mathbb{C}$	Vector Variable: $\mathbf{z} \in \mathbb{C}^N$	Matrix Variable: $\mathbf{Z} \in \mathbb{C}^{N\times M}$
Scalar Function: $f \in \mathbb{C}$	$\dfrac{\partial f}{\partial z} \in \mathbb{C}$	$\dfrac{\partial f}{\partial \mathbf{z}} = \left[\dfrac{\partial f}{\partial z_k}\right] \in \mathbb{C}^N$	$\dfrac{\partial f}{\partial \mathbf{Z}} = \left[\dfrac{\partial f}{\partial Z_{kl}}\right] \in \mathbb{C}^{N\times M}$
Vector Function: $\boldsymbol{f} \in \mathbb{C}^L$	$\dfrac{\partial \boldsymbol{f}}{\partial z} \in \mathbb{C}^{1\times L}$	$\dfrac{\partial \boldsymbol{f}}{\partial \mathbf{z}} = \left[\dfrac{\partial f_l}{\partial z_k}\right] \in \mathbb{C}^{N\times L}$	
Matrix Function: $\boldsymbol{F} \in \mathbb{C}^{L\times K}$	$\dfrac{\partial \boldsymbol{F}}{\partial z} \in \mathbb{C}^{K\times L}$		

the table is straightforward. The remaining three cases that are omitted from the table and that do not result in a matrix form can be either handled using the vectorization operator as in [46], or by using suitable definitions of differentials as in [7]. We introduce the vectorization operator in Section 1.2.3 and give an example of the use of the differential definition of [7] in Section 1.6.1 to demonstrate how one can alleviate the need to work with tensor representations.

The matrix notation used in Table 1.1 refers to the elements of the vectors or matrices. For the gradient vector $\nabla_{\mathbf{z}} f$, we have

$$\nabla_{\mathbf{z}} f = \frac{\partial f}{\partial \mathbf{z}} = \begin{bmatrix} \dfrac{\partial f}{\partial z_1} \\ \dfrac{\partial f}{\partial z_2} \\ \vdots \\ \dfrac{\partial f}{\partial z_N} \end{bmatrix}$$

and

$$\nabla_{\mathbf{Z}} f = \frac{\partial f}{\partial \mathbf{Z}} = \begin{bmatrix} \dfrac{\partial f}{\partial Z_{1,1}} & \dfrac{\partial f}{\partial Z_{1,2}} & \cdots & \dfrac{\partial f}{\partial Z_{1,M}} \\ \dfrac{\partial f}{\partial Z_{2,1}} & \dfrac{\partial f}{\partial Z_{2,2}} & \cdots & \dfrac{\partial f}{\partial Z_{2,M}} \\ \vdots & \vdots & \vdots & \vdots \\ \dfrac{\partial f}{\partial Z_{N,1}} & \dfrac{\partial f}{\partial Z_{N,2}} & \cdots & \dfrac{\partial f}{\partial Z_{N,M}} \end{bmatrix},$$

for the matrix gradient $\nabla_{\mathbf{Z}} f$. The $N \times L$ Jacobian matrix

$$J_{\mathbf{z}} f = \frac{\partial f}{\partial \mathbf{z}}$$

is also written similarly.

In the development we present in this chapter, we emphasize the use of derivatives directly in the form given in Table 1.1 rather than splitting the derivatives into real and imaginary parts and evaluating the two separately, which is the procedure most typically used in the literature when evaluating derivatives of nonanalytic functions. Our approach keeps all expressions in the complex domain where they are typically defined, rather than transforming to and from another domain, which typically is the real domain.

As such, when evaluating complex derivatives, all conventions and formulas used in the computation of real-valued derivatives can be directly used for both analytic and nonanalytic functions. A good reference for the computation of real-valued matrix

derivatives is [88]. As we show through a number of examples of interest for adaptive signal processing in Sections 1.4–1.6, these formulas can be used without much alteration for the complex case.

In the development, we use various representations for a given function $f(\cdot)$, that is, write it in terms of different arguments. When doing so, we keep the function variable, which is $f(\cdot)$ in this case, the same. It is important to note, however, that even though these representations are all equivalent, different arguments may result in quite different forms for the function. A simple example is given below.

■ **EXAMPLE 1.1**

For a given function $f(z) = |z|^2$, where $z = x + jy$, we can write

$$f(z, z^*) = zz^*$$

or

$$f(x, y) = x^2 + y^2.$$

It is also important to note that in some cases, explicitly writing the function in one of the two forms given above—as $f(z, z^*)$ or $f(x, y)$—is not possible. A simple example is the magnitude square of a nonlinear function, for example, $f(z) = |\tanh(z)|^2$. In such cases, the advantage of the approach we emphasize in this chapter, that is, directly working in the complex domain, becomes even more evident.

Depending on the application, one might have to work with functions defined to satisfy certain properties such as boundedness. When referring to such functions, that is, those that are defined to satisfy a given property, as well as traditional functions such as trigonometric functions, we use the terminology introduced in [61] to be able to differentiate among those as given in the next definition.

Definition 1 (Split-complex and fully-complex functions) *Functions that are defined in such a way that the real and imaginary—or the magnitude and the phase—are processed separately using real-valued functions are referred to* as split-complex *functions. An example is*

$$f(z) = \tanh x + j \tanh y.$$

Obviously, the form $f(x, y)$ follows naturally for the given example but the form $f(z, z^*)$ does not.

Complex functions that are naturally defined as $f: \mathbb{C} \mapsto \mathbb{C}$, on the other hand, are referred to as fully-complex *functions. Examples include trigonometric functions and their hyperbolic counterparts such as $f(z) = \tanh(z)$. These functions typically provide better approximation ability and are more efficient in the characterization of the underlying nonlinear problem structure than the split-complex functions [62].*

We define the scalar inner product between two matrices $\mathbf{W}, \mathbf{V} \in \mathcal{V}$ as

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

such that $\langle \mathbf{W}, \mathbf{W} \rangle = \|\mathbf{W}\|^2$ and the superscript in $(\cdot)^H$ denotes the transpose of the complex conjugate. The norm we consider in this chapter is the Frobenius—also called the Euclidean—norm. For vectors, the definition simplifies to $\langle \mathbf{w}, \mathbf{v} \rangle = \mathbf{v}^H \mathbf{w}$. The definition of an inner product introduces a well-defined notion of orthogonality as well as of norm, and provides both computational and conceptual convenience. Inner product satisfies certain properties.

Properties of inner product:

positivity: $\langle \mathbf{V}, \mathbf{V} \rangle > \mathbf{0}$ for all $\mathbf{V} \in \mathcal{V}$;

definiteness: $\langle \mathbf{V}, \mathbf{V} \rangle = \mathbf{0}$ if and only if $\mathbf{V} = \mathbf{0}$;

linearity (additivity and homogeneity): $\langle \alpha(\mathbf{U} + \mathbf{W}), \mathbf{V} \rangle = \alpha \langle \mathbf{U}, \mathbf{V} \rangle + \alpha \langle \mathbf{W}, \mathbf{V} \rangle$ for all $\mathbf{W}, \mathbf{U}, \mathbf{V} \in \mathcal{V}$;

conjugate symmetry: $\langle \mathbf{W}, \mathbf{V} \rangle^* = \langle \mathbf{V}, \mathbf{W} \rangle$ for all $\mathbf{V}, \mathbf{W} \in \mathcal{V}$.

In the definition of the inner product, we assumed linearity in the first argument, which is more commonly used in engineering texts, though the alternate definition is also possible. Since our focus in this chapter is the finite-dimensional case, the inner product space also defines the Hilbert space.

A complex matrix $\mathbf{W} \in \mathbb{C}^{N \times N}$ is called symmetric if $\mathbf{W}^T = \mathbf{W}$ and Hermitian if $\mathbf{W}^H = \mathbf{W}$. Also, $\mathbf{W}$ is orthogonal if $\mathbf{W}^T \mathbf{W} = \mathbf{I}$ and unitary if $\mathbf{W}^H \mathbf{W} = \mathbf{I}$ where $\mathbf{I}$ is the identity matrix [49].

1.2.2 Efficient Computation of Derivatives in the Complex Domain

Differentiability and Analyticity Given a complex-valued function

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

where $z = x + jy$, the derivative of $f(z)$ at a point z_0 is written similar to the real case as

$$f'(z_0) = \lim_{\Delta z \to 0} \frac{f(z_0 + \Delta z) - f(z_0)}{\Delta z}.$$

However, different from the real case, due to additional dimensionality in the complex case, there is the added requirement that the limit should be independent of the direction of approach. Hence, if we first let $\Delta y = 0$ and evaluate $f'(z)$ by letting $\Delta x \to 0$, we have

$$f'(z) = u_x + jv_x \tag{1.1}$$

and, similarly, if we first let $\Delta x = 0$, and then $\Delta y \to 0$, we obtain

$$f'(z) = v_y - ju_y \tag{1.2}$$

where we have defined $u_x \triangleq \partial u/\partial x$, $u_y \triangleq \partial u/\partial y$, $v_x \triangleq \partial v/\partial x$, and $v_y \triangleq \partial v/\partial y$. For the existence of $f'(z)$, we thus require the equality of (1.1) and (1.2) at $z = z_0$ and in some neighborhood of z_0, which leads to the Cauchy–Riemann equations given by

$$u_x = v_y \quad \text{and} \quad v_x = -u_y. \tag{1.3}$$

A similar set of equations can be derived for other coordinate systems as well, such as polar [1]. The conditions given by (1.3) state the necessary conditions for the differentiability of $f(z)$. If, in addition, the partial derivatives of $u(x, y)$ and $v(x, y)$ exist and are continuous, then (1.3) are sufficient conditions as well.

Differentiability refers to the property of the function at a single point, and a function is called *analytic* (or holomorphic) if it is differentiable at every point in a given region. For example, the function $f(z) = z^*$ is analytic nowhere and $f(z) = 1/z^2$ is analytic for all finite $z \neq 0$. On the other hand, $f(z) = e^z$ is analytic in the entire finite z plane. Such functions are called *entire*.

In the study of analytic functions, a very fundamental result is given by Cauchy's integral theorem, which states that for a function $f(z)$ that is analytic throughout a region $\mathcal{U}$, the contour integral of $f(z)$ along any closed path lying inside $\mathcal{U}$ is zero. One of the most important consequences of Cauchy's integral theorem is a result stated by *Liouville's theorem* [95]:

> A bounded entire function must be a constant in the complex plane.

Hence, we cannot identify a function that is both bounded and analytic in the entire complex domain. Since boundedness is deemed as important for the performance—in particular stability—of nonlinear signal processing algorithms, a common practice has been to define functions that do not satisfy the analyticity requirement but are bounded (see *e.g.*, [9, 36, 45, 67, 103]). This has been the main motivation in the definition of split- and fully-complex functions given in Definition 1. 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 Sections 1.5 and 1.6, we discuss applications of both types of functions, split nonlinear functions that are proposed to circumvent the boundedness issue, and solutions that fully exploit the efficiency of complex domain processing.

Singular Points Singularities of a complex function $f(z)$ are defined as points z_0 in the domain of the function where $f(z)$ fails to be analytic. Singular points can be at a single point, that is, *isolated*, or nonisolated as in branch cuts or boundaries. Isolated singularities can be classified as removable singularities, poles, and essential singularities [1].

- A singular point is called a *removable singular point* if we have $f(z_0) \triangleq \lim_{z \to z_0} f(z)$, that is, the limit exists even though the function is not defined at

that point. In this case, the function can be written as an analytic function by simply defining the function at z_0 as $f(z_0)$.

- When we have $\lim_{z\to z_0} |f(z)| \to \infty$ for $f(z)$ analytic in a region centered at z_0, that is, in $0 < |z - z_0| < R$, we say that z_0 is a *pole* of the function $f(z)$.
- If a singularity is neither a pole nor a removable singularity, it is called an *essential singularity*, that is, the limit $\lim_{z\to z_0} f(z)$ does not exist as a complex number and is not equal to infinity either.

A simple example for a function with removable singularity is the function

$$f(z) = \frac{\sin(z - z_0)}{z - z_0}$$

which is not defined at $z = z_0$, but can be made analytic for all z by simply augmenting the definition of the function by $f(z_0) = 1$.

The function

$$f(z) = \frac{1}{(z - z_0)^N}$$

where N is an integer, is an example for a function with a pole. The pole at $z = z_0$ is called a simple pole if $N = 1$ and an Nth order pole if $N > 1$.

The essential singularity class is an interesting case and the rare example is found in functions of the form

$$f(z) = e^{1/z}.$$

This function has different limiting values for $z = 0$ depending on the direction of approach as we have $\lim_{z\to 0^{\pm j}} f(z) = 1$, $\lim_{z\to 0^-} f(z) = 0$, and $\lim_{z\to 0^+} f(z) = \infty$. A powerful property of essential singular points is given by Picard's theorem, which states that in any neighborhood of an essential singularity, a function, $f(z)$, assumes all values, except possibly one of them, an infinite number of times [1].

A very important class of functions that are not analytic anywhere on the complex plane are functions that are real valued, that is, $f\colon \mathbb{C} \mapsto \mathbb{R}$ and thus have $v(x, y) = 0$. Since the cost functions are real valued, their optimization thus poses a challenge, and is typically achieved using separate evaluations of real and imaginary parts of the function. As we discuss next, Wirtinger calculus provides a convenient framework to significantly simplify the evaluations of derivatives in the complex domain.

Wirtinger Derivatives As discussed in Section 1.2.2, differentiability, and hence analyticity are powerful concepts leading to important results such as the one summarized by Liouville's theorem. But—perhaps not surprisingly—their powerful nature also implies quite stringent conditions that need to be satisfied. When we look closely at the conditions for differentiability described by the Cauchy–Riemann equations (1.3), it is quite evident that they impose a strong structure on $u(x, y)$ and $v(x, y)$, the real and imaginary parts of the function, and consequently on

$f(z)$, as also discussed in [64]. A simple demonstration of this fact is that, *to express the derivatives of an analytic function, we only need to specify either* $u(x, y)$ *or* $v(x, y)$, *and do not need both.*

An elegant approach due to Wirtinger [115], which we explain next, 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.

To proceed, we first introduce the notion of *real differentiability*. In the introduction of Wirtinger calculus, the commonly used definition of differentiability that leads to the Cauchy–Riemann equations is identified as *complex differentiability*, and *real differentiability* is defined as a more flexible form.

Definition 2 *A function* $f(z) = u(x, y) + jv(x, y)$ *is called real differentiable when* $u(x, y)$ *and* $v(x, y)$ *are differentiable as functions of real-valued variables* x *and* y.

Note that this definition is quite flexible in that most nonanalytic as well as analytic functions satisfy the property as long as they have real and imaginary parts that are smooth (differentiable) functions of x and y.

To derive the form of the differential operators, we write the two real-variables as

$$x = \frac{z + z^*}{2} \quad \text{and} \quad y = \frac{z - z^*}{2j} \tag{1.4}$$

and use the chain rule to derive the form of the two derivative operators for $f(z)$ as

$$\frac{\partial f}{\partial z} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial z} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial z} = \frac{\partial f}{\partial x}\frac{1}{2} + \frac{\partial f}{\partial y}\frac{1}{2j}$$

and

$$\frac{\partial f}{\partial z^*} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial z^*} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial z^*} = \frac{\partial f}{\partial x}\frac{1}{2} - \frac{\partial f}{\partial y}\frac{1}{2j}.$$

The key point in the derivation given above is to treat 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 which we introduce next.

We consider a given function $f\colon \mathbb{C} \mapsto \mathbb{C}$ as a function $f\colon \mathbb{R} \times \mathbb{R} \mapsto \mathbb{C}$ by writing it as $f(z) = f(x, y)$, and make use of the underlying $\mathbb{R}^2$ structure by the following theorem [15].

Theorem 1.2.1 *Let* $f\colon \mathbb{R} \times \mathbb{R} \to \mathbb{C}$ *be a function of real variables* x *and* y *such that* $f(z, z^*) = f(x, y)$, *where* $z = x + jy$ *and that* $f(z, z^*)$ *is analytic with respect to* z^* *and* z *independently. Then,*

(1) *The partial derivatives*

$$\frac{\partial f}{\partial z} = \frac{1}{2}\left(\frac{\partial f}{\partial x} - j\frac{\partial f}{\partial y}\right) \quad \textit{and} \quad \frac{\partial f}{\partial z^*} = \frac{1}{2}\left(\frac{\partial f}{\partial x} + j\frac{\partial f}{\partial y}\right) \tag{1.5}$$

can be computed by treating z^* and z as a constant in $f(z, z^*)$ respectively; and

(2) *A necessary and sufficient condition for f to have a stationary point is that $\partial f/\partial z = 0$. Similarly, $\partial f/\partial z^* = 0$ 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, that is, by evaluating the right side of the equations in (1.5). To do so, we write the given function $f(z)$ in the form $f(z, z^*)$ and when evaluating the derivative with respect to z, we treat z^* as a constant as done in the computation of multi-variable function derivatives, and similarly treat z as a constant when evaluating $\partial f/\partial z^*$. The requirement for the analyticity of $f(z, z^*)$ with respect to z and z^* independently is 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 in (1.4) [64, 95]. Even though the condition of real differentiability is easily satisfied, separate evaluations of real and imaginary parts has been the common practice in the literature (see *e.g.*, [34, 38, 39, 63, 67, 103]).

When $f(z)$ is analytic, that is, when the Cauchy–Riemann conditions hold in a given open set, $f(\cdot)$ becomes a function of only z, and the two derivatives, the one given in the theorem and the traditional one coincide [95]. Alternatively put, all analytic functions are independent of z^* and only depend on z. This point can be easily verified using the definitions given in (1.5) and observing that when the Cauchy–Riemann equations are satisfied, we do end up with $f'(z)$ as given in (1.1) and (1.2), and we have $f'(z^*) = 0$.

For the application of Wirtinger derivatives for scalar-valued functions, consider the following two examples.

■ EXAMPLE 1.2

Consider the real-valued function $f(z) = |z|^4 = x^4 + 2x^2y^2 + y^4$. The derivative of the function can be calculated using (1.5) as

$$f'(z) \triangleq \frac{\partial f}{\partial z} = \frac{1}{2}\left(\frac{\partial f}{\partial x} - j\frac{\partial f}{\partial y}\right) = 2x^3 + 2xy^2 - 2j(x^2y + y^3) \tag{1.6}$$

or, to make use of Wirtinger derivative, we can write the function as $f(z) = f(z, z^*) = z^2(z^*)^2$ and evaluate the derivative as

$$\frac{\partial f}{\partial z} = 2z(z^*)^2 \tag{1.7}$$

that is, by treating z^* as a constant in f when calculating the partial derivative. It can be easily shown that the two forms, (1.6) and (1.7), are equal.

We usually define functions of interest in terms of z and would like to keep the expressions in that form, hence typically, one would need to write (1.6) in terms of z. As this simple example demonstrates, depending on the function in question, this might not always be a straightforward task.

■ EXAMPLE 1.3

As another example, consider evaluation of the conjugate derivative for the real-valued function $f(z) = |g(z)|^2$ where $g(z)$ is any analytic function. Since, in general we cannot explicitly write the real and imaginary parts of such a function in terms of x and y, we write

$$g(z) = u(x, y) + jv(x, y)$$

so that we have

$$f(z) = u^2(x, y) + v^2(x, y).$$

The derivative can then be evaluated using (1.5) as

$$\begin{aligned}\frac{\partial f}{\partial z^*} &= \frac{1}{2}\left(\frac{\partial f}{\partial x} + j\frac{\partial f}{\partial y}\right) \\ &= uu_x + vv_x + j(uu_y + vv_y) \\ &= g(z)[g'(z)]^* \end{aligned} \tag{1.8}$$

where u_x, u_y, v_x, and v_y are defined in (1.1) and (1.2), and we used the Cauchy–Riemann conditions for $g(z)$ when writing the last equality.

Alternatively, we can write the function as

$$f(z) = g(z)[g(z)]^* = g(z)g(z^*)$$

where the last equality follows when we have $g^*(z)^* = g(z^*)$. Then, directly using the Wirtinger derivative we have the same form given in (1.8) for $\partial f/\partial z^*$.

The condition in Example 1.3, $g^*(z) = g(z^*)$ which also implies $[g'(z)]^* = g'(z^*)$, is satisfied for a wide class of functions. It is easy to observe that it is true for all real-valued functions, and also for all functions $g(z)$ that have a Taylor series expansion with all real coefficients in $|z| < R$. Hence, all functions that are analytic within a neighborhood of zero satisfy the equality.

Example 1.3 also underlines another important point we have made earlier in the chapter regarding the desirability of directly working in the complex domain. When using the approach that treats real and imaginary parts separately, we needed a certain relationship between the real and imaginary parts of the function to write the derivative $f'(z)$ again in terms of z. The condition in this example was satisfied by analyticity of

the function as we used the Cauchy–Riemann conditions, that is, a strict relationship between the real and imaginary parts of the function.

The same approach of treating the variable and its complex conjugate as independent variables, can be used when taking derivatives of functions of matrix variables as well so that expressions given for real-valued matrix derivatives can be directly used as shown in the next example.

■ **EXAMPLE 1.4**

Let $g(\mathbf{Z}, \mathbf{Z}^*) = \text{Trace}(\mathbf{Z}\mathbf{Z}^H)$. We can calculate the derivatives of g with respect to $\mathbf{Z}$ and $\mathbf{Z}^*$ by simply treating one variable as a constant and directly using the results from real-valued matrix differentiation as

$$\frac{\partial g}{\partial \mathbf{Z}} = \frac{\partial \text{Trace}[\mathbf{Z}(\mathbf{Z}^*)^T]}{\partial \mathbf{Z}} = \mathbf{Z}^*$$

and

$$\frac{\partial g}{\partial \mathbf{Z}^*} = \mathbf{Z}$$

A good reference for real-valued matrix derivatives is [88] and a number of complex-valued matrix derivatives are discussed in detail in [46].

For computing matrix derivatives, a convenient tool is the use of differentials. In this procedure, first the matrix differential is computed and then it is written in the canonical form by identifying the term of interest. The differential of a function is defined as the part of a function $f(\mathbf{Z} + \Delta\mathbf{Z}) - f(\mathbf{Z})$ that is linear in $\mathbf{Z}$. For example when computing the differential of the function $f(\mathbf{Z}, \mathbf{Z}^*) = \mathbf{Z}\mathbf{Z}^*$, we can first write the product of the two differentials

$$(\mathbf{Z} + d\mathbf{Z})(\mathbf{Z}^* + d\mathbf{Z}^*) = \mathbf{Z}\mathbf{Z}^* + (d\mathbf{Z})\mathbf{Z}^* + \mathbf{Z}\,d\mathbf{Z}^* + d\mathbf{Z}\,d\mathbf{Z}^*$$

and take the first-order term (part of the expansion linear in $\mathbf{Z}$ and $\mathbf{Z}^*$) to evaluate the differential of the function as

$$d(\mathbf{Z}\mathbf{Z}^*) = (d\mathbf{Z})\mathbf{Z}^* + \mathbf{Z}d\mathbf{Z}^*$$

as discussed in [74, 78]. The approach can significantly simplify certain derivations. We provide an example for the application of the approach in Section 1.6.1.

Integrals of the Function $f(z, z^*)$ Though the three representations of a function we have discussed so far: $f(z)$, $f(x, y)$, and $f(z, z^*)$ are all equivalent, certain care needs to be taken when using each form, especially when using the form $f(z, z^*)$. This is the form that enables us to treat z and z^* as independent variables when taking derivatives and hence provides a very convenient representation (mapping) of a complex function in most evaluations. Obviously, the two variables are not independent

as knowing z we already know its conjugate. This is an issue that needs special care in evaluations such as integrals, which is needed for example, when using $f(z, z^*)$ to denote probability density functions and calculating the probabilities with this form.

In the evaluation of integrals, when we consider $f(\cdot)$ as a function of real and imaginary parts, the definition of an integral is well understood as the integral of function $f(x, y)$ in a region $\mathcal{R}$ defined in the (x, y) space as

$$\int\int_{\mathcal{R}} f(x, y)\, dx\, dy.$$

However, the integral $\int\int f(z, z^*)\, dz\, dz^*$ is not meaningful as we cannot vary the two variables z and z^* independently, and cannot define the region corresponding to $\mathcal{R}$ in the complex domain. However, this integral representation serves as an intermediate step when writing the real-valued integral as a contour integral in the complex domain using Green's theorem [1] or Stokes's theorem [44, 48] as noted in [87]. We can use Green's theorem (or Stokes's theorem) along with the definitions for the complex derivative given in (1.5) to write

$$\int\int_{\mathcal{R}} f(x, y)\, dx\, dy = -\frac{j}{2} \oint_{\mathcal{C}_{\mathcal{R}}} F(z, z^*)\, dz \tag{1.9}$$

where

$$\frac{\partial F(z, z^*)}{\partial z^*} = f(z, z^*).$$

Here, we assume that $f(x, y)$ is continuous through the simply connected region $\mathcal{R}$ and $\mathcal{C}_{\mathcal{R}}$ describes its contour. Note that by transforming the integral defined in the real domain to a contour integral when the function is written as $f(z, z^*)$, the formula takes into account the dependence of the two variables, z and z^* in a natural manner.

In [87], the application of the integral relationship in (1.9) is discussed in detail for the evaluation of probability masses when $f(x, y)$ defines a probability density function. Three cases are identified as important and a number of examples are studied as application of the formula. The three specific cases to consider for evaluation of the integral in (1.9) are when

- $F(z, z^*)$ is an analytic function inside the given contour, that is, it is a function of z only in which case the integral is zero by Cauchy's theorem;
- $F(z, z^*)$ contains poles inside the contour, which in the case of probability evaluations will correspond to probability masses inside the given region;
- $F(z, z^*)$ is not analytic inside the given contour in which case the value of the integral will relate to the size of the region $\mathcal{R}$.

We demonstrate the use of the integral formula given in (1.9) in Section 1.6.4 in the derivation of an efficient representation for the score function for complex maximum likelihood based independent component analysis.

It is also worth noting that the dependence in the variables z and z^* is different in the computation of derivatives. In [31], the author discusses polarization of an analytic identity and notes that complex-valued functions of z and z^* have linearly independent differentials dz and dz^*, and hence z and z^* are *locally* functionally independent. Still, we treat the form $f(z, z^*)$ as primarily a notational form that renders computations of derivatives simple and note the fact that special care must be taken when using the form to define quantities such as probability density functions.

Derivatives of Cost Functions The functions we typically work with in the development of signal processing algorithms are cost functions, hence these are real valued such that $f \in \mathbb{R}$. Since the class of real-valued functions is a special case of the functions considered in Theorem 1.2.1, we can employ the same procedure for this case as well and take the derivatives by treating z and z^* as independent from each other. In this chapter, we mainly consider such functions as these are the cost functions used in the derivation of adaptive signal processing algorithms. However, in the discussion, we identify the deviation, if any, from the general $f\colon \mathbb{R} \times \mathbb{R} \to \mathbb{C}$ case for completeness. Also note that when $f(z)$ is real valued, we have

$$\left(\frac{\partial f}{\partial z}\right)^* = \frac{\partial f}{\partial z^*}$$

that is, the derivative and the conjugate derivative are complex conjugates of each other.

1.2.3 Complex-to-Real and Complex-to-Complex Mappings

In this chapter, we emphasize working in the original space in which the functions are defined, even when they are not analytic. The approach is attractive for two reasons. First, it is straightforward and eliminates the need to perform transformations to and back from another space where the computations are carried out. Second, it does not increase the dimensionality of the problem. In certain cases though, in particular for the form of multidimensional transformation defined by van den Bos [110], the increase in dimensionality might offer advantages. As we discuss in this section, the $C^N \mapsto C^{2N}$ mapping given by van den Bos provides a smart way of taking advantage of Wirtinger calculus, and can lead to certain simplifications in the expression. For completeness, we discuss all major transformations that have been used in the literature for multivariate complex analysis, especially when working with non-analytic functions.

Vector-Concatenation Type Mappings The two mappings in this class, the $\overline{(\cdot)}_R$ and $\overline{(\cdot)}_C$ mappings have very different uses. The most commonly used mapping $\mathbb{C}^N \mapsto \mathbb{R}^{2N}$ takes a very simple form and is written such that

$$\mathbf{z} \in \mathbb{C}^N \mapsto \bar{\mathbf{z}}_R = \begin{bmatrix} \mathbf{z}_r \\ \mathbf{z}_i \end{bmatrix} \in \mathbb{R}^{2N} \tag{1.10}$$

and for a matrix $\mathbf{A}$ as

$$\mathbf{A} \in \mathbb{C}^{M\times N} \mapsto \bar{\mathbf{A}}_R = \begin{bmatrix} \mathbf{A}_r & -\mathbf{A}_i \\ \mathbf{A}_i & \mathbf{A}_r \end{bmatrix} \in \mathbb{R}^{2M\times 2N}. \tag{1.11}$$

It can be easily shown that $(\overline{\mathbf{Az}})_R = \bar{\mathbf{A}}_R\bar{\mathbf{z}}_R$.

The mapping provides a natural isomorphism between $\mathbb{C}^N$ and $\mathbb{R}^{2N}$, and thus is a practical approach for derivations in the complex domain. For example, in [40], the mapping is used for statistical analysis of multivariate complex Gaussian distribution and in [20] to derive the relative gradient update rule for independent component analysis.

Note that the real-vector space defined through the $\overline{(\cdot)}$ mapping is isomorphic to the standard real vector space $\mathbb{R}^{2N}$. In fact, we can define an orthogonal decomposition of the space of $2N \times 2N$ matrices such that a given matrix $\mathbf{M} \in \mathbb{R}^{2N\times 2N}$ is written in terms of four blocks of size $N \times N$ as

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix}.$$

Thus, the linear space of $2N \times 2N$ matrices can be decomposed into two orthogonal spaces: $\mathbb{R}^{2N\times 2N} = \mathcal{M}^+ \oplus \mathcal{M}^-$ where $\mathcal{M}^+$ (*resp.* $\mathcal{M}^-$) contains any matrix such that $\mathbf{M}_{11} = \mathbf{M}_{22}$ and $\mathbf{M}_{12} = -\mathbf{M}_{21}$ (*resp.* $\mathbf{M}_{11} = -\mathbf{M}_{22}$ and $\mathbf{M}_{12} = \mathbf{M}_{21}$). Hence a $2N \times 2N$ real matrix has the orthogonal decomposition $\mathbf{M} = \mathbf{M}^+ + \mathbf{M}^-$ where

$$\begin{aligned} \mathbf{M}^+ &= \frac{1}{2}\begin{bmatrix} \mathbf{M}_{11} + \mathbf{M}_{22} & \mathbf{M}_{12} - \mathbf{M}_{21} \\ \mathbf{M}_{21} - \mathbf{M}_{12} & \mathbf{M}_{11} + \mathbf{M}_{22} \end{bmatrix} \in \mathcal{M}^+ \quad \text{and} \\ \mathbf{M}^- &= \frac{1}{2}\begin{bmatrix} \mathbf{M}_{11} - \mathbf{M}_{22} & \mathbf{M}_{12} + \mathbf{M}_{21} \\ \mathbf{M}_{21} + \mathbf{M}_{12} & -\mathbf{M}_{11} + \mathbf{M}_{22} \end{bmatrix} \in \mathcal{M}^-. \end{aligned} \tag{1.12}$$

Note that the set of invertible matrices of $\mathbf{M}^+$ form a group for the usual multiplication of matrices and we have $\bar{\mathbf{A}}_R \in \mathcal{M}^+$, which is defined in (1.11).

The following are some useful properties of this complex-to-real mapping and can be verified using the isomorphism between the two spaces [20, 33, 40].

Properties of Complex-to-Real Mapping $\overline{(\cdot)}$: $\mathbb{C}^N \to \mathbb{R}^{2N}$ Let $\mathbf{A}$, $\mathbf{B} \in \mathbb{C}^{N\times N}$ and $\mathbf{z}$, $\mathbf{y} \in \mathbb{C}^N$, then for the mapping $\overline{(\cdot)}_R$, we have

(1) $\overline{(\mathbf{AB})}_R = \bar{\mathbf{A}}_R\bar{\mathbf{B}}_R$ and thus $\overline{(\mathbf{A}^{-1})}_R = (\bar{\mathbf{A}}_R)^{-1}$.
(2) $|\det(\mathbf{A})|^2 = \det(\bar{\mathbf{A}}_R)$.
(3) $\mathbf{A}$ is Hermitian if and only if $\bar{\mathbf{A}}_R$ is symmetric.
(4) $\mathbf{A}$ is nonsingular if and only if $\bar{\mathbf{A}}_R$ is nonsingular.
(5) $\mathbf{A}$ is unitary if and only if $\bar{\mathbf{A}}_R$ is orthogonal.
(6) $\mathbf{A}$ is positive definite if and only if $\bar{\mathbf{A}}_R$ is positive definite.

(7) $\mathbf{z}^H\mathbf{A}\mathbf{z} = \bar{\mathbf{z}}_R^T\bar{\mathbf{A}}_R\bar{\mathbf{z}}$.
(8) $\overline{\mathbf{z}\mathbf{y}^H}_R = 2(\bar{\mathbf{z}}_R\bar{\mathbf{y}}_R)^+$ where $(\cdot)^+$ is defined in (1.12).

In certain scenarios, for example, when working with probabilistic descriptions, or when evaluating the derivative of matrix functions, the $\mathbb{C}^N \mapsto \mathbb{R}^{2N}$ transformation can simplify the evaluations and lead to simpler forms (see *e.g.* [4, 20]).

The second mapping in this class is defined by simple concatenation of the complex vector and its complex conjugate as

$$\mathbf{z} \in \mathbb{C}^N \mapsto \bar{\mathbf{z}}_C = \begin{bmatrix} \mathbf{z} \\ \mathbf{z}^* \end{bmatrix} \in \mathbb{C}^{2N}. \tag{1.13}$$

This mapping can be useful as an intermediate step when establishing certain relationships as shown in [64] and [71]. More importantly, this vector definition provides a convenient representation for the widely linear transform, which enables incorporation of full second-order statistical information into the estimation scheme and provides significant advantages when the signal is noncircular [94]. We discuss the approach and present the main results for minimum mean square error filtering using Wirtinger calculus in Section 1.4.

Element-wise Mappings In the development that leads to the definition of Wirtinger derivatives, the key observation is the duality of the two spaces: $\mathbb{R}^2$ and $\mathbb{C}^2$ through the transformation

$$(z_r, z_i) \iff (z, z^*).$$

Hence, if a function is real differentiable as a function of the two real-valued variables z_r and z_i, then it satisfies the condition for real differentiability, and the two variables, z and z^* can be treated as independent in $\mathbb{C}^2$ to take advantage of Wirtinger calculus. To extend this idea to the multidimensional case, van den Bos [110] defined the two mappings $\tilde{(\cdot)}$ given in Table 1.2 such that

$$\tilde{\mathbf{z}}_R = \begin{bmatrix} z_{r,1} \\ z_{i,1} \\ z_{r,2} \\ z_{i,2} \\ \vdots \\ z_{r,N} \\ z_{i,N} \end{bmatrix} \iff \tilde{\mathbf{z}}_C = \begin{bmatrix} z_1 \\ z_1^* \\ z_2 \\ z_2^* \\ \vdots \\ z_N \\ z_N^* \end{bmatrix} \tag{1.14}$$

where $\tilde{\mathbf{z}}_R \in \mathbb{R}^{2N}$ and $\tilde{\mathbf{z}}_C \in \mathbb{C}^{2N}$. In [110], the whole development is given as an extension of Brandwood's work [15] without any reference to Wirtinger calculus in particular.

Table 1.2 Four primary mappings defined for $\mathbf{z} = \mathbf{z}_r + j\mathbf{z}_i \in \mathbb{C}^N$

	Complex-to-Real: $\mathbb{C}^N \mapsto \mathbb{R}^{2N}$	Complex-to-Complex: $\mathbb{C}^N \mapsto \mathbb{C}^{2N}$
Vector-concatenation type mappings	$\bar{\mathbf{z}}_R = \begin{bmatrix} \mathbf{z}_r \\ \mathbf{z}_i \end{bmatrix}$	$\bar{\mathbf{z}}_C = \begin{bmatrix} \mathbf{z} \\ \mathbf{z}^* \end{bmatrix}$
Element-wise mappings	$\tilde{\mathbf{z}}_R = \begin{bmatrix} z_{r,1} \\ z_{i,1} \\ \vdots \\ z_{r,N} \\ z_{i,N} \end{bmatrix}$	$\tilde{\mathbf{z}}_C = \begin{bmatrix} z_1 \\ z_1^* \\ \vdots \\ z_N \\ z_N^* \end{bmatrix}$

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 [110], 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 matrix. In the next section, we establish the connections of the results of [110] to $\mathbb{C}^N$ for first- and second-order derivatives such that efficient second-order optimization algorithms can be derived by directly working in the original $\mathbb{C}^N$ space where the problems are typically defined.

Relationship Among Mappings It can be easily observed that all four mappings defined in Table 1.2 are related to each other through simple linear transformations, thus making it possible to work in one domain and then transfer the solution to another. Two key transformations are given by $\bar{\mathbf{z}}_C = \mathbf{U}\bar{\mathbf{z}}_R$ and $\tilde{\mathbf{z}}_C = \tilde{\mathbf{U}}\tilde{\mathbf{z}}_R$ where

$$\mathbf{U} = \begin{bmatrix} \mathbf{I} & j\mathbf{I} \\ \mathbf{I} & -j\mathbf{I} \end{bmatrix}$$

and $\tilde{\mathbf{U}} = \text{diag}\{\mathbf{U}_2, \ldots, \mathbf{U}_2\}$ where $\mathbf{U}_2 = \begin{bmatrix} 1 & j \\ 1 & -j \end{bmatrix}$. It is easy to observe that for the transformation matrices $\mathbf{U}$ defined above, we have $\mathbf{U}\mathbf{U}^H = \mathbf{U}^H\mathbf{U} = 2\mathbf{I}$ making it easy to obtain inverse transformations as we demonstrate in Section 1.3. For transformations between the two mappings, $\overline{(\cdot)}$ and $\tilde{(\cdot)}$, we can use permutation matrices that are orthogonal, thus allowing simple manipulations.

1.2.4 Series Expansions

Series expansions are a valuable tool in the study of nonlinear functions, and for analytic functions, that is, functions that are complex differentiable in a given

region, the Taylor series expression assumes the same form as in the real case given by

$$f(z) = \sum_{k=0}^{\infty} \frac{f^{(k)}(z_0)}{k!}(z - z_0)^k. \tag{1.15}$$

If $f(z)$ is analytic for $|z| \leq R$, then the Taylor series given in (1.15) converges uniformly in $|z| \leq R_1 < R$. The notation $f^{(k)}(z_0)$ refers to the kth order derivative evaluated at z_0 and when the power series expansion is written for $z_0 = 0$, it is called the Maclaurin series.

In the development of signal processing algorithms (parameter update rules) and in stability analyses, the first- and second-order expansions prove to be the most useful. For an analytic function $f(\mathbf{z})$: $\mathbb{C}^N \mapsto \mathbb{C}$, we define $\Delta f = f(\mathbf{z}) - f(\mathbf{z}_0)$ and $\Delta \mathbf{z} = \mathbf{z} - \mathbf{z}_0$ to write the second-order approximation to the function in the neighborhood of $\mathbf{z}_0$ as

$$\begin{aligned} \Delta f &\approx \Delta \mathbf{z}^T \nabla_{\mathbf{z}} f + \frac{1}{2} \Delta \mathbf{z}^T \mathbf{H}(\mathbf{z})\, \Delta \mathbf{z} \\ &= \left\langle \nabla_{\mathbf{z}} f,\ \Delta \mathbf{z}^* \right\rangle + \frac{1}{2} \left\langle \mathbf{H}(\mathbf{z})\, \Delta \mathbf{z},\ \Delta \mathbf{z}^* \right\rangle \end{aligned} \tag{1.16}$$

where

$$\nabla_{\mathbf{z}} f = \left. \frac{\partial f(\mathbf{z})}{\partial \mathbf{z}} \right|_{\mathbf{z}_0}$$

is the gradient evaluated at $\mathbf{z}_0$ and

$$\nabla_{\mathbf{z}}^2 f \triangleq \mathbf{H}(\mathbf{z}) = \left. \frac{\partial^2 f(\mathbf{z})}{\partial \mathbf{z} \partial \mathbf{z}^T} \right|_{\mathbf{z}_0}$$

is the Hessian matrix evaluated at $\mathbf{z}_0$. As in the real-valued case, the Hessian matrix in this case is symmetric and constant if the function is quadratic.

Second-order Taylor series expansions as given in (1.16) help summarize main results for optimization and local stability analysis. In particular, we can state the following three important observations for the real-valued case, that is, when the argument $\mathbf{z}$ and the function are real valued, by directly studying the expansion given in (1.16).

- Point $\mathbf{z}_0$ is a *local minimum* of $f(\mathbf{z})$ when $\nabla_{\mathbf{z}} f = 0$ and $\mathbf{H}(\mathbf{z})$ is positive semi-definite, that is, these are the necessary conditions for a local minimum.
- When $\mathbf{H}(\mathbf{z})$ is positive definite and $\nabla_{\mathbf{z}} f = 0$, $\mathbf{z}_0$ is *guaranteed* to be a local minimum, that is, positive-definiteness and zero gradient, together, define the sufficient condition.
- Finally, $\mathbf{z}_0$ is a *locally stable* point if, and only if, $\mathbf{H}(\mathbf{z})$ is positive definite and $\nabla_{\mathbf{z}} f = 0$, that is, in this case, the two properties define the sufficient and necessary conditions.

When deriving complex-valued signal processing algorithms, however, the functions of interest are real valued and have complex arguments $\mathbf{z}$, hence are not analytic

on the complex plane. In this case, we can use Wirtinger calculus and write the expansions by treating the function $f(\mathbf{z})$ as a function of two arguments, $\mathbf{z}$ and $\mathbf{z}^*$. In this approach, the main idea is treating the two arguments as independent from each other, when they are obviously dependent on each other as we discussed. When writing the Taylor series expansion, the idea is the same. We write the series expansion for a real-differentiable function $f(\mathbf{z}) = f(\mathbf{z}, \mathbf{z}^*)$ as if $\mathbf{z}$ and $\mathbf{z}^*$ were independent variables, that is, as

$$\Delta f(\mathbf{z}, \mathbf{z}^*) \approx \langle \nabla_{\mathbf{z}} f, \Delta \mathbf{z}^* \rangle + \langle \nabla_{\mathbf{z}^*} f, \Delta \mathbf{z} \rangle + \frac{1}{2} \left\langle \frac{\partial f}{\partial \mathbf{z} \partial \mathbf{z}^T} \Delta \mathbf{z}, \Delta \mathbf{z}^* \right\rangle + \left\langle \frac{\partial f}{\partial \mathbf{z} \partial \mathbf{z}^H} \Delta \mathbf{z}^*, \Delta \mathbf{z}^* \right\rangle + \frac{1}{2} \left\langle \frac{\partial f}{\partial \mathbf{z}^* \partial \mathbf{z}^H} \Delta \mathbf{z}^*, \Delta \mathbf{z} \right\rangle. \tag{1.17}$$

In other words, the series expansion has the same form as a real-valued function of two variables which happen to be replaced by $\mathbf{z}$ and $\mathbf{z}^*$ as the two independent variables. Note that when $f(\mathbf{z}, \mathbf{z}^*)$ is real valued, we have

$$\langle \nabla_{\mathbf{z}} f, \Delta \mathbf{z}^* \rangle + \langle \nabla_{\mathbf{z}^*} f, \Delta \mathbf{z} \rangle = 2\mathrm{Re}\{ \langle \nabla_{\mathbf{z}^*} f, \Delta \mathbf{z} \rangle \} \tag{1.18}$$

since in this case we have $\nabla f(\mathbf{z}^*) = [\nabla f(\mathbf{z})]^*$. Using the Cauchy–Bunyakovskii–Schwarz inequality [77], we have

$$|\Delta \mathbf{z}^H \nabla f(\mathbf{z}^*)| \leq \|\Delta \mathbf{z}\| \, \|\nabla f(\mathbf{z}^*)\|$$

which holds with equality when $\Delta \mathbf{z}$ is in the same direction as $\nabla f(\mathbf{z}^*)$. Hence, it is the gradient with respect to the complex conjugate of the variable $\nabla f(\mathbf{z}^*)$ that yields the maximum change in function $\Delta f(\mathbf{z}, \mathbf{z}^*)$.

It is also important to note that when $f(\mathbf{z}, \mathbf{z}^*) = f(\mathbf{z})$, that is, the function is analytic (complex differentiable), all derivatives with respect to $\mathbf{z}^*$ in (1.17) vanish and (1.17) coincides with (1.16). As noted earlier, the Wirtinger formulation for real-differentiable functions includes analytic functions, and when the function is analytic, all the expressions used in the formulations reduce to the traditional ones for analytic functions.

For the transformations that map the function to the real domain as those given in Table 1.2, the $\tilde{(\cdot)}_R$ and $\overline{(\cdot)}_R$ mappings, the expansion is straightforward since in this case, the expansion is written in the real domain as in

$$\Delta f(\tilde{\mathbf{z}}_R) \approx \langle \nabla_{\tilde{\mathbf{z}}_R} f(\tilde{\mathbf{z}}_R), \Delta \tilde{\mathbf{z}}_R \rangle + \frac{1}{2} \langle \mathbf{H}(\tilde{\mathbf{z}}_R) \Delta \tilde{\mathbf{z}}_R, \Delta \tilde{\mathbf{z}}_R \rangle.$$

By using the complex domain transformation defined by van den Bos (1.14), a very similar form for the expansion can be obtained in the complex domain as well, and it is given by [110]

$$\Delta f(\tilde{\mathbf{z}}_C) \approx \langle \nabla_{\tilde{\mathbf{z}}_C^*} f(\tilde{\mathbf{z}}_C), \Delta \tilde{\mathbf{z}}_C \rangle + \frac{1}{2} \langle \mathbf{H}(\tilde{\mathbf{z}}_C) \Delta \tilde{\mathbf{z}}_C, \Delta \tilde{\mathbf{z}}_C \rangle \tag{1.19}$$

where

$$\mathbf{H}(\tilde{\mathbf{z}}_C) = \left.\frac{\partial^2 f(\tilde{\mathbf{z}}_C)}{\partial \tilde{\mathbf{z}}_C^* \, \partial \tilde{\mathbf{z}}_C^T}\right|_{\tilde{\mathbf{z}}_{C_0}}.$$

When writing the expansions in the transform domain, we assume that the function $f(\cdot)$ is written in terms of the transformed arguments, for example, we have $f(\mathbf{z}) = f(\tilde{\mathbf{z}}_C)$. Hence, in the expansions given in this section, we have included the variable explicitly in all the expressions.

The two Hessian matrices, $\mathbf{H}(\tilde{\mathbf{z}}_R)$ and $\mathbf{H}(\tilde{\mathbf{z}}_C)$ are related through the mapping

$$\mathbf{H}(\tilde{\mathbf{z}}_R) = \tilde{\mathbf{U}}^H \mathbf{H}(\tilde{\mathbf{z}}_C)\tilde{\mathbf{U}}$$

where $\tilde{\mathbf{U}}$ is defined in Section 1.2.3. Since the real-valued Hessian is a symmetric matrix—we assume the existence of continuous second-order derivatives of $f(\cdot)$—and $\tilde{\mathbf{U}}\tilde{\mathbf{U}}^H = 2\mathbf{I}$, the complex Hessian matrix $\mathbf{H}(\tilde{\mathbf{z}}_C)$ is Hermitian. Hence, we can write

$$\mathbf{H}(\tilde{\mathbf{z}}_R) - \lambda\mathbf{I} = \tilde{\mathbf{U}}^H[\mathbf{H}(\tilde{\mathbf{z}}_C) - 2\lambda\mathbf{I}]\tilde{\mathbf{U}}$$

and observe that if λ is an eigenvalue of $\mathbf{H}(\tilde{\mathbf{z}}_C)$, then 2λ is an eigenvalue of $\mathbf{H}(\tilde{\mathbf{z}}_R)$. Thus, when checking whether the Hessian is a positive definite matrix—for example, for local optimality and local stability properties—one can work with either form of the Hessian. Hence, other properties of the Hessian such as its condition number, which is important in a number of scenarios for example, when deriving second-order learning algorithms, are also preserved under the transformation [110].

Even though it is generally more desirable to work in the original space where the functions are defined, which is typically $\mathbb{C}^N$, the transformations given in Section 1.2.3 can provide simplifications to the series expansions. For example, the mapping $\tilde{(\cdot)}_C$ given in (1.14) can lead to simplifications in the expressions as demonstrated in [86] in the derivation and local stability analysis of a complex independent component analysis algorithm. The use of Wirtinger calculus through the $\mathbb{R}^2 \mapsto \mathbb{C}^2$ mapping in this case leads to a simpler block structure for the final Hessian matrix $\mathbf{H}(\tilde{\mathbf{z}}_C)$ compared to $\mathbf{H}(\tilde{\mathbf{z}}_R)$, hence simplifying assumptions such as circularity of random variables as done in [13] for a similar setting can be avoided.

In this section, we concentrated on functions of vector variables. For matrix variables, a first-order expansion can be obtained in a very similar manner. For a function $f(\mathbf{Z}, \mathbf{Z}^*)$: $\mathbb{C}^{N\times M} \times \mathbb{C}^{N\times M} \to \mathbb{R}$, we have

$$\begin{aligned}\Delta f(\mathbf{Z}, \mathbf{Z}^*) &\approx \langle \nabla_{\mathbf{Z}} f, \Delta\mathbf{Z}^* \rangle + \langle \nabla_{\mathbf{Z}^*} f, \Delta\mathbf{Z} \rangle \\ &= 2\mathrm{Re}\{\langle \nabla_{\mathbf{Z}^*} f, \Delta\mathbf{Z} \rangle\}\end{aligned} \tag{1.20}$$

where $\partial f/\partial \mathbf{Z}$ is an $N \times M$ matrix whose (k, l)th entry is the partial derivative of f with respect to w_{kl} and the last equality follows only for real-valued functions. Again, it is the gradient with respect to the conjugate variable, that is, $\nabla_{\mathbf{Z}^*} f$, the quantity that defines the direction of the maximum rate of change in f with respect to $\mathbf{Z}$ not the gradient $\nabla_{\mathbf{Z}} f$.

Since the definition of a Hessian for a function of the form $f(\mathbf{Z}, \mathbf{Z}^*)$ does not result in a matrix form and cannot be written as one of the six forms given in Table 1.1, there are a number of options when working with the second-order expansions in this case. One approach is to write the expression directly in terms of each element, which is given by

$$\nabla^2_{\mathbf{Z}} f = \frac{1}{2}\sum_{m,n}\sum_{k,l}\frac{\partial^2 f}{\partial z_{mn}\partial z_{kl}}dz_{mn}dz_{kl} + \frac{1}{2}\sum_{m,n}\sum_{k,l}\frac{\partial^2 f}{\partial z^*_{mn}\partial z^*_{kl}}dz^*_{mn}dz^*_{kl} + \sum_{m,n}\sum_{k,l}\frac{\partial^2 f}{\partial z_{mn}\partial z^*_{kl}}dz_{mn}dz^*_{kl}.$$

Note that this form is written by evaluating the second-order term in (1.17) with respect to every entry of matrix $\mathbf{Z}$. In certain cases, second-order matrix differentials can be put into compact forms using matrix differentials introduced in Section 1.2.2 and invariant transforms as in [7]. Such a procedure allows for efficient derivations while keeping all the evaluations in the original transform domain as demonstrated in the derivation of maximum likelihood based relative gradient update rule for complex independent component analysis in [68].

Another approach for calculating differential or Hessian expressions of matrix variables is to use the vectorization operator vec$(\cdot)$ that converts the matrix to a vector form by stacking the columns of a matrix into a long column vector starting from the first column [50]. Then the analysis proceeds by using vector calculus. The approach requires working with careful definitions of functions for manipulating the variables defined as such and then their reshaping at the end. This is the approach taken in [46] for defining derivatives of functions with matrix arguments.

1.2.5 Statistics of Complex-Valued Random Variables and Random Processes

Statistical Description of Complex Random Variables and Vectors A complex-valued random variable $X = X_r + jX_i$ is defined through the joint probability density function (pdf) $f_X(x) \triangleq f_{X_rX_i}(x_r, x_i)$ provided that it exists. For a pdf $f_{X_rX_i}(x_r, x_i)$ that is differentiable with respect to x_r and x_i individually, we can write $f_{X_rX_i}(x_r, x_i) = f(x, x^*)$ where $x = x_r + jx_i$, and use the expression written in terms of x and x^* in the evaluations to take advantage of Wirtinger calculus.

Note that writing the pdf in the form $f(x, x^*)$ is mainly a representation, which in most instances, significantly simplifies the evaluations. Thus, it is primarily a computational tool. As in the case of representation of any function using the variables x and x^* rather than only x, the form is degenerate since the two variables are not independent of each other. In [87], the evaluation of probability masses using the form $f(x, x^*)$ is discussed in detail, both for continuous and mixed-distribution random variables. When evaluating expected values using a pdf written as $f(x, x^*)$, we have to thus consider the contour integrals as given in (1.9).

The joint pdf for a complex random vector $\mathbf{X} \in \mathbb{C}^N$ is extended to the form $f(\mathbf{x}, \mathbf{x}^*)$: $\mathbb{C}^N \times \mathbb{C}^N \mapsto \mathbb{R}$ similarly. In the subsequent discussion, we write the expectations with respect to the corresponding joint pdf, pdf of a scalar or vector random variable as defined here.

Second-order statistics of a complex random vector $\mathbf{X}$ are completely defined through two (auto) covariance matrices: the covariance matrix

$$\mathbf{C}_{XX} = E\{(\mathbf{X} - E\{\mathbf{X}\})(\mathbf{X} - E\{\mathbf{X}\})^H\}$$

that is commonly used, and in addition, the *pseudo-covariance* [81] matrix—also called the complementary covariance [101] or the relation matrix [92]—given by

$$\mathbf{P}_{XX} = E\{(\mathbf{X} - E\{\mathbf{X}\})(\mathbf{X} - E\{\mathbf{X}\})^T\}.$$

Expressions are written similarly for the cross-covariance matrices $\mathbf{C}_{XY}$ and $\mathbf{P}_{XY}$ of two complex random vectors $\mathbf{X}$ and $\mathbf{Y}$. The properties given in Section 1.2.3 for complex-to-real mappings can be effectively used to work with covariance matrices in either the complex- or the double-dimensioned real domain. In the sequel, we drop the indices used in matrix definitions here when the matrices in question are clear from the context, and assume that the vectors are zero mean without loss of generality.

Through their definitions, the covariance matrix is a Hermitian and the pseudo-covariance matrix is a complex symmetric matrix. As is easily shown, the covariance matrix is nonnegative definite—and in practice typically positive definite. Hence, the nonnegative eigenvalues of the covariance matrix can be identified using simple eigenvalue decomposition. For the pseudo-covariance matrix, however, we need to use Takagi's factorization [49] to obtain the spectral representation such that

$$\mathbf{P} = \mathbf{Q}\mathbf{D}\mathbf{Q}^T$$

where $\mathbf{Q}$ is a unitary matrix and $\mathbf{D} = \text{diag}\{\kappa_1, \kappa_2, \ldots, \kappa_N\}$ contains the singular values, $1 \geq \kappa_1 \geq \kappa_2 \geq \cdots \geq \kappa_N \geq 0$, on its diagonal. The values κ_n are canonical correlations of a given vector and its complex conjugate [100] and are called the *circularity coefficients* [33]—though noncircularity coefficients might be the more appropriate name—since for a second-order circular random vector, which we define next, these values are all zero.

The vector transformation $\mathbf{z} \in \mathbb{C}^N \mapsto \bar{\mathbf{z}}_C \in \mathbb{C}^{2N}$ given in (1.13) can be used to define a single matrix summarizing the second-order properties of a random vector $\mathbf{X}$, which is called the augmented correlation matrix [92, 101]

$$E\{\bar{\mathbf{X}}_C \bar{\mathbf{X}}_C^H\} = E\left\{\begin{bmatrix}\mathbf{X} \\ \mathbf{X}^*\end{bmatrix}[\mathbf{X}^H \mathbf{X}^T]\right\} = \begin{bmatrix}\mathbf{C} & \mathbf{P} \\ \mathbf{P}^* & \mathbf{C}^*\end{bmatrix}$$

and is used in the study of widely linear least mean squares filter which we discuss in Section 1.4.

Circularity Properties of a Complex Random Variable and Random Vector An important property of complex-valued random variables is related to their circular nature.

A zero-mean complex random variable is called *second-order circular* [91] (or proper [81, 101]) when its pseudo-covariance is zero, that is,

$$E\{X^2\} = 0$$

which implies that $\sigma_{X_r} = \sigma_{X_i}$ and $E\{X_r X_i\} = 0$ where σ_{X_r} and σ_{X_i} are the standard deviations of the real and imaginary parts of the variable.

For a random vector $\mathbf{X}$, the condition for second-order circularity is written in terms of the pseudo-covariance matrix as $\mathbf{P} = 0$, which implies that $E\{\mathbf{X}_r\mathbf{X}_r^T\} = E\{\mathbf{X}_i\mathbf{X}_i^T\}$ and $E\{\mathbf{X}_r\mathbf{X}_i^T\} = -E\{\mathbf{X}_i\mathbf{X}_r^T\}$.

A stronger condition for circularity is based on the pdf of the random variable.

A random variable X is called *circular in the strict-sense*, or simply *circular*, if X and $Xe^{j\theta}$ have the same pdf, that is, the pdf is rotation invariant [91].

In this case, the phase is non-informative and the pdf is a function of only the magnitude, $f_X(x) = g(|x|)$ where g: $\mathbb{R} \mapsto \mathbb{R}$, hence the pdf can be written as a function of zz^* rather than z and z^* separately. A direct consequence of this property is that $E\{X^p(X^*)^q\} = 0$ for all $p \neq q$ if X is circular. Circularity is a strong property, preserved under linear transformations, and since it implies noninformative phase, a real-valued approach and a complex-valued approach for this case are usually equivalent [109].

As one would expect, circularity implies second-order circularity, and only for a Gaussian-distributed random variable, second-order circularity implies (strict sense) circularity. Otherwise, the reverse is not true.

For random vectors, in [91], three different types of circularity are identified. A random vector $\mathbf{X} \in \mathbb{C}^N$ is called

- *marginally circular* if each component of the random vector X_n is a circular random variable;
- *weakly circular* if $\mathbf{X}$ and $\mathbf{X}e^{j\theta}$ have the same distribution for any given θ; and
- *strongly circular* if $\mathbf{X}$ and $\mathbf{X}'$ have the same distribution where $\mathbf{X}'$ is formed by rotating the corresponding entries (random variables) in $\mathbf{X}$ by θ_n, such that $X'_n = X_n e^{j\theta_n}$. This condition is satisfied when θ_k are independent and identically distributed random variables with uniform distribution in $[-\pi, \pi]$ and are independent of the amplitude of the random variables, X_n.

As the definitions suggest, strong circularity implies weak circularity, and weak circularity implies marginal circularity.

Differential Entropy of Complex Random Vectors The differential entropy of a zero mean random vector $\mathbf{X} \in \mathbb{C}^N$ is given by the joint entropy

$H(\mathbf{X}_r, \mathbf{X}_i)$, and satisfies [81]:

$$H(\mathbf{X}) \leq \log[(\pi e)^N \det(\mathbf{C})] \tag{1.21}$$

with equality if, and only if, $\mathbf{X}$ is second-order circular and Gaussian with zero mean. Thus, it is a *circular* Gaussian random variable that maximizes the entropy for the complex case. It is also worthwhile to note that orthogonality and Gaussianity, together do not imply independence for complex Gaussian random variables, unless the variable is circular.

For a noncircular Gaussian random vector, we have [33, 100]

$$H_{\text{noncirc}} = \underbrace{\log[(\pi e)^N \det(\mathbf{C})]}_{H_{\text{circ}}} + \frac{1}{2}\log \prod_{n=1}^{N} (1 - \kappa_n^2)$$

where κ_n are the singular values of $\mathbf{P}$ as defined and $\kappa_n = 0$ when the random vector is circular. Hence, the circularity coefficients provide an attractive measure for quantifying circularity and a number of those measures are studied in [100]. Since $\kappa_n \leq 1$ for all n, the second term is negative for noncircular random variables decreasing the overall differential entropy as a function of the circularity coefficients.

Complex Random Processes In [8, 27, 81, 90, 91], the statistical characterization and properties of complex random processes are discussed in detail. In particular, [91] explores the strong relationship between stationarity and circularity of a random process through definitions of circularity and stationarity with varying degrees of assumptions on the properties of the process.

In our introduction to complex random processes, we focus on discrete-time processes and primarily use the notations and terminology adopted by [81] and [91]. The covariance function for a complex discrete-time random process $X(n)$ is written as

$$c(n, m) = E\{X(n)X^*(m)\} - E\{X(n)\}E\{X^*(m)\}$$

and the correlation function as $E\{X(n)X^*(m)\}$.

To completely define the second-order statistics, as in the case of random variables, we also define the pseudo-covariance function [81]—also called the complementary covariance [101] and the relation function [91]—as

$$p(n, m) = E\{X(n)X(m)\} - E\{X(n)\}E\{X(m)\}.$$

In the sequel, to simplify the expressions, we assume zero mean random processes, and hence, the covariance and correlation functions coincide.

Stationarity and Circularity Properties of Random Processes A random signal $X(n)$ is stationary if all of its statistical properties are invariant to any given time shift (translations by the origin), or alternatively, if the family of

distributions that describe the random process as a collection of random variables are all invariant to any time shift. As in the case of a random variable, the distribution for a complex random process is defined as the joint distribution of real and imaginary parts of the process.

For second-order stationarity, again we need to consider the complete characterization using the pseudo-covariance function.

A complex random process $X(n)$ is called *wide sense stationary* (WSS) if $E\{X(n)\} = m_x$, is independent of n and if

$$E\{X(n)X^*(m)\} = r(n-m)$$

and it is called *second-order stationary* (SOS) if it is WSS and in addition, its pseudo-covariance function satisfies and

$$E\{X(n)X(m)\} = p(n-m)$$

that is, it is a function of the time difference $n - m$.

Obviously, the two definitions are equivalent for real-valued signals and second-order stationarity implies WSS but the reverse is not true. In [81], second-order stationarity is identified as circular WSS and a WSS process is defined as an SOS process.

Let $X(n)$ be a second-order zero mean stationary process. Using the widely-linear transform for the scalar-valued random process $X(n)$, $\bar{\mathbf{X}}_C(n) = [X(n)\ X^*(n)]^T$ we define the spectral matrix of $\bar{\mathbf{X}}_C(n)$ as the Fourier transform of the covariance function of $\bar{\mathbf{X}}_C(n)$ [93], which is given by

$$\mathbf{C}_C(f) \triangleq \mathcal{F}\{E\{\bar{\mathbf{X}}_C(n)\bar{\mathbf{X}}_C^H(n)\}\} = \begin{bmatrix} C(f) & P(f) \\ P^*(-f) & C(-f) \end{bmatrix}$$

and where $C(f)$ and $P(f)$ denote the Fourier transforms of the covariance and pseudo-covariance functions of $X(n)$, that is, of $c(k)$ and $p(k)$ respectively.

The covariance function is nonnegative definite and the pseudo-covariance function of a SOS process is symmetric. Hence its Fourier transform also satisfies $P(f) = P(-f)$. Since, by definition, the spectral matrix $\boldsymbol{C}_C(f)$ has to be nonnegative definite, we obtain the condition

$$|P(f)|^2 \leq C(f)C(-f)$$

from the condition for nonnegative definiteness of $\boldsymbol{C}_C(f)$. The inequality also states the relationship between the power spectrum $C(f)$ and the Fourier transform of a pseudo-covariance function.

A random process is called second-order circular if its pseudo-covariance function

$$p(k) = 0, \quad \forall k$$

a condition that requires the process to be SOS.

Also, it is easy to observe that an analytic signal constructed from a WSS real signal is always second-order circular, since for an analytic signal we have $C(f) = 0$ for $f < 0$, which implies that $P(f) = 0$. An analytic signal corresponding to a nonstationary real signal is, on the other hand, in general noncircular [93].

■ EXAMPLES

In Figure 1.3, we show scatter plots of three random processes: (1) a circular complex autoregressive (AR) process driven by a circular Gaussian signal; (2) a 16 quadrature amplitude modulated (QAM) signal; and (3) a noncircular complex AR process driven by a circular Gaussian signal. The processes shown in the figure are circular, second-order circular, and noncircular respectively. The corresponding covariance and pseudo-covariance functions [$c(k)$ and $p(k)$] are shown in Figure 1.4, which demonstrate that for the first two processes, the pseudo-covariance function is zero since both are second-order circular.

Note that even though the 16-QAM signal is second-order circular, it is not circular as it is not invariant to phase rotations. A binary phase shift keying signal, on the other hand, is noncircular when interpreted as a complex signal, and since the signal is actually real valued, its covariance and pseudo-covariance

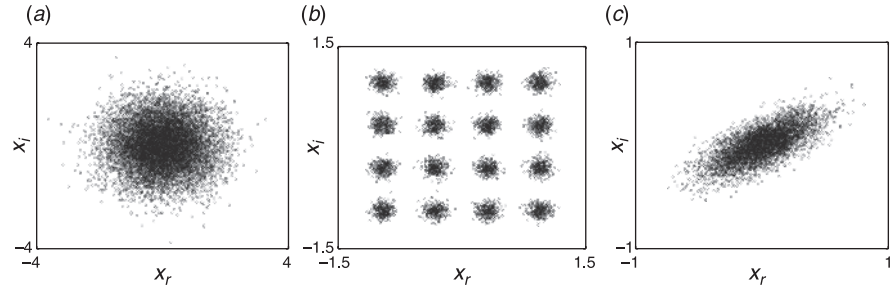

Figure 1.3 Scatter plots for a strictly (*a*) circular, (*b*) second-order circular 16-QAM, and (*c*) noncircular AR process.

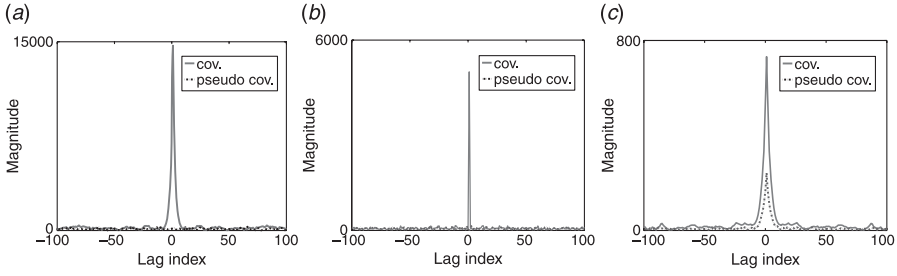

Figure 1.4 Covariance and pseudo-covariance function plots for the strictly (*a*) circular, (*b*) second-order circular 16-QAM, and (*c*) noncircular AR processes shown in Figure 1.3.

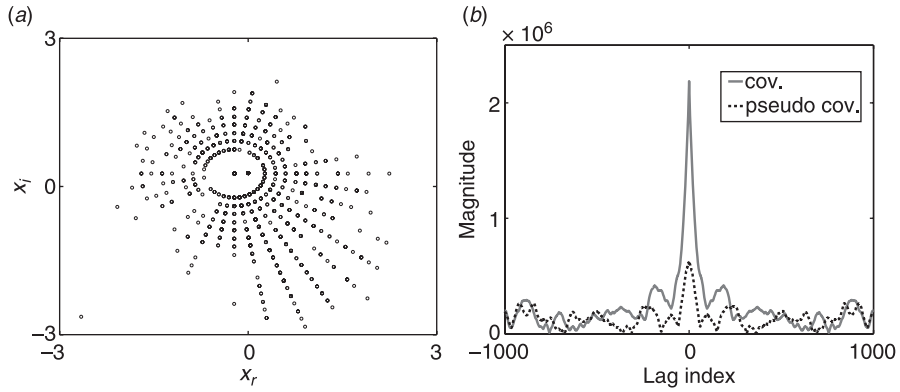

Figure 1.5 (*a*) Scatter plot and the (*b*) covariance and pseudo-covariance function plots for a sample wind data.

functions are the same. Hence, it has a non-zero pseudo-covariance function thus quantitatively verifying its noncircular nature.

In Figures 1.5 and 1.6, we show examples of real-world signals where the samples within each data set are normalized to zero mean and unit variance. The scatter plot of a sample of wind data obtained from http://mesonet.agron.iastate.edu is shown in Figure 1.5 along with its covariance and pseudo-covariance functions. The data are interpreted as complex by combining its strength as the magnitude and direction as the phase information. As observed from the scatter plot as well as its nonzero pseudo-covariance function, the signal is noncircular. Two more samples are shown in Figure 1.6. The example in Figure 1.6*a* shows

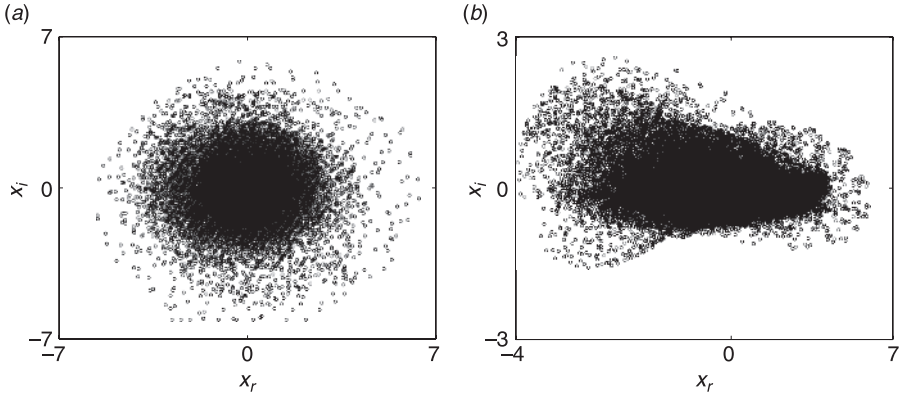

Figure 1.6 Scatter plots of (*a*) a circular (radar) data and (*b*) a noncircular (fMRI) data.

a sample Ice Multiparameter Imaging X-Band Radar (IPIX) data from the website http://soma.crl.mcmaster.ca/ipix/. As observed in the figure, the data have circular characteristics. In Figure 1.6*b*, we show the scatter plot of a functional MRI data volume. The paradigm used in the collection of the data is a simple motor task with a box-car type time-course, that is, the stimulus has periodic on and off periods. Since fMRI detects intensity changes, to evaluate the value of the fMRI signal at each voxel, we have calculated the average difference between the intensity values during the period the stimulus was "on" and "off" as a function of time. The scatter plot suggests a highly noncircular signal. The noncircular nature of fMRI data is also noted in [47] as the a large signal change in magnitude is noted as being accompanied by a corresponding change in the phase. Even though in these examples we have based the classifications on circular nature on simple visual observations, such a classification can be statistically justified by using a proper measure of noncircularity and a statistical test such as the generalized likelihood ratio test [100, 102].

As demonstrated by these examples, noncircular signals commonly arise in practice even though circularity has been a common assumption for many signal processing problems. Thus, we emphasize the importance of designing algorithms for the general case where signals may be noncircular and not to make assumptions such as circularity.

1.3 OPTIMIZATION IN THE COMPLEX DOMAIN

Most problems in signal processing involve the optimization of a real-valued cost function, which, as we noted, is not differentiable in the complex domain. Using Wirtinger calculus, however, we can relax the stringent requirement for differentiability (complex differentiability) and when the more relaxed condition of real differentiability is satisfied, can perform optimization in the complex domain in a way quite similar to the real domain. In this section, we provide the basic relationships that enable the transformation between the real and the complex domains and demonstrate how they can be used to extend basic update rules to the complex domain. We first provide a basic review of first- and second-order learning rules in the real domain and then discuss the development of appropriate tools in $\mathbb{C}^N$.

1.3.1 Basic Optimization Approaches in $\mathbb{R}^N$

Most signal processing applications use an iterative optimization procedure to determine the parameter vector $\mathbf{w}$ for a given nonlinear function $f(\mathbf{w})$: $\mathbb{R}^N \mapsto \mathbb{R}$ that cannot be directly solved for $\mathbf{w}$. We start with an initial guess for the parameter vector (weights) $\mathbf{w}(0) \in \mathbb{R}^N$ and generate a sequence of iterations for the weights as $\mathbf{w}(1), \mathbf{w}(2), \ldots, \mathbf{w}(n)$ such that the cost function $f(\mathbf{w})$ decreases (increases) until it reaches a local minimum (maximum). At each iteration n (or typically time index

for most signal processing applications), the weights are updated such that

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\,\mathbf{d}(n)$$

where μ is the stepsize and $\mathbf{d}(n)$ is the line search direction, that is, the update vector. Without loss of generality, if we consider a minimization problem, both μ and $\mathbf{d}(n)$ should be chosen such that $f[\mathbf{w}(n+1)] < f[\mathbf{w}(n)]$. In the derivation of the form of the update vector $\mathbf{d}(n)$, Taylor series expansions discussed in Section 1.2.4 play a key role.

To derive the gradient descent (also called the steepest descent) updates for the minimization of $f(\mathbf{w})$, we write the first-order Taylor series expansion of $f(\mathbf{w})$ at $\mathbf{w}(n+1)$ as

$$f[\mathbf{w}(n+1)] = f[\mathbf{w}(n)] + \langle \mu\mathbf{d}(n),\, \nabla_{\mathbf{w}(n)} f \rangle$$

where $\nabla_{\mathbf{w}(n)} f$ is the gradient vector of $f(\cdot)$ at $\mathbf{w}(n)$. The inner product between the gradient and the update vector is written as

$$\langle \mathbf{d}(n),\, \nabla_{\mathbf{w}(n)} f \rangle = \mathbf{d}^T(n) \nabla_{\mathbf{w}(n)} f = \|\mathbf{d}(n)\| \|\nabla_{\mathbf{w}(n)} f\| \cos\theta$$

where θ is the angle between the two vectors. Thus, for a fixed stepsize μ and magnitude of $\mathbf{d}(n)$, maximum decrease in $f[\mathbf{w}(n)]$ is achieved when $\mathbf{d}(n)$ and $\nabla_{\mathbf{w}(n)} f$ are in reverse directions yielding the gradient descent update rule

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \nabla_{\mathbf{w}(n)} f.$$

Newton method, on the other hand, assumes that the function can be locally approximated as a quadratic function in the region around the optimum. Thus, to derive the Newton update, we write the Taylor series expansion of $f[\mathbf{w}(n+1)]$ up to the second order as

$$\begin{aligned} f[\mathbf{w}(n+1)] &= f[\mathbf{w}(n)] + \mathbf{d}^T(n) \nabla_{\mathbf{w}(n)} f + \frac{1}{2} \mathbf{d}^T(n) \mathbf{H}[\mathbf{w}(n)] \mathbf{d}(n) \\ &= f[\mathbf{w}(n)] + \langle \nabla_{\mathbf{w}(n)} f, \mathbf{d}(n) \rangle + \frac{1}{2} \langle \mathbf{H}[\mathbf{w}(n)] \mathbf{d}(n), \mathbf{d}(n) \rangle \end{aligned}$$

where $\mathbf{H}[\mathbf{w}(n)] \triangleq \nabla^2_{\mathbf{w}(n)} f$ is the Hessian matrix of $f(\mathbf{w})$ at $\mathbf{w}(n)$ and the stepsize μ is set to 1. Setting the derivative of this expansion [with respect to $\mathbf{d}(n)$] to zero, we obtain

$$\nabla_{\mathbf{w}(n)} f + \mathbf{H}[\mathbf{w}(n)] \mathbf{d}(n) = 0 \tag{1.22}$$

as the necessary condition for the optimum function change. The optimum direction

$$\mathbf{d}(n) = -(\mathbf{H}[\mathbf{w}(n)])^{-1} \nabla_{\mathbf{w}(n)} f \tag{1.23}$$

is called the Newton direction if $\mathbf{H}[\mathbf{w}(n)]$ is nonsingular. Newton method converges quadratically to a local optimum if $\mathbf{w}(0)$ is sufficiently close to this point and if the Hessian is positive definite. However, the method faces difficulties when the quadratic approximation is not a reasonable one at the current weight update and/or the Hessian is not positive definite. Thus a number of modifications have been proposed to the Newton method, such as performing a line search along the Newton direction, rather than using the stepsize that minimizes the quadratic model assumption. More importantly, a number of procedures are introduced that use an approximate Hessian rather than the actual Hessian that allow better numerical properties. These include the Davidon–Fletcher–Powell (DFP) method and the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method [82].

Another approach is to solve (1.22) iteratively, which is desirable also when the dimensionality of the problem is high and/or the numerical properties of the Hessian are known to be poor. For the task, we can employ the well known conjugate gradient algorithm, which generates a sequence $\mathbf{d}(1), \mathbf{d}(2), \ldots, \mathbf{d}(k)$ such that $\mathbf{d}(k)$ converges to the optimal direction $-(\mathbf{H}[\mathbf{w}(n)])^{-1}\nabla_{\mathbf{w}(n)} f$.

A set of nonzero vectors $[\mathbf{c}(0), \mathbf{c}(1), \ldots, \mathbf{c}(n)]$ are said to be conjugate with respect to a symmetric positive definite matrix $\mathbf{A}$ if

$$\mathbf{c}^T(k)\mathbf{A}\mathbf{c}(l) = 0, \quad \text{for all } k \neq l$$

where, in this case $\mathbf{A} = \mathbf{H}[\mathbf{w}(n)]$.

It can be shown that for any $\mathbf{d}(0) \in \mathbb{R}^N$, the sequence $\mathbf{d}(k)$ generated by the conjugate direction algorithm as

$$\begin{aligned}
\mathbf{d}(k+1) &= \mathbf{d}(k) + \alpha_k \mathbf{c}(k) \\
\alpha(k) &= -\frac{\mathbf{q}^T(k)\mathbf{c}(k)}{\mathbf{c}^T(k)\mathbf{H}[\mathbf{w}(n)]\mathbf{c}(k)} \\
\mathbf{q}(k) &= \nabla_{\mathbf{w}(n)} f + \mathbf{H}[\mathbf{w}(n)]\mathbf{d}(k)
\end{aligned}$$

converges to the optimal solution at most N steps. The question that remains is how to construct the set of conjugate directions. Generally $\mathbf{c}(k)$ is selected to be a linear combination of $\mathbf{q}(k)$ and the previous direction $\mathbf{c}(k-1)$ as

$$\mathbf{c}(k) = -\mathbf{q}(k) + \beta(k)\mathbf{c}(k-1)$$

where

$$\beta(k) = \frac{\mathbf{q}^T(k)\mathbf{H}[\mathbf{w}(n)]\mathbf{c}(k-1)}{\mathbf{c}^T(k-1)\mathbf{H}[\mathbf{w}(n)]\mathbf{c}(k-1)}$$

is determined by the constraint that $\mathbf{c}(k)$ and $\mathbf{c}(k-1)$ must be conjugate to the Hessian matrix.

1.3.2 Vector Optimization in $\mathbb{C}^N$

Given a real-differentiable cost function $f(\mathbf{w})$: $\mathbb{C}^N \mapsto \mathbb{R}$, we can write $f(\mathbf{w}) = f(\mathbf{w}, \mathbf{w}^*)$ and take advantage of Wirtinger calculus as discussed in Section 1.2.2. The first-order Taylor series expansion of $f(\mathbf{w}, \mathbf{w}^*)$ is given by (1.18), and as discussed in Section 1.2.4, it is the gradient with respect to the *conjugate* of the variable that results in the maximum change for the complex case. Hence, the updates for gradient optimization of f is written as

$$\Delta\mathbf{w} = \mathbf{w}(n+1) - \mathbf{w}(n) = -\mu\nabla_{\mathbf{w}^*(n)} f. \tag{1.24}$$

The update given in (1.24) leads to a nonpositive increment, $\Delta f = -2\mu\|\nabla_{\mathbf{w}(n)} f\|^2$, while the update that uses $\Delta\mathbf{w} = -\mu\nabla_{\mathbf{w}(n)} f$, leads to changes of the form $\Delta f = -2\mu\mathrm{Re}\{\langle\nabla_{\mathbf{w}^*(n)} f, \nabla_{\mathbf{w}(n)} f\rangle\}$, which are not guaranteed to be nonpositive. Here, we consider only first-order corrections since μ is typically very small.

The complex gradient update rule given in (1.24) can be also derived through the relationship given in the following proposition, which provides the connection between the real-valued and the complex-valued gradients. Using the mappings defined in Table 1.2 (Section 1.2.3) and the linear transformations among them, we can extend Wirtinger derivatives to the vector case both for the first- and second-order derivatives as stated in the following proposition.

Proposition 1 *Given a function $f(\mathbf{w}, \mathbf{w}^*)$: $\mathbb{C}^N \times \mathbb{C}^N \mapsto \mathbb{R}$ that is real differentiable up to the second-order. If we write the function as $f(\bar{\mathbf{w}}_R)$: $\mathbb{R}^{2N} \mapsto \mathbb{R}$ using the definitions for $\bar{\mathbf{w}}_C$ and $\bar{\mathbf{w}}_R$ given in Table 1.2 we have*

$$\frac{\partial f}{\partial \bar{\mathbf{w}}_R} = \mathbf{U}^H \frac{\partial f}{\partial \bar{\mathbf{w}}_C^*} \tag{1.25}$$

$$\frac{\partial^2 f}{\partial \bar{\mathbf{w}}_R \partial \bar{\mathbf{w}}_R^T} = \mathbf{U}^H \frac{\partial^2 f}{\partial \bar{\mathbf{w}}_C^* \partial \bar{\mathbf{w}}_C^T} \mathbf{U} \tag{1.26}$$

where $\mathbf{U} = \begin{bmatrix} \mathbf{I} & j\mathbf{I} \\ \mathbf{I} & -j\mathbf{I} \end{bmatrix}$.

Proof 1 *Since we have $\mathbf{U}\mathbf{U}^H = 2\mathbf{I}$, $\bar{\mathbf{w}}_C = \mathbf{U}\bar{\mathbf{w}}_R$ and $\bar{\mathbf{w}}_R = \frac{1}{2}\mathbf{U}^H\bar{\mathbf{w}}_C$. We can thus write the two Wirtinger derivatives given in (1.5) in vector form as*

$$\frac{\partial f}{\partial \bar{\mathbf{w}}_C} = \frac{1}{2}\mathbf{U}^* \frac{\partial f}{\partial \bar{\mathbf{w}}_R}$$

in a single equation. Rewriting the above equality as

$$\frac{\partial f}{\partial \bar{\mathbf{w}}_R} = \mathbf{U}^T \frac{\partial f}{\partial \bar{\mathbf{w}}_C} = \mathbf{U}^H \frac{\partial f}{\partial \bar{\mathbf{w}}_C^*} \tag{1.27}$$

we obtain the first-order connection between the real and the complex gradient.

Taking the transpose of the first equality in (1.27), we have

$$\frac{\partial f}{\partial \bar{\mathbf{w}}_R^T} = \frac{\partial f}{\partial \bar{\mathbf{w}}_C^T}\mathbf{U}. \tag{1.28}$$

We regard the kth element of the two row vectors in (1.28) as two equal scalar-valued functions defined on $\bar{\mathbf{w}}_R$ *and* $\bar{\mathbf{w}}_C$*, and take their derivatives to obtain*

$$\frac{\partial\left(\dfrac{\partial f}{\partial \bar{\mathbf{w}}_R^T}\right)_k}{\partial \bar{\mathbf{w}}_R} = \mathbf{U}^T \frac{\partial\left(\dfrac{\partial f}{\partial \bar{\mathbf{w}}_C^T}\mathbf{U}\right)_k}{\partial\, \bar{\mathbf{w}}_C}.$$

We can then take the conjugate on each side and write the equality in vector form as

$$\frac{\partial^2 f}{\partial \bar{\mathbf{w}}_R \partial \bar{\mathbf{w}}_R^T} = \mathbf{U}^H \frac{\partial^2 f}{\partial \bar{\mathbf{w}}_C^* \partial \bar{\mathbf{w}}_C^T}\mathbf{U} = \mathbf{U}^T \frac{\partial^2 f}{\partial \bar{\mathbf{w}}_C \partial \bar{\mathbf{w}}_C^T}\mathbf{U}$$

to obtain the second-order relationship given in (1.26).

The second-order differential relationship for vector parameters given in (1.26) is first reported in [111] but is defined with respect to variables $\tilde{\mathbf{w}}_R$ and $\tilde{\mathbf{w}}_C$ using element-wise transforms given in Table 1.2. Using the mapping $\bar{\mathbf{w}}_C$ as we have shown here rather than the element-wise transform enables one to easily reduce the dimension of problem from C^{2N} to C^N. The second-order Taylor series expansion using the two forms ($\tilde{\mathbf{w}}_C$ and $\bar{\mathbf{w}}_C$) are the same, as expected, and we can write using either $\tilde{\mathbf{w}}_C$ or $\bar{\mathbf{w}}_C$

$$\Delta f \approx \Delta \bar{\mathbf{w}}_C^T \frac{\partial f}{\partial \bar{\mathbf{w}}_C} + \frac{1}{2}\Delta\, \bar{\mathbf{w}}_C^H \frac{\partial^2 f}{\partial \bar{\mathbf{w}}_C^* \partial \bar{\mathbf{w}}_C^T}\Delta \bar{\mathbf{w}}_C \tag{1.29}$$

as in (1.19), a form that demonstrates the fact that the $C^{2N\times 2N}$ Hessian in (1.29) can be decomposed into three $C^{N\times N}$ Hessians which are given in (1.17).

The two complex-to-real relationships given in (1.25) and (1.26) are particularly useful for the derivation of update rules in the complex domain. Next, we show their application in the derivation of the complex gradient and the complex Newton updates, and note the connection to the corresponding update rules in the real domain.

Complex Gradient Updates Given a real-differentiable function f as defined in Proposition 1, the well-known gradient update rule for $f(\bar{\mathbf{w}}_R)$ is

$$\Delta \bar{\mathbf{w}}_R = -\mu \frac{\partial f}{\partial \bar{\mathbf{w}}_R}$$

which can be mapped to the complex domain using (1.25) as

$$\Delta \bar{\mathbf{w}}_C = \mathbf{U}\Delta \bar{\mathbf{w}}_R = -\mu \mathbf{U}\frac{\partial f}{\partial \bar{\mathbf{w}}_R} = -2\mu \frac{\partial f}{\partial \bar{\mathbf{w}}_C^*}$$

The dimension of the update equation can be further decreased as

$$\begin{bmatrix} \Delta\mathbf{w} \\ \Delta\mathbf{w}^* \end{bmatrix} = -2\mu \begin{bmatrix} \dfrac{\partial f}{\partial \mathbf{w}^*} \\ \dfrac{\partial f}{\partial \mathbf{w}} \end{bmatrix} \Longrightarrow \Delta\mathbf{w} = -2\mu \frac{\partial f}{\partial \mathbf{w}^*}.$$

Complex Newton Updates

Proposition 2 *Given function $f(\cdot)$ defined in Proposition 1, Newton update in $\mathbb{R}^{2N}$ given by*

$$\frac{\partial^2 f}{\partial \bar{\mathbf{w}}_R \partial \bar{\mathbf{w}}_R^T} \Delta\bar{\mathbf{w}}_R = -\frac{\partial f}{\partial \bar{\mathbf{w}}_R} \tag{1.30}$$

is equivalent to

$$\Delta\mathbf{w} = -(\mathbf{H}_2^* - \mathbf{H}_1^* \mathbf{H}_2^{-1} \mathbf{H}_1)^{-1} \left(\frac{\partial f}{\partial \mathbf{w}^*} - \mathbf{H}_1^* \mathbf{H}_2^{-1} \frac{\partial f}{\partial \mathbf{w}} \right) \tag{1.31}$$

in $\mathbb{C}^N$, where

$$\mathbf{H}_1 \triangleq \frac{\partial^2 f}{\partial \mathbf{w} \partial \mathbf{w}^T} \quad \text{and} \quad \mathbf{H}_2 \triangleq \frac{\partial^2 f}{\partial \mathbf{w} \partial \mathbf{w}^H}. \tag{1.32}$$

Proof 2 *By using (1.25) and (1.26), the real domain Newton updates given in (1.30) can be written as*

$$\frac{\partial^2 f}{\partial \bar{\mathbf{w}}_C^* \partial \bar{\mathbf{w}}_C^T} \Delta\bar{\mathbf{w}}_C = -\frac{\partial f}{\partial \bar{\mathbf{w}}_C^*}$$

which can then put into the form

$$\begin{bmatrix} \mathbf{H}_2^* & \mathbf{H}_1^* \\ \mathbf{H}_1 & \mathbf{H}_2 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{w} \\ \Delta\mathbf{w}^* \end{bmatrix} = -\begin{bmatrix} \dfrac{\partial f}{\partial \mathbf{w}^*} \\ \dfrac{\partial f}{\partial \mathbf{w}} \end{bmatrix}$$

where $\mathbf{H}_1$ and $\mathbf{H}_2$ are defined in (1.32).

We can use the formula for the inverse of a partitioned positive definite matrix ([49], p. 472) when the nonnegative definite matrix $\dfrac{\partial^2 f}{\partial \bar{\mathbf{w}}_C^* \partial \bar{\mathbf{w}}_C^T}$ is positive definite, to write

$$\begin{bmatrix} \Delta\mathbf{w} \\ \Delta\mathbf{w}^* \end{bmatrix} = -\begin{bmatrix} \mathbf{T}^{-1} & -\mathbf{H}_2^{-*} \mathbf{H}_1^* \mathbf{T}^{-*} \\ -\mathbf{T}^{-*} \mathbf{H}_1 \mathbf{H}_2^{-*} & \mathbf{T}^{-*} \end{bmatrix} \begin{bmatrix} \dfrac{\partial f}{\partial \mathbf{w}^*} \\ \dfrac{\partial f}{\partial \mathbf{w}} \end{bmatrix}$$

where $\mathbf{T} \triangleq \mathbf{H}_2^* - \mathbf{H}_1^*\mathbf{H}_2^{-1}\mathbf{H}_1$ *and* $(\cdot)^{-*}$ *denotes* $[(\cdot)^*]^{-1}$. *Since* $\dfrac{\partial^2 f}{\partial \bar{\mathbf{w}}_C^* \partial \bar{\mathbf{w}}_C^T}$ *is Hermitian, we finally obtain the complex Newton's method given in (1.31). The expression for* $\Delta\mathbf{w}^*$ *is the conjugate of (1.31).*

In [80], it has been shown that the Newton algorithm for N complex variables cannot be written in a form similar to the real-valued case. However, as we have shown, by including the conjugate of N variables, it can be written as shown in (1.31), a form that is equivalent to the Newton method in $\mathbb{R}^{2n}$. This form is also given in [110] using the variables $\tilde{\mathbf{w}}_R$ and $\tilde{\mathbf{w}}_C$, which is shown to lead to the form given in (1.31) using the same notation in [64]. Also, a quasi-Newton update is given in [117] by setting the matrix $\mathbf{H}_1$ to a zero matrix, which might not define a descent direction for every case, as also noted in [64].

1.3.3 Matrix Optimization in $\mathbb{C}^N$

Complex Matrix Gradient Gradient of a matrix-valued variable can also be written similarly using Wirtinger calculus. For a real-differentiable $f(\mathbf{W}, \mathbf{W}^*)$: $\mathbb{C}^{N\times N} \times \mathbb{C}^{N\times N} \mapsto \mathbb{R}$, we recall the first-order Taylor series expansion given in (1.20)

$$\begin{aligned} \Delta f &\approx \left\langle \Delta\mathbf{W}, \frac{\partial f}{\partial \mathbf{W}^*} \right\rangle + \left\langle \Delta\mathbf{W}^*, \frac{\partial f}{\partial \mathbf{W}} \right\rangle \\ &= 2\mathrm{Re}\left\{ \left\langle \Delta\mathbf{W}, \frac{\partial f}{\partial \mathbf{W}^*} \right\rangle \right\} \end{aligned} \tag{1.33}$$

where $\dfrac{\partial f}{\partial \mathbf{W}}$ is an $N \times N$ matrix whose (m, n)th entry is the partial derivative of f with respect to w_{mn}. As in the vector case, the matrix gradient with respect to the conjugate $\dfrac{\partial f}{\partial \mathbf{W}^*}$ defines the direction of the maximum rate of change in f with respect to the variable $\mathbf{W}$.

Complex Relative Gradient Updates We can use the first-order Taylor series expansion to derive the relative gradient update rule [21] for complex matrix variables, which is usually directly extended to the complex case without a derivation [9, 18, 34]. To write the relative gradient rule, we consider an update of the parameter matrix $\mathbf{W}$ in the invariant form $G(\mathbf{W})\mathbf{W}$ [21]. We then write the first-order Taylor series expansion for the change of the form $G(\mathbf{W})\mathbf{W}$ as

$$\begin{aligned} \Delta f &\approx \left\langle G(\mathbf{W})\mathbf{W}, \frac{\partial f}{\partial \mathbf{W}^*} \right\rangle + \left\langle G(\mathbf{W}^*)\mathbf{W}^*, \frac{\partial f}{\partial \mathbf{W}} \right\rangle \\ &= 2\mathrm{Re}\left\{ \left\langle G(\mathbf{W}), \frac{\partial f}{\partial \mathbf{W}^*}\mathbf{W}^H \right\rangle \right\} \end{aligned}$$

to determine the quantity that maximizes the rate of change in the function. Using the Cauchy–Bunyakovskii–Schwarz inequality, it is clear that $G(\mathbf{W})$ has to be in the same direction as $\dfrac{\partial f}{\partial \mathbf{W}^*}\mathbf{W}^H$ to maximize the change. Therefore we define the complex relative gradient of $f(\cdot)$ at $\mathbf{W}$ as $\dfrac{\partial f}{\partial \mathbf{W}^*}\mathbf{W}^H$ to write the relative gradient update term as

$$\Delta\mathbf{W} = -\mu G(\mathbf{W})\mathbf{W} = -\mu \frac{\partial f}{\partial \mathbf{W}^*}\mathbf{W}^H\mathbf{W}. \tag{1.34}$$

Upon substitution of $\Delta\mathbf{W}$ into (1.33), we observe that $\Delta f = -2\mu\|(\partial f/\partial \mathbf{W}^*)\mathbf{W}^H\|_{\text{Fro}}^2$, that is, it is a nonpositive quantity, thus a proper update term.

Complex Matrix Newton Update To derive the matrix Newton update rule, we need to write the Taylor series expansion up to the second order with respect to matrix variables. However, since the variables are matrix quantities, the resulting Hessian in this case is a tensor with four indices.

The Taylor series expansion up to the second order can be written as

$$\begin{aligned}\Delta f \approx & \sum_{m,n} \frac{\partial f}{\partial w_{mn}}\, dw_{mn} + \sum_{m,n} \frac{\partial f}{\partial w_{mn}^*}\, dw_{mn}^* + \sum_{m,n}\sum_{k,l} \frac{\partial^2 f}{\partial w_{mn}\partial w_{kl}^*}\, dw_{mn}\, dw_{kl}^* \\ & + \frac{1}{2}\sum_{m,n}\sum_{k,l} \frac{\partial^2 f}{\partial w_{mn}\partial w_{kl}}\, dw_{mn}\, dw_{kl} + \frac{1}{2}\sum_{m,n}\sum_{k,l} \frac{\partial^2 f}{\partial w_{mn}^*\partial w_{kl}^*}\, dw_{mn}^*\, dw_{kl}^*.\end{aligned}$$

For the update of a single element w_{mn}, the Newton update rule is derived by taking the partial derivatives of the Taylor series expansion with respect to the differential dw_{mn} and setting it to zero

$$\frac{\partial(\Delta f)}{\partial(dw_{mn})} = \frac{\partial f}{\partial w_{mn}} + \sum_{k,l}\left(\frac{\partial^2 f}{\partial w_{mn}\partial w_{kl}}\, dw_{kl} + \frac{\partial^2 f}{\partial w_{mn}\partial w_{kl}^*}\, dw_{kl}^*\right) = 0 \tag{1.35}$$

where we have given the expression in element-wise form in order to keep the notation simple.

The solution of Newton equation in (1.35) thus yields the element-wise matrix Newton update rule for w_{mn}. In certain applications, such as independent component analysis, the Newton equation given in (1.35) can be written in a compact matrix form instead of the element-wise form given here. This point will be illustrated in Section 1.6.1 in the derivation of complex Newton updates for maximum likelihood independent component analysis.

1.3.4 Newton–Variant Updates

As we have shown in Section 1.3.2, equations (1.25) and (1.26) given in Proposition 1 play a key role in the derivation of the complex gradient and Newton update rules. Also, they can be used to extend the real-valued Newton variations that are proposed

in the literature to the complex domain such that the limitations of the Newton method can be mitigated.

Linear Conjugate Gradient (CG) Updates For the Newton's method given in (1.3.1), in order to achieve convergence, we require the search direction $\Delta\bar{\mathbf{w}}_R$ to be a descent direction when minimizing a given cost function. This is the case when the Hessian $\dfrac{\partial^2 f}{\partial\bar{\mathbf{w}}_R\partial\bar{\mathbf{w}}_R^T}$ is positive definite. However, when the Hessian is not positive definite, $\Delta\bar{\mathbf{w}}_R$ 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 (1.30) using the CG method, terminating the updates if $\Delta\bar{\mathbf{w}}_R^T\left(\dfrac{\partial^2 f}{\partial\bar{\mathbf{w}}_R\partial\bar{\mathbf{w}}_R^T}\right)\Delta\bar{\mathbf{w}}_R \leq 0$.

In general, a complex-valued function is defined in $\mathbb{C}^N$. Hence, writing it in the form $f(\mathbf{w}, \mathbf{w}^*)$ is much more straightforward than converting it to a function of the $2N$ dimensional real variable as in $f(\bar{\mathbf{w}}_R)$. Using a procedure similar to the derivation of complex gradient and Newton updates, the complex-valued CG updates can be derived using the real-valued version given in Section 1.3.1. Using (1.25) and (1.26), and defining $\mathbf{s} \triangleq \partial f/\partial\mathbf{w}^*$, the complex CG method can be derived as:

Complex Conjugate Gradient Updates

Given an initial gradient $\mathbf{s}(0)$;
Set $\mathbf{x}(0) = \mathbf{0}$, $\mathbf{c}(0) = -\mathbf{s}(0)$, k = 0;
while |s(k)| $\neq$ 0

$$\alpha(k) = \frac{\mathbf{s}^H(k)\mathbf{s}(k)}{\mathrm{Re}\{\mathbf{c}^T(k)\mathbf{H}_2\mathbf{c}^*(k) + \mathbf{c}^T(k)\mathbf{H}_1\mathbf{c}(k)\}};$$

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \alpha(k)\mathbf{c}(k);$$
$$\mathbf{s}(k+1) = \mathbf{s}(k) + \alpha(k)(\mathbf{H}_2^*\mathbf{c}(k) + \mathbf{H}_1^*\mathbf{c}^*(k));$$

$$\beta(k+1) = \frac{\mathbf{s}^H(k+1)\mathbf{s}(k+1)}{\mathbf{s}^H(k)\mathbf{s}(k)};$$

$$\mathbf{c}(k+1) = -\mathbf{s}(k+1) + \beta(k+1)\mathbf{c}(k);$$
$$k = k+1;$$
end(while)

where $\mathbf{H}_1$ and $\mathbf{H}_2$ is defined in (1.32).
The complex line search Newton-CG algorithm is given as:
for k = 0,1,2,. . .
Compute a search direction $\Delta\mathbf{w}$ by applying the complex CG update rule, starting at $\mathbf{x}(0)$= 0.
Terminate when $\mathrm{Re}\{\mathbf{c}^T(k)\mathbf{H}_2\mathbf{c}^*(k) + \mathbf{c}^T(k)\mathbf{H}_1\mathbf{c}(k)\} \leq 0$;
Set $\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\Delta\mathbf{w}$, where μ satisfies a complex Wolfe condition.
end

The complex Wolfe condition [82] can be easily obtained from the real Wolfe condition using (1.25). It should be noted that the complex CG algorithm is a linear version. It is straightforward to obtain a nonlinear version based on the linear version as shown in [82] for the real case.

Other Newton Variant Updates As shown for the derivation of complex gradient and Newton update rules, we can easily obtain complex versions of other real Newton variant methods using (1.25) and (1.26). In [70], this is demonstrated for the real-valued scaled conjugate gradient (SCG) method [79]. SCG belongs to the class of CG methods and shows superlinear convergence in many optimization problems.

When the cost function takes a least-squares form, a complex version of the Gauss–Newton algorithm can be developed as in [64]. In the Gauss–Newton algorithm, the original Hessian matrix in the Newton update is replaced with a Gauss–Newton Hessian matrix, which has better numerical properties hence providing better performance. For more general cost functions, BFGS is a popular and efficient Newton variant method [82] and can be extended to the complex domain similarly.

1.4 WIDELY LINEAR ADAPTIVE FILTERING

As discussed in Section 1.2.5, in order to completely characterize the second-order statistics of a complex random process, we need to specify both the covariance and the pseudo-covariance functions. Only when the process is circular, the covariance function is sufficient since the pseudo-covariance in this case is zero. A fundamental result in this context, introduced in [94], states that a *widely linear* filter rather than the typically used linear one provides significant advantages in minimizing the mean-square error when the traditional circularity assumptions on the data do not hold. A widely linear filter augments the data vector with the conjugate of the data, thus providing both the covariance and pseudo-covariance information for a filter designed using a second-order error criterion.

The assumption of circularity is a limiting assumption as, in practice, the real and imaginary parts of a signal typically will have correlations and/or different variances. One of the reasons for the prevalence of the circularity assumption in signal processing has been due to the inherent assumption of stationarity of signals. Since the complex envelope of a stationary signal is second-order circular [91], circularity is directly implied in this case. However many signals are not stationary, and a good number of complex-valued signals such as fMRI and wind data as shown in Section 1.2.5, do not necessarily have circular distributions. Thus, the importance of widely linear filters started to be noted and widely linear filters have been proposed for applications such as interference cancelation, demodulation, and equalization for direct sequence code-division-multiple-access systems and array receivers [23, 56, 99] implemented either in direct form, or computed adaptively using the least-mean-square (LMS) [99] or recursive least squares (RLS) algorithms [55]. Next, we present the widely linear mean-square error filter and discuss its properties, in particular when computed using LMS updates as discussed in [5]. We use the vector notation introduced in

Section 1.2.3 which allows direct extension of most main results of a linear filter to the widely linear one.

1.4.1 Linear and Widely Linear Mean-Square Error Filter

A linear filter approximates the desired sequence $d(n)$ through a linear combination of a window of input samples $x(n)$ such that the estimate of the desired sequence is

$$y(n) = \mathbf{w}^H \mathbf{x}(n)$$

where the input vector at time n is written as $\mathbf{x}(n) = [x(n)\ x(n-1)\ \cdots\ x(n-N+1)]^T$ and the filter weights as $\mathbf{w} = [w_0\ w_1\ \cdots\ w_{N-1}]^T$. The minimum mean-square error (MSE) filter is designed such that the error

$$J_L(\mathbf{w}) = E\{|e(n)|^2\} = E\{|d(n) - y(n)|^2\}$$

is minimized. To evaluate the weights $\mathbf{w}_{\text{opt}}$ given by

$$\mathbf{w}_{\text{opt}} = \arg\min_{\mathbf{w}} J_L(\mathbf{w})$$

we can directly take the derivative of the MSE with respect to $\mathbf{w}^*$ (by treating the variable $\mathbf{w}$ as a constant)

$$\begin{aligned} \frac{\partial E\{e(n)e^*(n)\}}{\partial \mathbf{w}^*} &= \frac{\partial E\{[d(n) - \mathbf{w}^H\mathbf{x}(n)][d^*(n) - \mathbf{w}^T\mathbf{x}^*(n)]\}}{\partial \mathbf{w}^*} \\ &= -E\{\mathbf{x}(n)[d^*(n) - \mathbf{w}^T\mathbf{x}^*(n)]\} \end{aligned} \tag{1.36}$$

and obtain the *complex Wiener–Hopf equation*

$$E\{\mathbf{x}(n)\mathbf{x}^H(n)\}\mathbf{w}_{\text{opt}} = E\{d^*(n)\mathbf{x}(n)\}$$

by setting (1.36) to zero. For simplicity, we assume that the input is zero mean so that the covariance and correlation functions coincide. We define the input covariance matrix $\mathbf{C} = E\{\mathbf{x}\mathbf{x}^H\}$ and the cross-covariance vector $\mathbf{p} = E\{d^*(n)\mathbf{x}\}$, to write

$$\mathbf{w}_{\text{opt}} = \mathbf{C}^{-1}\mathbf{p}$$

when the input is persistenly exciting, that is, the covariance matrix is nonsingular, which is typically the case and our assumption for the rest of the discussion in this section.

We can also compute the weight vector $\mathbf{w}$ adaptively using gradient descent updates as discussed in Section 1.3.2

$$\begin{aligned}\mathbf{w}(n+1) &= \mathbf{w}(n) - \mu \frac{\partial J_L(\mathbf{w})}{\partial \mathbf{w}^*(n)} \\ &= \mathbf{w}(n) + \mu E\{e^*(n)\mathbf{x}(n)\}\end{aligned}$$

or using stochastic gradient updates as in

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e^*(n)\mathbf{x}(n)$$

which leads to the popular least-mean-square (LMS) algorithm [113]. For both updates, $\mu > 0$ is the stepsize that determines the trade-off between the rate of convergence and the minimum error $J_L(\mathbf{w}_{\text{opt}})$.

Widely Linear MSE Filter A widely linear filter forms the estimate of $d(n)$ through the inner product

$$y_{WL}(n) = \mathbf{v}^H \bar{\mathbf{x}}(n) \tag{1.37}$$

where the weight vector $\mathbf{v} = [v_0\ v_1\ \cdots\ v_{2N-1}]^T$, that is, it has double dimension compared to the linear filter and

$$\bar{\mathbf{x}}(n) = \begin{bmatrix} \mathbf{x}(n) \\ \mathbf{x}^*(n) \end{bmatrix}$$

as defined in Table 1.2 and the MSE cost in this case is written as

$$J_{WL}(\mathbf{w}) = E\{|d(n) - y_{WL}(n)|^2\}.$$

As in the case for the linear filter, the minimum MSE optimal weight vector is the solution of

$$\frac{\partial J_{WL}(\mathbf{v})}{\partial \mathbf{v}^*} = 0$$

and results in the *widely linear* complex Wiener–Hopf equation given by

$$E\{\bar{\mathbf{x}}(n)\bar{\mathbf{x}}^H(n)\}\mathbf{v}_{\text{opt}} = E\{d^*(n)\bar{\mathbf{x}}(n)\}.$$

We can solve for the optimal weight vector as

$$\mathbf{v}_{\text{opt}} = \bar{\mathbf{C}}^{-1}\bar{\mathbf{p}}$$

where

$$\bar{\mathbf{C}} = E\{\bar{\mathbf{x}}(n)\bar{\mathbf{x}}^H(n)\} = \begin{bmatrix} \mathbf{C} & \mathbf{P} \\ \mathbf{P}^* & \mathbf{C}^* \end{bmatrix}$$

and

$$\bar{\mathbf{p}} = E\{d^*(n)\bar{\mathbf{x}}(n)\} = \begin{bmatrix} \mathbf{p} \\ \mathbf{q}^* \end{bmatrix}$$

with the definition of the pseudo-covariance matrix $\mathbf{P} = E\{\mathbf{x}(n)\mathbf{x}^T(n)\}$ and the pseudo cross covariance vector $\mathbf{q} = E\{d(n)\mathbf{x}(n)\}$ in addition to the definitions for $\mathbf{C}$ and $\mathbf{p}$ given earlier for the linear MSE filter. Matrix $\bar{\mathbf{C}}$ provides the complete second-order statistical characterization for a zero-mean complex random process and is called the augmented covariance matrix.

The minimum MSE value for the two linear models can be calculated as

$$\begin{aligned} J_{L,\min} &\triangleq J_L(\mathbf{w}_{\text{opt}}) = E\{|d(n)|^2\} - \mathbf{p}^H\mathbf{C}^{-1}\mathbf{p} \\ J_{WL,\min} &\triangleq J_{WL}(\mathbf{v}_{\text{opt}}) = E\{|d(n)|^2\} - \bar{\mathbf{p}}^H\bar{\mathbf{C}}^{-1}\bar{\mathbf{p}} \end{aligned} \tag{1.38}$$

and the difference between the two is given by [94]

$$\begin{aligned} J_{\text{diff}} &= J_{L,\min} - J_{WL,\min} \\ &= (\mathbf{q}^* - \mathbf{P}^*\mathbf{C}^{-1}\mathbf{p})^H(\mathbf{C}^* - \mathbf{P}^*\mathbf{C}^{-1}\mathbf{P})^{-1}(\mathbf{q}^* - \mathbf{P}^*\mathbf{C}^{-1}\mathbf{p}). \end{aligned} \tag{1.39}$$

Since the covariance matrix $\mathbf{C}$ is assumed to be nonsingular and thus is positive definite, the error difference J_{diff} is always nonnegative. When the joint-circularity condition is satisfied, that is, when $\mathbf{P} = \mathbf{0}$ and $\mathbf{q} = \mathbf{0}$, the performance of the two filters, the linear and the widely linear filter, coincide, and there is no gain in using a widely linear filter. It can be shown that the performance of the two filters can be equal even for cases where the input is highly noncircular (see Problems 1.6 and 1.7). However, when certain circularity properties do not hold, widely linear filters provide important advantages in terms of performance [23, 94, 101] by including the complete statistical information.

Widely Linear LMS Algorithm The widely linear LMS algorithm is written similar to the linear case as

$$\mathbf{v}(n+1) = \mathbf{v}(n) + \mu e^*(n)\bar{\mathbf{x}}(n) \tag{1.40}$$

where μ is the stepsize and $e(n) = d(n) - \mathbf{v}^H(n)\bar{\mathbf{x}}(n)$.

The study of the properties of the LMS filter, which was introduced in 1960 [114], has been an active research topic and a thorough account of these is given in [43] based on the different types of assumptions that can be invoked to simplify the analysis. With

the augmented vector notation, most of the results for the behavior of the linear LMS filter can be readily extended to the widely linear one.

The convergence of the LMS algorithm depends on the eigenvalues of the input covariance matrix, which in the case of a widely linear LMS filter, is replaced by the eigenvalues of the augmented covariance matrix. A main result in this context can be described through the natural modes of the LMS algorithm [16, 43] as follows.

Define $\boldsymbol{\varepsilon}(n)$ as the weight vector error difference $\boldsymbol{\varepsilon}(n) = \mathbf{v}(n) - \mathbf{v}_{\text{opt}}$ and let the desired response be written as

$$d(n) = \mathbf{v}_{\text{opt}}^H \bar{\mathbf{x}}(n) + e_0(n).$$

When the noise term $e_0(n)$ is strongly uncorrelated with the input, that is, uncorrelated with $x(n)$ and its conjugate, we have

$$E\{\boldsymbol{\varepsilon}(n+1)\} = (\mathbf{I} - \mu\bar{C})E\{\boldsymbol{\varepsilon}(n)\}$$

We introduce the rotated version of the weight vector error difference $\boldsymbol{\varepsilon}'(n) = \mathbf{Q}^H\boldsymbol{\varepsilon}(n)$ where $\mathbf{Q}$ is the unitary matrix composed of the eigenvectors associated with the eigenvalues of $\bar{\mathbf{C}}$, that is, we assume that the augmented covariance matrix is written through the unitary similarity transformation $\bar{\mathbf{C}} = \mathbf{Q}\bar{\Lambda}\mathbf{Q}^H$. The mean value of the natural mode $\varepsilon_k(n)$, that is, the kth element of vector $\boldsymbol{\varepsilon}'(n)$ can then be written as

$$E\{\varepsilon_k'(n)\} = \varepsilon_k'(0)(1 - \mu\bar{\lambda}_k)^n \tag{1.41}$$

where $\bar{\lambda}_k$ is the kth eigenvalue of $\bar{\mathbf{C}}$.

Thus for the convergence of LMS updates to the true solution in the mean, the step-size has to be chosen such that

$$0 < \mu < \frac{2}{\bar{\lambda}_{\max}}$$

where $\bar{\lambda}_{\max}$ is the maximum eigenvalue of the augmented covariance matrix $\bar{\mathbf{C}}$. Also, as is evident from the expression given in (1.41), small eigenvalues significantly slow down the convergence in the mean. These conclusions hold for the linear LMS filter by simply replacing the eigenvalues of $\bar{\mathbf{C}}$ by the the eigenvalues of $\mathbf{C}$, λ_ks.

A measure typically used for measuring the eigenvalue disparity of a given matrix is the condition number (or the eigenvalue spread), which is written as

$$\kappa(\mathbf{C}) = \frac{\lambda_{\max}}{\lambda_{\min}}$$

for a Hermitian matrix $\mathbf{C}$, a property satisfied by the covariance and augmented covariance matrices.

When the signal is circular, the augmented covariance matrix assumes the block diagonal form

$$\bar{\mathbf{C}}_{\text{circ}} = \begin{bmatrix} \mathbf{C} & 0 \\ 0 & \mathbf{C}^* \end{bmatrix}$$

and has eigenvalues that occur with even multiplicity. In this case, the conditioning of the augmented covariance matrix $\bar{\mathbf{C}}$ and $\mathbf{C}$ are the same. As the noncircularity of the signal increases, the values of the entries of the pseudo covariance matrix moves away from zero increasing the condition number of the augmented covariance matrix $\bar{\mathbf{C}}$, thus the advantage of using a widely linear filter for noncircular signals comes at a cost when the LMS algorithm is used when estimating the widely linear MSE solution. An update scheme such as recursive least squares algorithm [43] which is less sensitive to the eigenvalue spread can be more desirable in such cases. In the next example, we demonstrate the impact of noncircularity on the convergence of LMS algorithm using a simple input model.

■ EXAMPLE 1.5

Define a random process

$$X(n) = \sqrt{1-\rho^2}X_r(n) + j\rho X_i(n) \tag{1.42}$$

where $X_r(n)$ and $X_i(n)$ are two uncorrelated real-valued random processes, both Gaussian distributed with zero mean and unit variance. By changing the value of $\rho \in [0, 1]$, we can change the degree of noncircularity of $X(n)$ and for $\rho = 1/\sqrt{2}$, the random process $X(n)$ becomes circular. Note that since second-order circularity implies strict-sense circularity for Gaussian signals, this model lets us to generate a circular signal as well.

If we define the random vector $\mathbf{X}(n) = [X(n)X(n-1)\cdots X(n-N+1)]^T$, we can show that the covariance matrix of $\mathbf{X}(n)$ is given by $\mathbf{C} = \mathbf{I}$, and the pseudo covariance matrix as $\mathbf{P} = (1-2\rho^2)\mathbf{I}$. The eigenvalues of the augmented covariance matrix $\bar{\mathbf{C}}$ can be shown to be $2\rho^2$ and $2(1-\rho^2)$, each with multiplicity N. Hence, the condition number is given by

$$\kappa(\bar{\mathbf{C}}) = \frac{1}{\rho^2} - 1$$

if $\rho \in [0, 1/\sqrt{2}]$ and by its inverse if $\rho \in [1/\sqrt{2}, 1]$.

In Figure 1.7, we show the convergence behavior of a linear and a widely linear LMS filter with input generated using the model in (1.42) for identification of a system with coefficients $w_{\text{opt},n} = \alpha[1 + \cos(2\pi(n-3)/5] - j[1 + \cos(2\pi(n-3)/10)])$, $n = 1, \ldots, 5$, and α is chosen so that the weight norm is unity (in this case, $\alpha = 0.432$). The input signal to noise ratio is 20 dB and the step size is fixed at

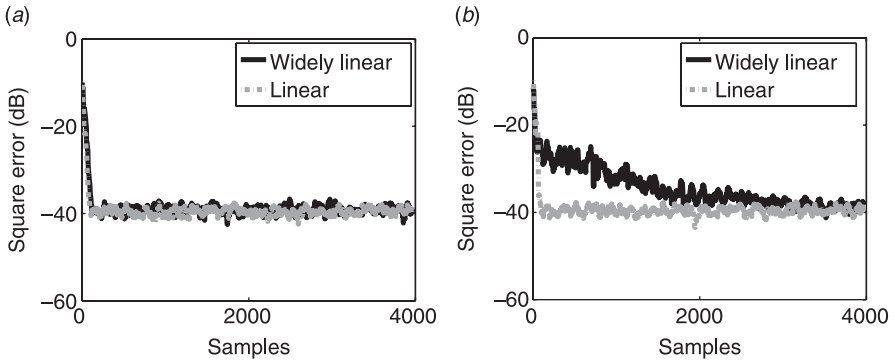

Figure 1.7 Convergence of the linear and widely linear filter for a circular input $\rho = 1\sqrt{2}$) (*a*) and a noncircular input ($\rho = 0.1$) (*b*) for a linear finite impulse response system identification problem.

$\mu = 0.04$ for all runs. In Figure 1.7*a*, we show the learning curve for a circular input, that is, $\rho = 1/\sqrt{2}$, and in Figure 1.7*b*, with a noncircular input where $\rho = 0.1$. For the first case, the condition numbers for both $\mathbf{C}$ and $\bar{\mathbf{C}}$ are approximately unity whereas for the second case, $\kappa(\mathbf{C}) \approx 1$ but $\kappa(\bar{\mathbf{C}}) \approx 100$. As expected, when the input is noncircular, the convergence rate of the widely linear LMS filter decreases. Since the lengths for the linear and widely linear filter are selected to match that of the unknown system (as 5 and 10 respectively), as discussed in Problem 6, both filters yield similar steady-state mean square error values.

In this example, even though the input is noncircular, the use of a widely linear filter does not provide an additional advantage in terms of MSE, and in addition, the convergence rate of the LMS algorithm decreases when the input is noncircular. Another observation to note for Example 1.5 is that the steady-state error variance for the widely linear filter is slightly higher compared to the linear filter. The steady-state MSE for the widely linear LMS filter can be approximated as

$$J_{WL}(\infty) = J_{WL,\min} + \frac{\mu J_{WL,\min}}{2} \sum_{k=1}^{2N} \bar{\lambda}_k$$

when the stepsize is assumed to be small. The steady-state error expression for the linear LMS filter has the same form except the very last term, which is replaced by $\sum_{k=1}^{N} \lambda_k$ where λ_k denotes the eigenvalues of $\mathbf{C}$ [43]. Since we have $\sum_{k=1}^{2N} \bar{\lambda}_k = \text{Trace}(\bar{\mathbf{C}}) = 2N\sigma^2$ and $\sum_{k=1}^{N} \lambda_k = N\sigma^2$ where $\sigma^2 = E\{|\mathbf{X}(n)|^2\}$, compared to the linear LMS filter, doubling the dimension for the widely linear filter increases the residual mean-square error compared to the linear LMS filter as expected. The difference can be eliminated by using an annealing procedure such that the step size is also adjusted such that $\mu(n) \to 0$ as $n \to \infty$.

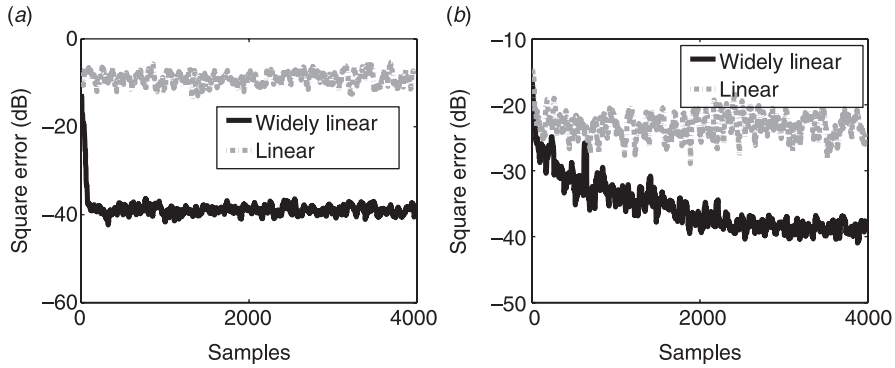

Figure 1.8 Convergence of the linear and widely linear filter for a circular input ($\rho = 1/\sqrt{2}$) (*a*) and a noncircular input ($\rho = 0.1$) (*b*) for the identification of a widely linear system.

■ **EXAMPLE 1.6**

In Figure 1.8, we show the learning curves for the linear and widely linear LMS filters for a widely linear channel. All the settings for the simulation are the same as those in Example 1.5 except that the unknown system output is given by

$$d(n) = \mathrm{Re}\{\mathbf{w}_{\mathrm{opt}}^H \mathbf{x}(n)\}$$

and the filter coefficients $w_{\mathrm{opt},n}$ are selected as before.

As observed in the figures, for both the circular and noncircular cases, the widely linear filter provides smaller MSEs, though its convergence is again slower for the noncircular input due to the increased eigenvalue spread.

An interesting point to note in Example 1.6 is that the advantage of using a widely linear filter—in terms of the minimum MSE that is achieved—is more pronounced in this case for circular input, even though the advantages of widely linear filters are, in general, emphasized for noncircular statistics.

For a circular input, the MSE gain by using a widely linear filter given in (1.39) reduces to

$$J_{\mathrm{diff}} = \|\mathbf{q}\|^2 = \|E\{d(n)\,\mathbf{x}(n)\}\|^2$$

and is clearly nonzero for the widely linear system chosen in this example, as observed in Figure 1.8 resulting in significant performance gain with the widely linear filter.

1.5 NONLINEAR ADAPTIVE FILTERING WITH MULTILAYER PERCEPTRONS

Neural network structures such as multilayer perceptron (MLP) and the radial basis function (RBF) filters have been successfully used for adaptive signal processing in

the real domain for problems that require nonlinear signal processing capability [42]. Both the MLP and the RBF filters are shown to be universal approximators of any smooth nonlinear mapping [30, 35, 51] and their use has been extended to the complex domain, see for example [12, 14, 67, 108].

A main issue in the implementation of nonlinear filters in the complex domain has been the choice of the activation function. Primarily due to stability considerations, the importance of boundedness has been emphasized, and identified as a property an activation function should satisfy for use in a complex MLP [36, 119]. Thus, the typical practice has been the use of split-type activation functions, which are defined in Section 1.2.1. Fully-complex activation functions, as we discuss next, are more efficient in approximating nonlinear functions, and can be shown to be universal approximators as well. In addition, when a fully-complex nonlinear function is used as the activation function, it enables the use of Wirtinger calculus so that derivations for the learning rules for the MLP filter can be carried out in a manner very similar to the real-valued case, making many efficient learning procedures developed for the real-valued case readily accessible in the complex domain. These results can be extended to RBF filters in a similar manner.

1.5.1 Choice of Activation Function for the MLP Filter

As noted in Section 1.2.2, Liouville's theorem states the conflict between the boundedness and differentiability of functions in the complex domain. For example, the tanh nonlinearity that has been the most typically used activation function for real-valued MLPs, has periodic singular points as shown in Figure 1.13.

Since boundedness is deemed as important for the stability of algorithms, a practical solution when designing MLP filters for the complex domain has been to define nonlinear functions that process the real and imaginary parts separately through bounded real-valued nonlinearities as defined in Section 1.2.1 and given for the typically employed function tanh as

$$f(z) \triangleq \tanh(x) + j \tanh(y) \tag{1.43}$$

for a complex variable $z = x + jy$ where tanh: $\mathbb{R} \mapsto \mathbb{R}$. The activation function can also be defined through real-valued functions defined for the magnitude and phase of z as introduced in [45]

$$f(z) = f(re^{j\theta}) \triangleq \tanh\left(\frac{r}{m}\right)e^{j\theta} \tag{1.44}$$

where m is any number different than 0. Another such activation function is proposed in [36]

$$f(z) \triangleq \frac{z}{c + |z|/d} \tag{1.45}$$

where again c and d are arbitrary constants with $d \neq 0$. The characteristics of the activation functions given in (1.43)–(1.45) are shown in Figures 1.9–1.11. As observed

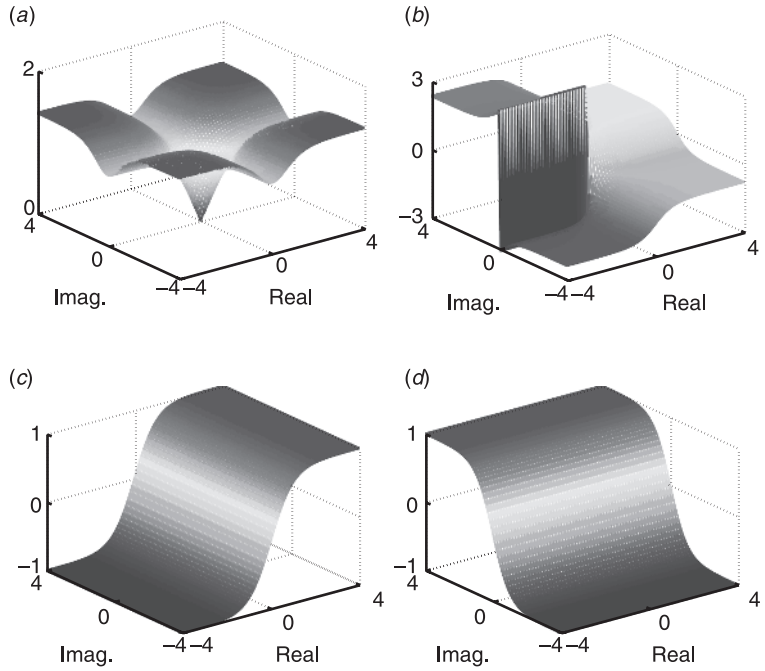

Figure 1.9 (*a*) Magnitude, (*b*) phase, (*c*) real and (*d*) imaginary responses of the split tanh function given in (1.43).

in the figures, though bounded, none of these functions can provide sufficient discrimination ability, the split function in (1.43) shown in Figure 1.9 provides a decoupled real and imaginary response while those shown in Figures 1.10 and 1.11, provide smooth radially symmetric magnitude responses and a phase response that is simply linear. The responses of the real and imaginary parts for these two functions are the same as in the case for the split function. In [61–63] examples in system identification and channel equation are provided to show that these functions cannot use the

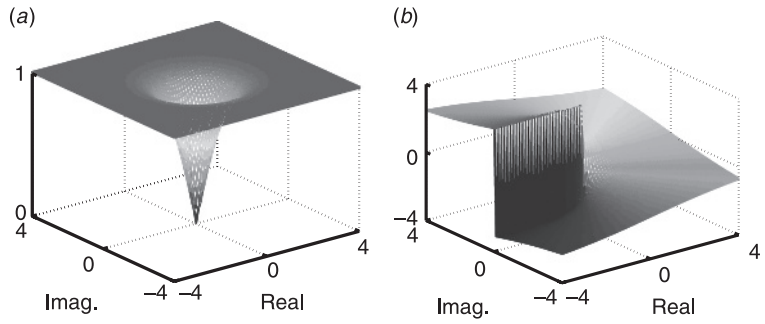

Figure 1.10 (*a*) Magnitude and (*b*) phase responses of the activation function given in (1.44) for $m = 1$.

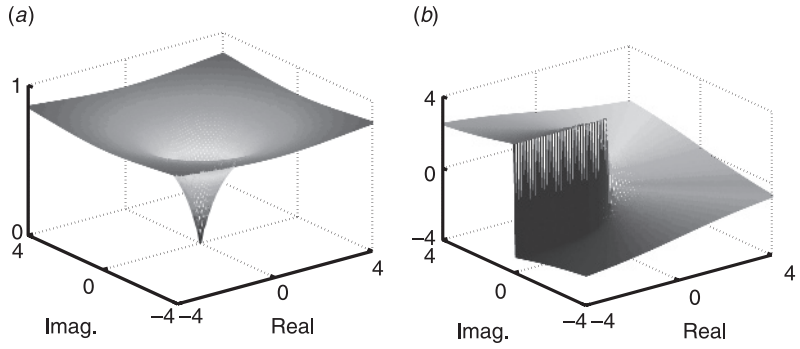

Figure 1.11 (*a*) Magnitude and (*b*) phase responses of the activation function given in (1.45) for $c = d = 1$.

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. Fully complex activation functions, or more simply, complex analytic functions, on the other hand provide a much more powerful modeling ability compared to split functions. It is worth noting that the universal approximation ability of MLPs that employ split activation functions as given in (1.43) can be easily shown by simply extending the universal approximation theorem from the real domain to the complex one [10]. However, as we demonstrate in this section, they cannot make efficient use of the available information.

In [63], a number of fully-complex—or simply analytic functions are proposed as activation functions and it is shown by a number of recent examples that MLPs using these activation functions provide a more powerful modeling ability compared to split functions [38, 39, 41, 61–63]. These functions all have well-defined first-order derivatives and squashing type characteristics that are generally required for nonlinear filters to be used as global approximators, such as the MLPs. These functions can be divided into four classes as

- Circular functions:

$$\tan z = \frac{e^{jz} - e^{-jz}}{j(e^{jz} - e^{-jz})} \qquad \frac{d}{dz}\tan z = \sec^2 z$$

$$\sin z = \frac{e^{jz} - e^{-jz}}{2j} \qquad \frac{d}{dz}\sin z = \cos z.$$

- Inverse circular functions:

$$\operatorname{atan} z = \int_0^z \frac{dt}{1+t^2} \qquad \frac{d}{dz}\operatorname{atan} z = \frac{1}{1+z^2}$$

$$\text{asin } z = \int_0^z \frac{dt}{(1-t)^{1/2}} \quad \frac{d}{dz}\text{asin } z = (1-z^2)^{-1/2}$$

$$\text{acos } z = \int_0^z \frac{dt}{(1-t^2)^{1/2}} \quad \frac{d}{dz}\text{acos } z = -(1-z^2)^{-1/2}.$$

- Hyperbolic functions:

$$\tanh z = \frac{\sinh z}{\cosh z} = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad \frac{d}{dz}\tanh z = \text{sech}^2 z$$

$$\sinh z = \frac{e^z - e^{-z}}{2} \qquad \frac{d}{dz}\sinh z = \cosh z.$$

- Inverse hyperbolic functions:

$$\text{a}\tanh z = \int_0^z \frac{dt}{1-t^2} \qquad \frac{d}{dz}\,\text{a}\tanh z = \frac{1}{1-z^2}$$

$$\text{a}\sinh z = \int_0^z \frac{dt}{(1+t^2)^{1/2}} \quad \frac{d}{dz}\text{a}\sinh z = \frac{1}{1+z^2}.$$

The magnitude and phase characteristics of these functions are shown in Figures 1.12–1.16, and for the case of the atanh function, also the responses of real and imaginary parts are shown to emphasize the variability of the responses of these functions for real and imaginary parts when compared to the split type functions shown in Figures 1.9–1.11. Note that hyperbolic functions and their trigonometric counterparts (*e.g.*, asinh and asin) have very similar responses except that they are $\pi/2$ rotated versions of each other.

In [63], three types of approximation theorems are given for MLP networks that use complex activation functions as those listed above from the trigonometric and hyperbolic family. The theorems are based on type of singularity a function possesses, as discussed in Section 1.2.2. The approximation theorems for the first two classes of functions are very general and resemble the universal approximation theorem for the real-valued feedforward MLP whereas the third approximation theorem for the complex MLP is unique in that it is uniform only in the analytic domain of convergence. As in the real case, the structure of the MLP network is a single hidden layer network as shown in Figure 1.17 with nonlinear activation functions in the hidden layer and a linear output layer. The three approximation theorems are given as [63]:

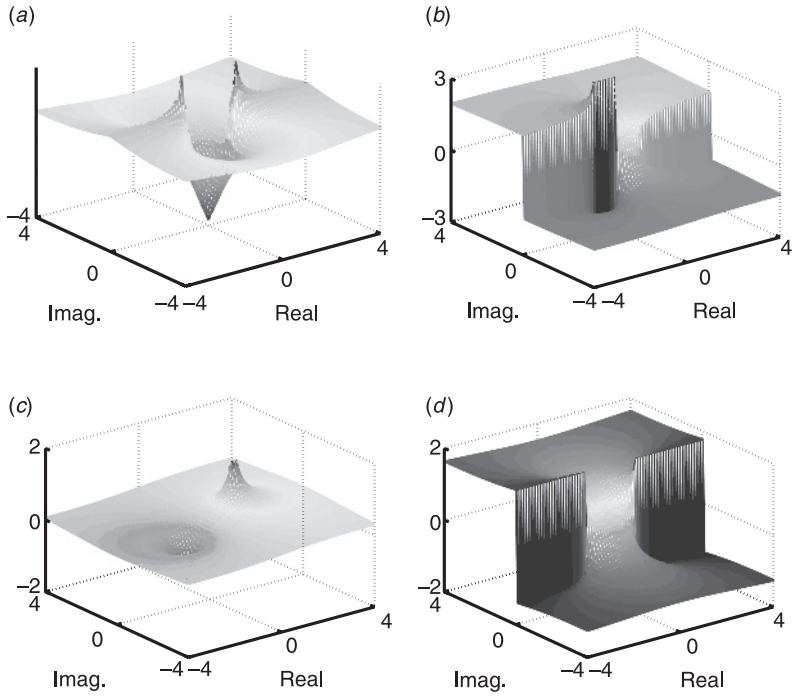

Figure 1.12 (*a*) Magnitude, (*b*) phase, (*c*) real and (*d*) imaginary responses of atanh.

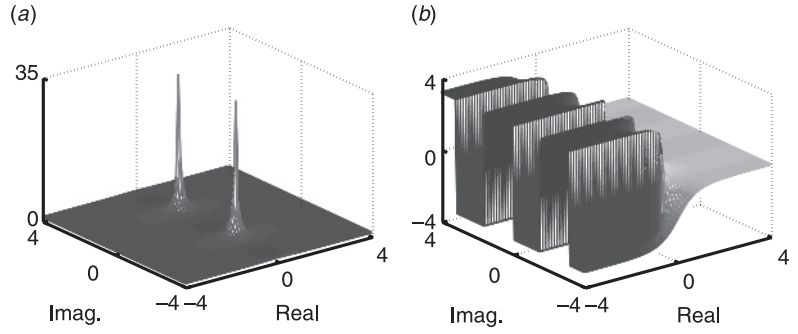

Figure 1.13 (*a*) Magnitude and (*b*) phase responses of tanh.

- MLPs that use continuous nonlinear activation functions without any singular points can achieve universal approximation of any continuous nonlinear mapping over a compact set in $\mathbb{C}^N$. Note that these functions are not bounded, as shown for the sinh function in Figure 1.16, but by bounding the region of interest using scaling, for example, for range around the unit circle for the sinh function, they can be used as activation functions and can provide good approximation as demonstrated in [63].

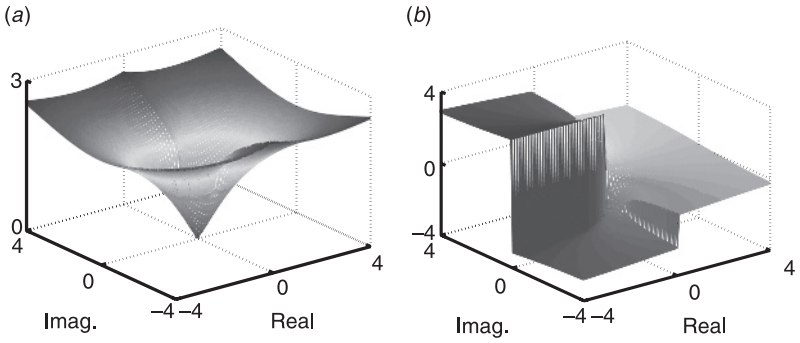

Figure 1.14 (*a*) Magnitude and (*b*) phase responses of asinh.

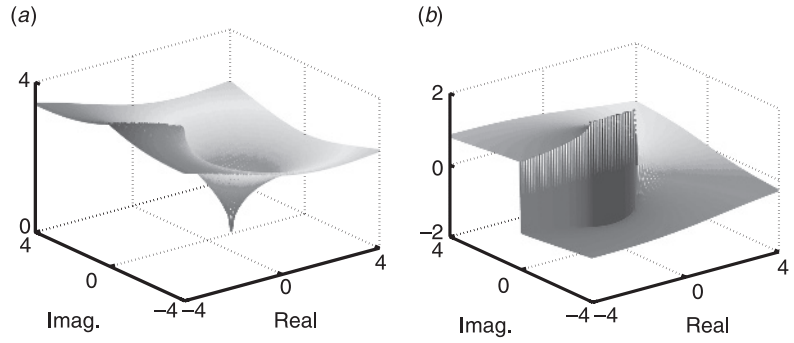

Figure 1.15 (*a*) Magnitude and (*b*) phase responses of acosh.

- The second group of functions considered are those with bounded singularities, such as branch cuts over a bounded domain and removable singularities. Examples include the asinh and acosh functions—and their trigonometric counterparts—which are all bounded complex measurable functions. It is shown that MLP using activation functions with bounded singularities provides universal approximation *almost everywhere* over a compact set in $\mathbb{C}^N$.
- Finally, the third theorem considers unbounded measurable activation functions, that is, those with poles, such as tanh and atanh and their trigonometric counterparts, as well as non-measurable nonlinear activation functions, those with essential singularity. For MLPs that use these activation functions, the approximation of any nonlinear mapping is uniform over the deleted annulus of singularity nearest to the origin. If there are multiple singularities, the radius of convergence is the shortest distance to a singularity from the origin.

Hence, complex functions such as trigonometric and hyperbolic functions can be effectively used as activation functions, and when the MLP structure involves

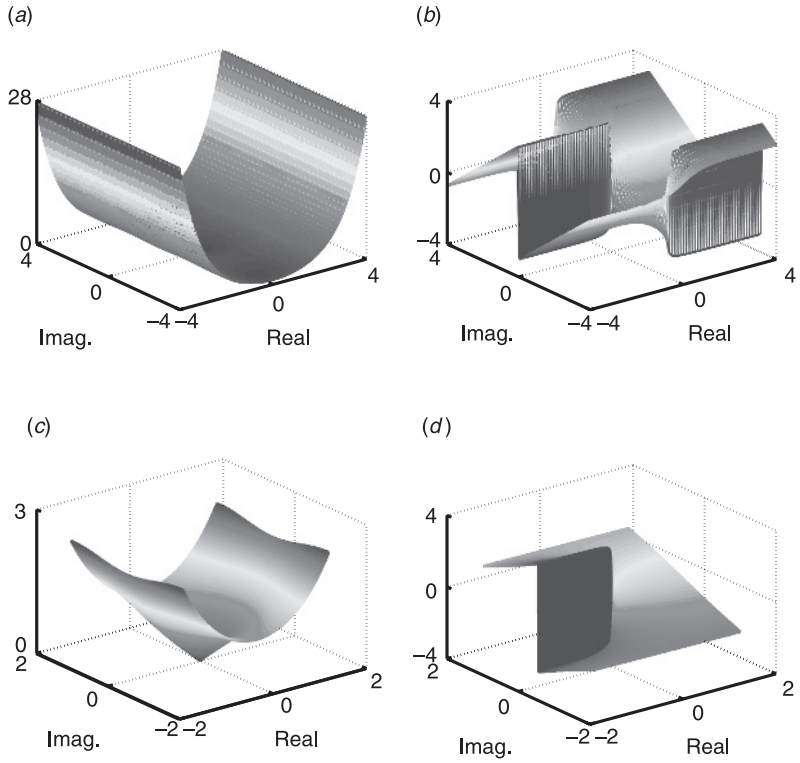

Figure 1.16 (*a, c*) Magnitude and (*b, d*) phase responses of sinh in two distinct ranges.

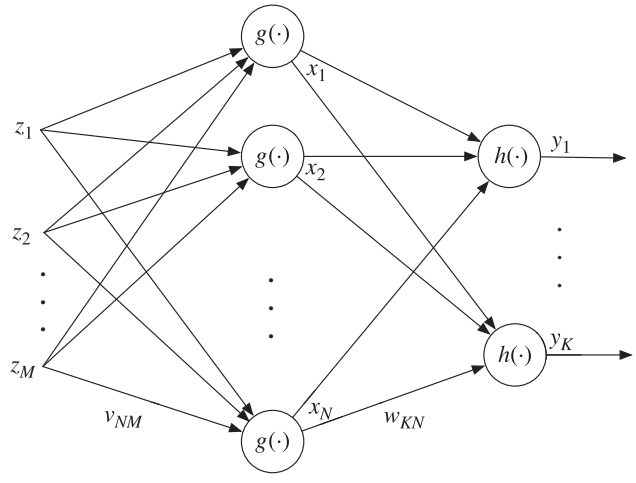

Figure 1.17 A single hidden layer (*M-N-K*) MLP filter.

such nonlinearities rather than the split type functions given in (1.43)–(1.45), the update rules for the MLP can be derived in a manner very similar to the real case as we demonstrate next for the derivation of the back-propagation algorithm.

1.5.2 Derivation of Back-Propagation Updates

For the MLP filter shown in Figure 1.17, we write the square error cost function as

$$J(\mathbf{V}, \mathbf{W}) = \sum_{k=1}^{K} (d_k - y_k)(d_k^* - y_k^*)$$

where

$$y_k = h\left(\sum_{n=1}^{N} w_{kn} x_n\right)$$

and

$$x_n = g\left(\sum_{m=1}^{M} v_{nm} z_m\right).$$

When both activation functions $h(\cdot)$ and $g(\cdot)$ satisfy the property $[f(z)]^* = f(z^*)$, then the cost function can be written as $J(\mathbf{V}, \mathbf{W}) = G(z)G(z^*)$ making it very practical to evaluate the gradients using Wirtinger calculus by treating the two variables z and z^* as independent in the computation of the derivatives. 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 $[f(z)]^* = f(z^*)$ as noted in [6] and [71].

Examples of such functions include polynomials and most trigonometric functions and their hyperbolic counterparts (all of the functions whose characteristics are shown in Figs. 1.12–1.16), which also provide universal approximation ability as discussed in Section 1.5.1. In addition, the activation functions given in (1.43)–(1.45) that process the real and imaginary or the magnitude and phase of the signals separately also satisfy this property. Hence, there is no real reason to evaluate the gradients through separate real and imaginary part computations as traditionally done. Indeed, this approach can easily get quite cumbersome as evidenced by [12, 14, 39, 41, 62, 67, 107, 108, 118] as well as a recent book [75] where the development using Wirtinger calculus is presented as an afterthought, with the result in [6] and [71] that enables the use of Wirtinger calculus given without proper citation.

When the fully-complex functions introduced in Section 1.5.1 are used as activation functions as opposed to those given in (1.43)–(1.45), the MLP filter can achieve significantly better performance in challenging signal processing problems such as equalization of highly nonlinear channels [61, 62] both 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.

For the MLP filter shown in Figure 1.17, where y_k is the output and z_m the input, when the activations functions $g(\cdot)$ and $h(\cdot)$ are chosen as functions that are $\mathbb{C} \mapsto \mathbb{C}$, we can directly write the back-propagation update equations using Wirtinger derivatives as shown next.

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

$$\begin{aligned}
\frac{\partial J}{\partial w_{kn}^*} &= \frac{\partial J}{\partial y_k^*} \frac{\partial y_k^*}{\partial w_{kn}^*} \\
&= \frac{\partial [(d_k - y_k)(d_k^* - y_k^*)]}{\partial y_k^*} \frac{\partial h\left(\sum_n w_{kn}^* x_n^*\right)}{\partial w_{kn}^*} \\
&= -(d_k - y_k) h'\left(\sum_n w_{kn}^* x_n^*\right) x_n^*. \qquad (1.46)
\end{aligned}$$

We define $\delta_k = -(d_k - y_k) h'\left(\sum_n w_{kn}^* x_n^*\right)$ 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

$$\begin{aligned}
\frac{\partial J}{\partial v_{nm}^*} &= \sum_k \frac{\partial J}{\partial y_k^*} \frac{\partial y_k^*}{\partial x_n^*} \frac{\partial x_n^*}{\partial v_{nm}^*} \\
&= \frac{\partial x_n^*}{\partial v_{nm}^*} \sum_k \frac{\partial J}{\partial y_k^*} \frac{\partial y_k^*}{\partial x_n^*} \\
&= g'\left(\sum_m v_{nm}^* z_m^*\right) z_m^* \sum_k \frac{\partial J}{\partial y_k^*} \frac{\partial y_k^*}{\partial x_n^*} \\
&= g'\left(\sum_m v_{nm}^* z_m^*\right) z_m^* \left(\sum_k -(d_k - y_k) h'\left(\sum_l w_{kl}^* x_l^*\right) w_{kn}^*\right) \\
&= z_m^* g'\left(\sum_m v_{nm}^* z_m^*\right) \left(\sum_k \delta_k w_{kn}^*\right). \qquad (1.47)
\end{aligned}$$

Thus, (1.46) and (1.47) define the gradient updates for computing the hidden and the output layer coefficients, w_{kn} and v_{nm}, through back-propagation. Note that the derivations in this case are very similar to the real-valued case as opposed to the derivations given in [12, 62, 67], where separate evaluations with respect to the real and imaginary parts are carried out, and hence the steps in the derivations assume more complicated forms. Also, the derivations for new learning rules such as those

given in [38, 39, 41] can be considerably simplified by using Wirtinger derivatives. As we demonstrate in the next example, the split activation functions are not efficient in their use of the information when learning nonlinear mappings, and hence are not desirable for use as activation functions.

■ EXAMPLE 1.7

In Figure 1.18, we show the convergence characteristics of two MLP filters, one using split tanh and a second one that uses the complex tanh as the activation function. The input is generated using the same model as in Example 1.5 with $\rho = 1/\sqrt{2}$ and the nonlinear output of the system is given as $d(n) + 0.2d^2(n)$ where $d(n) = \mathbf{w}_{\text{opt}}^H \mathbf{x}(n)$ with the coefficients w_{opt} selected as in Example 1.5. The size of the input layer is chosen as 5 to match the memory of the finite impulse response component of the system, and the filter has a single output. The stepsize is chosen as 0.01 for both the split and the fully complex MLP filters and the convergence behavior is shown for two different filter sizes, one with a filter using 15 hidden nodes and a second one with 40 hidden nodes. As observed in the figures, the MLP filter using a fully complex activation function produces lower squared error value, however the performance advantage of the fully complex filter decreases when the number of hidden nodes increases as observed in Figure 1.18*b*.

The example demonstrates that the fully complex MLP filter is able to use information more efficiently, however, the performance of the filter that uses a split-type activation function can be improved by increasing the filter complexity. Note that the universal approximation of MLP filters can be demonstrated for both filter types, and the approximation result guarantees that the MLP structure can come

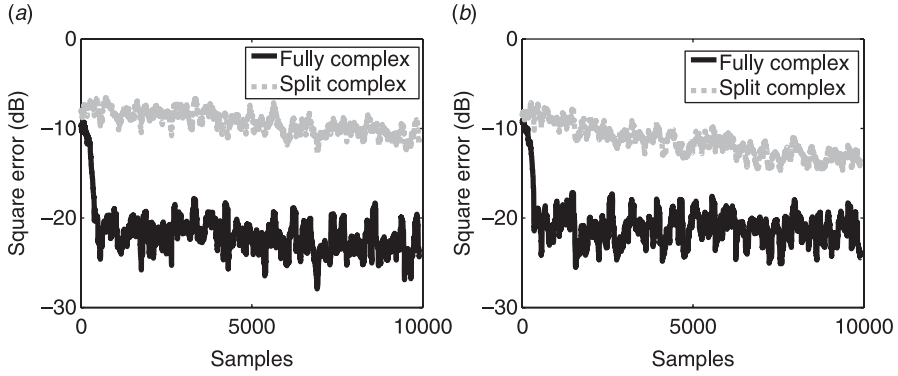

Figure 1.18 Performance of two (*a*) 5-15-1 and (*b*) 5-40-1 MLP filters for a nonlinear system identification problem using split and fully-complex tanh activation functions.

arbitrarily close to approximating any given mapping (subject to regularity conditions) *if* the number of hidden nodes chosen is sufficiently large.

The results recently given in the literature using fully complex activation functions suggest that they are promising solutions for challenging nonlinear signal processing problems, and derivation of new learning rules as well as design or selection of such activation functions is thus a research direction that deserves attention.

1.6 COMPLEX INDEPENDENT COMPONENT ANALYSIS

Independent component analysis (ICA) has emerged as an attractive analysis tool for discovering hidden factors in observed data and has been successfully applied to numerous signal processing problems in areas as diverse as biomedicine, communications, finance, and remote sensing [54]. In order to perform ICA of complex-valued data, there are a number of options. Algorithms such as joint approximate diagonalization of eigenmatrices (JADE) [22] or those using second order statistics [32, 65] achieve ICA without the need to use nonlinear functions in the algorithm. The second-order complex blind source separation algorithm, strongly uncorrelating transform (SUT) [32], though efficient, requires the sources to be noncircular and have distinct spectral coefficients. Thus a second ICA algorithm should be utilized after its application as a preprocessing step when the sources happen to be circular [34]. JADE is based on the joint diagonalization of cumulant matrices and is quite robust, however, its performance suffers as the number of sources increases, and the cost of computing and diagonalizing cumulant matrices becomes prohibitive for separating a large number of sources (see *e.g.*, [69]). On the other hand, ICA approaches that use nonlinear functions, such as maximum likelihood [89], information-maximization (Infomax) [11], nonlinear decorrelations [25, 58], and maximization of non-Gaussianity (*e.g.*, the FastICA algorithm) [53], are all intimately related to each other and present an attractive alternative for performing ICA. A number of comparison studies have demonstrated their desirable performance over other ICA algorithms such as JADE and second-order algorithms. For example, in [29], this efficiency is observed for the ICA of fMRI and fMRI-like data.

In the development of complex ICA algorithms with nonlinear functions, traditionally the same approach discussed for MLPs in Section 1.5 has been followed and a number of limitations have been imposed on the nature of complex sources either directly, or indirectly through the selection of the nonlinear function. A number of algorithms have used complex split nonlinear functions such that the real and imaginary parts (or the magnitude and phase) of the argument are processed separately through real-valued nonlinear functions [96, 103]. Another approach processes the magnitude of the argument by a real-valued function [9, 13], thus limiting the algorithm to circular sources. These approaches, while yielding satisfactory performance for a class of problems, are not effective in generating the higher order statistics required to establish independence for all possible source distributions.

In this section based on the work in [6], we concentrate on the complex ICA approaches that use nonlinear functions without imposing any limitations on the type of source distribution and demonstrate how Wirtinger calculus can be used for efficient derivation of algorithms and for working with probabilistic characterizations, which is important in the development of density matching mechanisms that play a key role in this class of ICA algorithms. We present the two main approaches for performing ICA: maximum likelihood (ML) and maximization of non-Gaussianity (MN). We discuss their relationship to each other and to other closely related ICA approaches, and in particular note the importance of source density matching for both approaches. We present extensions of source density matching mechanisms for the complex case, and note a few key points for special classes of sources, such as Gaussian sources, and those that are strictly second-order circular. We present examples that clearly demonstrate the performance equivalence of ML- and MN-based ICA algorithms when exact source matching is used for both cases.

In the development, we consider the traditional ICA problem such that

$$\mathbf{x} = \mathbf{A}\mathbf{s}$$

where $\mathbf{x}, \mathbf{s} \in \mathbb{C}^N$ and $\mathbf{A} \in \mathbb{C}^{N \times N}$, that is, the number of sources and observations are equal and all variables are complex valued.

The sources s_i where $\mathbf{s} = [s_1, \ldots, s_N]^T$ are assumed to be statistically independent and the source estimates u_i where $\mathbf{u} = [u_1, \ldots, u_N]^T$, are given by $\mathbf{u} = \mathbf{W}\mathbf{x}$. If the mixtures are whitened and sources are assumed to have unit variance, $\mathbf{WA}$ approximates a permutation matrix when the ICA problem is solved, where we assume that the mixing matrix is full rank. For the complex case, an additional component of the scaling ambiguity is the phase of the sources since all variables are assumed to be complex valued. In the case of perfect separation, the permutation matrix will have one nonzero element. Separability in the complex case is guaranteed as long as the mixing matrix $\mathbf{A}$ is of full column rank and there are no two complex Gaussian sources with the same circularity coefficient [33], where the circularity coefficients are defined as the singular values of the pseudo-covariance matrix of the source random vector. This is similar to the real-valued case where second-order algorithms that exploit the correlation structure in the mixtures use joint diagonalization of two covariance matrices [106].

1.6.1 Complex Maximum Likelihood

As in the case of numerous estimation problems, maximum likelihood theory provides a natural formulation for the ICA problem. For T independent samples $\mathbf{x}(t) \in \mathbb{C}^N$, we can write the log-likelihood function as

$$\mathcal{L}(\mathbf{W}) = \sum_{t=1}^{T} \ell_t(\mathbf{W}),$$

where

$$\ell_t(\mathbf{W}) = \log p(\mathbf{x}(t)|\mathbf{W}) = \log p_S(\mathbf{W}\mathbf{x}) + \log|\det \overline{\mathbf{W}}|$$

and the density of the transformed random variables is written through the computation of the Jacobian as

$$p(\mathbf{x}) = |\det \overline{\mathbf{W}}| p_S(\mathbf{W}\mathbf{x}) \tag{1.48}$$

where $\overline{\mathbf{W}}$ is defined in (1.11).

We use the notation that $p_S(\mathbf{W}\mathbf{x}) \triangleq \prod_{n=1}^{N} p_{S_n}(\mathbf{w}_n^H \mathbf{x})$, where $\mathbf{w}_n$ is the nth row of $\mathbf{W}$, $p_{S_n}(u_n) \triangleq p_{S_n}(u_{n_r}, u_{n_i})$ is the joint pdf of source n, $n = 1, \ldots, N$, with $u_n = u_{n_r} + j u_{n_i}$, and defined $\mathbf{W} = \mathbf{A}^{-1}$, that is, we express the likelihood in terms of the inverse mixing matrix, which provides a convenient change of parameter. Note that the time index in $\mathbf{x}(t)$ has been omitted in the expressions for simplicity.

To take advantage of Wirtinger calculus, we write each pdf as $p_{S_n}(u_r, u_i) = g_n(u, u^*)$ to define $g(\mathbf{u}, \mathbf{u}^*)$: $\mathbb{C}^N \times \mathbb{C}^N \mapsto \mathbb{R}^N$ so that we can directly evaluate

$$\frac{\partial \log g(\mathbf{u}, \mathbf{u}^*)}{\partial \mathbf{W}^*} = \frac{\partial \log g(\mathbf{u}, \mathbf{u}^*)}{\partial \mathbf{u}^*} \mathbf{x}^H \triangleq -\psi(\mathbf{u}, \mathbf{u}^*)\mathbf{x}^H \tag{1.49}$$

where $\mathbf{u} = \mathbf{W}\mathbf{x}$ and we have defined the score function $\psi(\mathbf{u}, \mathbf{u}^*)$ that is written directly by using the result in Brandwood's theorem given by (1.5)

$$\psi(\mathbf{u}, \mathbf{u}^*) = -\frac{1}{2}\left(\frac{\partial \log p_S(\mathbf{u}_r, \mathbf{u}_i)}{\partial \mathbf{u}_r} + j\frac{\partial \log p_S(\mathbf{u}_r, \mathbf{u}_i)}{\partial \mathbf{u}_i}\right). \tag{1.50}$$

When writing (1.49) and (1.50), we used a compact vector notation where each element of the score function is given by

$$\psi_n(u, u^*) = -\frac{\partial \log g_n(u_n, u_n^*)}{\partial u_n^*} = -\frac{1}{2}\left(\frac{\partial \log p_{S_n}(u_{r,n}, u_{i,n})}{\partial u_{r,n}} + j\frac{\partial \log p_{S_n}(u_{r,n}, u_{i,n})}{\partial u_{i,n}}\right). \tag{1.51}$$

To compute $\partial \log|\det \overline{\mathbf{W}}|/\partial \mathbf{W}$, we first observe that $\partial \log|\det \overline{\mathbf{W}}| = \mathrm{Trace}(\overline{\mathbf{W}}^{-1} \partial \overline{\mathbf{W}}) = \mathrm{Trace}(\partial \overline{\mathbf{W}} \mathbf{P}\mathbf{P}^{-1} \overline{\mathbf{W}}^{-1})$, and then choose

$$\mathbf{P} = \frac{1}{2}\begin{bmatrix} \mathbf{I} & j\mathbf{I} \\ j\mathbf{I} & \mathbf{I} \end{bmatrix}$$

to write

$$\begin{aligned} \partial \log|\det \overline{\mathbf{W}}| &= \mathrm{Trace}(\mathbf{W}^{-1}\partial \mathbf{W}) + \mathrm{Trace}((\mathbf{W}^*)^{-1}\partial \mathbf{W}^*) \\ &= \langle \mathbf{W}^{-H}, \partial \mathbf{W}\rangle + \langle \mathbf{W}^{-T}, \partial \mathbf{W}^*\rangle. \end{aligned} \tag{1.52}$$

Here, we have used

$$\mathbf{P}^{-1}\,\overline{\mathbf{W}}^{-1} = \frac{1}{2}\begin{bmatrix} \mathbf{W}^* & j\mathbf{W} \\ j\mathbf{W}^* & \mathbf{W} \end{bmatrix}^{-1} = \begin{bmatrix} (\mathbf{W}^*)^{-1} & -j(\mathbf{W}^*)^{-1} \\ -j\mathbf{W}^{-1} & \mathbf{W}^{-1} \end{bmatrix}.$$

We define $\Delta g(\mathbf{W}, \mathbf{W}^*)\partial \log|\det \overline{\mathbf{W}}|$ and write the first-order Taylor series expansion given in (1.20) as

$$\Delta g(\mathbf{W}, \mathbf{W}^*) = \langle \Delta\mathbf{W}, \nabla_{\mathbf{W}^*} \log|\det \overline{\mathbf{W}}|\rangle + \langle \Delta\mathbf{W}^*, \nabla_{\mathbf{W}} \log|\det \overline{\mathbf{W}}|\rangle$$

which, upon comparison with (1.52) gives us the required result for the matrix gradient

$$\frac{\partial \log|\det\overline{\mathbf{W}}|}{\partial \mathbf{W}^*} = \mathbf{W}^{-H}. \tag{1.53}$$

We can then write the relative (natural) gradient updates to maximize the likelihood function using Eqs. (1.34), (1.49) and (1.53) as

$$\Delta\mathbf{W} = (\mathbf{W}^{-H} - \psi(\mathbf{u})\mathbf{x}^H)\mathbf{W}^H\mathbf{W} = (\mathbf{I} - \psi(\mathbf{u})\mathbf{u}^H)\mathbf{W}. \tag{1.54}$$

The update given above and the score function $\psi(\mathbf{u})$ defined in (1.50) coincide with the one derived in [20] using a $\mathbb{C} \mapsto \mathbb{R}^{2n}$ isomorphic mapping in a relative gradient update framework and the one given in [34] considering separate derivatives.

The update equation given in (1.54) can be also derived without explicit use of the relative gradient update rule given in (1.34). We can use (1.49), (1.53), and $\partial\mathbf{u} = (\partial\mathbf{W})\mathbf{x}$, to write the first-order differential of the likelihood term $\ell_t(\mathbf{W})$ as

$$\partial\ell_t = \text{Trace}(\partial\mathbf{W}\mathbf{W}^{-1}) + \text{Trace}(\partial\mathbf{W}^*\mathbf{W}^{-*}) - \psi^H(\mathbf{u})\partial\mathbf{u} - \psi^T(\mathbf{u})\partial\mathbf{u}^*. \tag{1.55}$$

Defining $\partial\mathbf{Z} \triangleq (\partial\mathbf{W})\mathbf{W}^{-1}$, we obtain $\partial\mathbf{u} = (\partial\mathbf{W})\mathbf{x} = \partial\mathbf{W}(\mathbf{W}^{-1})\mathbf{u} = (\partial\mathbf{Z})\mathbf{u}$, $\partial\mathbf{u}^* = (\partial\mathbf{Z}^*)\mathbf{u}^*$. By treating $\mathbf{W}$ as a constant matrix, the differential matrix $\partial\mathbf{Z}$ has components ∂z_{ij} that are linear combinations of ∂w_{ij} and is a non-integrable differential form. However, this transformation allows us to easily write (1.55) as

$$\partial\ell_t = \text{Trace}(\partial\mathbf{Z}) + \text{Trace}(\partial\mathbf{Z}^*) - \psi^H(\mathbf{u})(\partial\mathbf{Z})\mathbf{u} - \psi^T(\mathbf{u})(\partial\mathbf{Z}^*)\mathbf{u}^* \tag{1.56}$$

where we have treated $\mathbf{Z}$ and $\mathbf{Z}^*$ as two independent variables using Wirtinger calculus. Therefore, the gradient update rule for $\mathbf{Z}$ is given by

$$\Delta\mathbf{Z} = \frac{\partial\ell_t}{\partial\mathbf{Z}^*} = (\mathbf{I} - \mathbf{u}^*\psi^T(\mathbf{u}))^T = \mathbf{I} - \psi(\mathbf{u})\mathbf{u}^H \tag{1.57}$$

which is equivalent to (1.54) since $\partial\mathbf{Z} = (\partial\mathbf{W})\mathbf{W}^{-1}$.

The two derivations we have given here for the score function represent a very straightforward and simple evaluation compared to those in [20, 34], and more importantly, show how to bypass a major limitation in the development of ML theory for complex valued signal processing, that is working with probabilistic descriptions using complex algebra. In the second derivation, the introduction of the differential form $\partial\mathbf{Z}$, which is not a true differential as it is not integrable, provides a convenient form and is especially attractive in evaluation of higher-order differential expressions as demonstrated in [72].

Newton Updates for ML ICA The same definition, $\partial\mathbf{Z} \triangleq (\partial\mathbf{W})\mathbf{W}^{-1}$, can be used also to derive a Newton update rule in a compact form, as opposed to the element-wise form given in (1.35). To simplify the notation, we first define $l \triangleq -\ell_t$, and consider Newton updates to minimize the negative likelihood l, and then evaluate the second-order differential of the likelihood term l.

To write the differential of the term $\partial\ell = -\partial\,\ell_t$ given in (1.56) which is a function of $\{\mathbf{Z}, \mathbf{Z}^*, \mathbf{u}, \mathbf{u}^*\}$, we use Wirtinger calculus to write $\partial(\text{Trace}(\partial\mathbf{Z}))/\partial\mathbf{Z} = \mathbf{0}$ and $\partial(\text{Trace}(\partial\mathbf{Z}^*))/\partial\mathbf{Z}^* = \mathbf{0}$. Then, the second-order differential can be written as

$$\begin{aligned}\partial^2 l &= \partial[\psi^H(\mathbf{u})\partial\mathbf{Z}\mathbf{u} + \psi^T(\mathbf{u})\partial\mathbf{Z}^*\mathbf{u}^*] \\ &= 2\text{Re}\{\mathbf{u}^T\partial\mathbf{Z}^T\eta(\mathbf{u}, \mathbf{u}^*)\partial\mathbf{Z}\mathbf{u} + \mathbf{u}^T\partial\mathbf{Z}^T\theta(\mathbf{u}, \mathbf{u}^*)\partial\mathbf{Z}^*\mathbf{y}^* + \psi^H(\mathbf{u})\partial\mathbf{Z}\partial\mathbf{Z}\mathbf{u}\}\end{aligned}$$

where $\eta(\mathbf{u}, \mathbf{u}^*)$ is a diagonal matrix with ith diagonal element

$$-\frac{\partial \log p_i(u_i, u_i^*)}{\partial u_i \partial u_i}$$

and $\theta(\mathbf{u}, \mathbf{u}^*)$ is another diagonal matrix with ith diagonal element

$$-\frac{\partial \log p_i(u_i, u_i^*)}{\partial u_i \partial u_i^*}.$$

Using some simple algebra, we can write the expected value of the second differential term as

$$E\{\partial^2 l\} = \sum_{i\neq j} [\partial z_{ij}\ \partial z_{ji}\ \partial z_{ij}^*\ \partial z_{ji}^*] \begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_2 \\ \mathbf{H}_2^* & \mathbf{H}_1 \end{bmatrix} \begin{bmatrix} \partial z_{ij}^* \\ \partial z_{ji}^* \\ \partial z_{ij} \\ \partial z_{ji} \end{bmatrix} + \sum_i [\partial z_{ii}\ \partial z_{ii}^*]\, \mathbf{H}_3 \begin{bmatrix} \partial z_{ii}^* \\ \partial z_{ii} \end{bmatrix}$$

where $\mathbf{H}_1 = \begin{bmatrix} \beta_j\delta_i & 0 \\ 0 & \beta_i\delta_j \end{bmatrix}$, $\mathbf{H}_2 = \begin{bmatrix} \alpha_j\gamma_i & 1 \\ 1 & \alpha_i\gamma_j \end{bmatrix}$, $\mathbf{H}_3 = \begin{bmatrix} v_i & q_i + 1 \\ q_i^* + 1 & v_i \end{bmatrix}$, $\alpha_i = E\{u_i^2\}$, $\beta_i = E\{|u_i|^2\}$, $\gamma_i = E\{\eta_i(u_i, u_i^*)\}$, $\delta_i = E\{\theta_i(u_i, u_i^*)\}$, $q_i = E\{u_i^2\eta_i(u_i, u_i^*)\}$, and $v_i = E\{|u_i|^2\theta_i(u_i, u_i^*)\}$.

As given in (1.57), we have

$$\frac{\partial E\{l\}}{\partial \mathbf{Z}^*} = E\{\psi(\mathbf{u})\mathbf{u}^H\} - \mathbf{I}.$$

To derive the Newton update, we consider the diagonal and the off-diagonal elements of $E\{\partial^2 l\}$ separately. We define $\partial\tilde{\mathbf{z}}_{ii} \triangleq \begin{bmatrix} \partial z_{ii} \\ \partial z_{ii}^* \end{bmatrix}$, and can write

$$\frac{\partial E\{l\}}{\partial \tilde{\mathbf{z}}_{ii}} = \begin{bmatrix} (E\{\psi(\mathbf{u})\mathbf{u}^H\} - \mathbf{I})_{ii}^* \\ (E\{\psi(\mathbf{u})\mathbf{u}^H\} - \mathbf{I})_{ii} \end{bmatrix} \quad \text{and}$$

$$\frac{\partial^2 E\{l\}}{\partial \tilde{\mathbf{z}}_{ii}} = \mathbf{H}_3^* \partial\tilde{\mathbf{z}}_{ii}.$$

Therefore the Newton rule for updating $\partial\tilde{\mathbf{z}}_{ii}$ can be written by solving

$$\frac{\partial^2 E\{l\}}{\partial \tilde{\mathbf{z}}_{ii}} = -\frac{\partial E\{l\}}{\partial \tilde{\mathbf{z}}_{ii}}$$

as in (1.35) to obtain

$$\partial\tilde{\mathbf{z}}_{ii} = -\mathbf{H}_3^{-*} \begin{bmatrix} (E\{\psi(\mathbf{u})\mathbf{u}^H\} - \mathbf{I})_{ii}^* \\ (E\{\psi(\mathbf{u})\mathbf{u}^H\} - \mathbf{I})_{ii} \end{bmatrix} \tag{1.58}$$

and the update for ∂z_{ii}^* is simply the conjugate of ∂z_{ii}.

For each off-diagonal element pair ∂z_{ij}, we write $\partial\tilde{\mathbf{z}}_{ij} \triangleq \begin{bmatrix} \partial z_{ij} \\ \partial z_{ji} \\ \partial z_{ij}^* \\ \partial z_{ji}^* \end{bmatrix}$. As in the updates of the diagonal elements, we obtain

$$\frac{\partial E\{l\}}{\partial \tilde{\mathbf{z}}_{ij}} = \begin{bmatrix} (E\{\psi(\mathbf{u})\mathbf{u}^H\} - \mathbf{I})_{ij}^* \\ (E\{\psi(\mathbf{u})\mathbf{u}^H\} - \mathbf{I})_{ji}^* \\ (E\{\psi(\mathbf{u})\mathbf{u}^H\} - \mathbf{I})_{ij} \\ (E\{\psi(\mathbf{u})\mathbf{u}^H\} - \mathbf{I})_{ji} \end{bmatrix}$$

$$\frac{\partial^2 E\{l\}}{\partial \tilde{\mathbf{z}}_{ij}} = \begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_2^* \\ \mathbf{H}_2 & \mathbf{H}_1 \end{bmatrix} \partial\tilde{\mathbf{z}}_{ij}$$

and obtain the Newton update rule for the parameters $\partial\tilde{\mathbf{z}}_{ij}$ as in the previous case

$$\partial\tilde{\mathbf{z}}_{ij} = -\begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_2^* \\ \mathbf{H}_2 & \mathbf{H}_1 \end{bmatrix}^{-1} \begin{bmatrix} (E\{\psi(\mathbf{u})\mathbf{u}^H\} - \mathbf{I})_{ij}^* \\ (E\{\psi(\mathbf{u})\mathbf{u}^H\} - \mathbf{I})_{ji}^* \\ (E\{\psi(\mathbf{u})\mathbf{u}^H\} - \mathbf{I})_{ij} \\ (E\{\psi(\mathbf{u})\mathbf{u}^H\} - \mathbf{I})_{ji} \end{bmatrix} \tag{1.59}$$

where only the upper half elements of $\mathbf{Z}$ need to be updated as the lower half is given by the conjugates of the upper half elements.

Thus, the two sets of updates, (1.58) and (1.59) give the complete Newton update rule for $\partial\mathbf{Z}$. The final update rule for $\mathbf{W}$ is simply given by $\partial\mathbf{W} = \partial\mathbf{Z}\mathbf{W}$, which implies that the given Newton update can be called a *relative* Newton algorithm as its structure is similar to the relative gradient update given in (1.34). Also, note that if both Hessian terms in (1.58) and (1.59) are nonsingular, that is, positive definite, then the resulting Hessian in the updates will be equal to the identity matrix in the solution point $\mathbf{W} = \mathbf{A}^{-1}$ as discussed in [7] for the real-valued case.

1.6.2 Complex Maximization of Non-Gaussianity

Another natural cost function for performing ICA is the maximization of non-Gaussianity [28, 53]. Independence is achieved by moving the transformed mixture, that is, the independent source estimates $\mathbf{w}^H\mathbf{x}$ away from a Gaussian distribution. The natural cost in this case is negentropy that measures the entropic distance of a distribution from that of a Gaussian and can be written for the complex source as

$$\mathcal{J}(\mathbf{w}) = H(v_r, v_i) - H(u_r, u_i) \tag{1.60}$$

where $H(\cdot, \cdot)$ is the differential entropy of the given bivariate distribution and $v = v_r + jv_i$ denotes the Gaussian-distributed complex variable. Gaussian density yields the largest entropy when the covariances of the two variables v and u are fixed and attains its maximum for the circular case [81]. Hence, the first term in (1.60) is constant for a given covariance matrix, and the maximization of $\mathcal{J}(\mathbf{w})$ can be achieved by minimizing the differential entropy $H(u_r, u_i) = -E\{\log p_S(u_r, u_i)\}$ under a variance constraint. Hence, we can define the ICA cost function to minimize as

$$J_G(\mathbf{w}) = E\{|G(u)|^2\} = E\{|G(\mathbf{w}^H\mathbf{x})|^2\} \tag{1.61}$$

subject to a variance constraint, and choose the nonlinear function $G: \mathbb{C} \mapsto \mathbb{C}$ to match the source pdf, that is, as

$$p_s(u) = p_s(u_r, u_i) = K \exp(-|G(u)|^2)$$

where K is a constant, so that the minimization of (1.61) is equivalent to the maximization of (1.60). While writing the form of the pdf in terms of the nonlinearity $G(\cdot)$, we assumed that the expectations in (1.60) and (1.61) are written using ensemble averages over T samples using ergodic theorems. Unit variance is a typical and convenient constraint and has been a practical choice in this class of algorithms [54].

Note that for maximization of negentropy, we proceed by estimating a single source at a time, that is, an individual direction that is maximally non-Gaussian while in the case of ML estimation, the formulation leads to the estimation of all independent sources through the computation of a single demixing matrix $\mathbf{W}$. Hence, we need a

mechanism to avoid convergence to the same solution when estimating multiple sources, and, in addition, to impose a bound on the variance of the estimates. When we assume that the source signals have unit variance, that is, $E\{\mathbf{s}\mathbf{s}^H\} = \mathbf{I}$, then whitening the mixtures $\mathbf{v}$ prior to ICA such that $\mathbf{x} = \mathbf{M}\mathbf{v}$ and $E\{\mathbf{x}\mathbf{x}^H\} = \mathbf{I}$ implies that the demixing matrix $\mathbf{W}$ is unitary. Therefore in this case, we can perform ICA by first computing $\max_{\|\mathbf{w}_i\|^2=1} E\{|G(u_i)|^2\}$, and after the computation of each $\mathbf{w}_i$, by performing an orthogonalization procedure such as the Gram–Schmidt procedure [77] as in [53] such that $\mathbf{w}_i$ is orthogonal to $\{\mathbf{w}_j\}$, $1 \leq j < i$. The estimated sources are given by $u_i = \mathbf{w}_i^H \mathbf{x}$, $i = 1, \ldots, N$.

The cost function given in (1.61) provides a case where the $\mathbb{R}^2 \mapsto \mathbb{C}^2$ mapping used by Wirtinger calculus follows naturally. Note that the cost function can be written as

$$J_G(\mathbf{w}) = E\{G(u)(G(u))^*\} = E\{G(u)G(u^*)\} \tag{1.62}$$

where the last equality follows when $G(u)$ is analytic for $|u| < R$ with a Taylor series expansion with all real coefficients in $|u| < R$. Polynomial and most trigonometric functions and their hyperbolic counterparts satisfy this condition.

When written in the form $E\{G(u)G(u^*)\}$ as shown in (1.62), it is easy to see that the function $J_G(\mathbf{w})$ becomes complex-differentiable when considered separately with respect to the two arguments u and u^* (and consequently $\mathbf{w}$ and $\mathbf{w}^*$) if the function is chosen as an analytic function G: $\mathbb{C} \mapsto \mathbb{C}$ thus making it even easier to take advantage of Wirtinger calculus in the gradient evaluation.

For the cost function given in (1.62), the gradient is directly written as

$$\frac{\partial J_G(\mathbf{w})}{\partial \mathbf{w}^*} = E\{\mathbf{x}G(\mathbf{w}^T\mathbf{x}^*)G'(\mathbf{w}^H\mathbf{x})\} = E\{\mathbf{x}G^*(u)G'(u)\} \tag{1.63}$$

instead of evaluating the derivatives with respect to the real and imaginary parts as given in [83]. Here, we have $G'(\cdot) = dG(u)/du$. Similar to the real-valued algorithm for maximization of non-Gaussianity using gradient updates, for a general function $G(\cdot)$—a function not necessarily matched to the source pdf—we need to determine whether the cost function is being maximized or minimized by evaluating a factor γ during the updates such that $\gamma = E\{|G(u)|^2\} - E\{|G(v)|^2\}$. Since γ is a real-valued quantity and does not change the stationary points of the solution, we can simply include its sign estimate in the online updates and use $\Delta\mathbf{w} = \text{sign}(\gamma)\mu\mathbf{x}G^*(\mathbf{w}^H\mathbf{x})$ $G'(\mathbf{w}^H\mathbf{x})$ where $\mu > 0$ is the learning rate, and ensure the satisfaction of the unit norm constraints through a practical update scheme $\mathbf{w} \leftarrow \mathbf{w}/\|\mathbf{w}\|$ after each iteration of the weight vector. A more efficient update algorithm for performing ICA using the cost function in (1.61) is given in [84] using a constrained optimization formulation to ensure $\|\mathbf{w}\| = 1$ and using a modified Newton approach. The updates for this case are given by

$$\mathbf{w} \leftarrow E\{G'(u)(G')^*(u)\}\mathbf{w} - E\{G^*(u)G'(u)\mathbf{x}\} + E\{\mathbf{x}\mathbf{x}^T\}E\{G^*(u)G''(u)\}\mathbf{w}^* \tag{1.64}$$

where a following normalization step is used to ensure $\|\mathbf{w}\| = 1$ as in the gradient updates.

1.6.3 Mutual Information Minimization: Connections to ML and MN

As discussed in Sections 1.6.1 and 1.6.2, we can solve the complex ICA problem by maximizing the log likelihood function given by

$$\mathcal{L}(\mathbf{W}) = \sum_{t=1}^{T} \sum_{n=1}^{N} \log p_{S_n}(\mathbf{w}_n^H \mathbf{x}) + T \log|\det \overline{\mathbf{W}}|. \tag{1.65}$$

The weight matrix $\mathbf{W}$ to maximize the log likelihood can be computed using relative gradient update equation given in (1.54).

When using negentropy maximization as the objective, all sources can be estimated by maximizing the cost function

$$\begin{aligned} \mathcal{J}(\mathbf{W}) &= \sum_{n=1}^{N} E\{\log p_{S_n}(\mathbf{w}_n^H \mathbf{x})\} \\ &\approx \frac{1}{T} \sum_{t=1}^{T} \sum_{n=1}^{N} \log p_{S_n}(\mathbf{w}_n^H \mathbf{x}) \end{aligned} \tag{1.66}$$

under the unitary constraint for $\mathbf{W}$. The mean ergodic theorem is used to write (1.66) and when compared to the ML formulation given in (1.65), it is clear that the two objective functions are equivalent if we constrain the weight matrix $\mathbf{W}$ to be unitary for complex ML. Since $\det(\overline{\mathbf{W}}) = |\det(\mathbf{W})|^2$ [40], when $\mathbf{W}$ is unitary, the second term in (1.65) vanishes.

Similar to the real case given in [21], for the complex case, we can satisfy the unitary constraint for the weight matrix by projecting $\Delta\mathbf{W}$ to the space of skew-hermitian matrices. The resulting update equation is then given by

$$\Delta\mathbf{W} = (\mathbf{I} - \mathbf{u}\mathbf{u}^H - \psi(\mathbf{u})\mathbf{u}^H + \mathbf{u}\psi^H(\mathbf{u}))\mathbf{W}. \tag{1.67}$$

On the other hand, for the MN criterion, the weight matrix can be estimated in symmetric mode, or the individual rows of the weight matrix $\mathbf{W}$ can be estimated sequentially in a deflationary mode as in [52]. The latter procedure provides a more flexible formulation for individual source density matching than ML where each element of the score function $\psi(\mathbf{u})$ given in (1.51) needs to be matched individually.

As in the real case, the two criteria are intimately linked to mutual information. Written as the Kullback–Leibler distance between the joint and factored marginal

source densities, the mutual information is given by

$$\mathcal{I}(\mathbf{W}) = D\left(\|p(\mathbf{u})\| \prod_{n=1}^{N} p_{S_n}(u_n)\right) = \sum_{n=1}^{N} H(u_n) - H(\mathbf{u})$$

$$= \sum_{n=1}^{N} H(u_n) - H(\mathbf{x}) - \log|\det\overline{\mathbf{W}}| \tag{1.68}$$

where in the last line, we have again used the complex-to-real transformation for the source density given in (1.48). Since $H(\mathbf{x})$ is constant, using the mean ergodic theorem for the estimation of entropy, it is easy to see that minimization of mutual information is equivalent to ML, and when the weight matrix is constrained to be unitary, to the MN criterion.

1.6.4 Density Matching

For all three approaches for achieving ICA, the ML, MN, and mutual information minimization discussed in Sections 1.6.1–1.6.3, the nonlinearity used in the algorithm is expected to be matched as much as possible to the density for each estimated source. Also, the desirable large sample properties of the ML estimator assume their optimal values when the score function is matched to the source pdf, for example, the asymptotic covariance matrix of the ML estimator is minimum when the score function is chosen to match the source pdfs [89]. A similar result is given for the maximization of negentropy in [52]. A number of source density adaptation schemes have been proposed for performing ICA in the real-valued case, in particular for ML-based ICA (see *e.g.*, [24, 59, 66, 112, 120]) and more recently for the complex case [84, 85] for maximization of negentropy.

The most common approach for density adaptation has been the use of a flexible parametric model and to estimate the parameters—or a number of key parameters—of the model along with the estimation of the demixing matrix. In [89], a true ML ICA scheme has been differentiated as one that estimates both the source pdfs and the demixing matrix $\mathbf{W}$, and the common form of ML ICA where the nonlinearity is fixed and only the demixing matrix is estimated is referred to as quasi-maximum likelihood. Given the richer structure of possible distributions in the two-dimensional space compared to the real-valued, that is, single dimensional case, the pdf estimation problem becomes more challenging for complex-valued ICA. In the real-valued case, a robust nonlinearity such as the sigmoid nonlinearity provides satisfactory performance for most applications [29, 54] and the performance can be improved by matching the nonlinearity to the sub- or super-Gaussian nature of the sources [66]. In the complex case, the circular/noncircular nature of the sources is another important factor affecting the performance [3, 84]. Also, obviously the unimodal versus multimodal structure of the density requires special care in both the real and the complex case. Hence, in general, it is important to take *a priori* information into account when performing source matching.

If a given source has a circular distribution, that is, $p_{S_n}(u) = g(|u|)$, the corresponding entry of the score function vector can be easily evaluated as

$$\psi_n(u) = -\frac{\partial \log g(\sqrt{uu^*})}{\partial u^*} = -\frac{u}{2|u|}\left(\frac{g'(|u|)}{g(|u|)}\right).$$

Thus, the score function always has the same phase as its argument. This is the form of the score function proposed in [9] where all sources are assumed to be circular.

If the real and imaginary parts of a given source are mutually independent, the score function takes the form

$$\psi_n(u, u^*) = -\frac{1}{2}\left(\frac{\partial \log p_{S_r}(u_r)}{\partial u_r} + j\frac{\partial \log p_{S_i}(u_i)}{\partial u_i}\right)$$

and suggests the need to use separate real-valued functions for processing the real and imaginary arguments. For example, the score function proposed in [103] for complex Infomax, $\psi(u) = \tanh(u_r) + j\ \tanh(u_i)$, is shown to provide good performance for independent and circular sources [3].

For density matching, approaches such as the Gram–Charlier and Edgeworth expansions are proposed for the real case [19], and for the complex case, bivariate expansions such as those given in [76] can be adopted. However, such expansions usually perform well for unimodal distributions that are close to the Gaussian and their estimators are very sensitive to outliers thus usually requiring large number of samples. With the added dimensionality of the problem for the complex case, in comparison to the real (univariate) case, such expansions become even less desirable for complex density matching. Limitations of such expansions are discussed in detail in [104] where an efficient procedure for least-mean-square estimation of the score function is proposed for the real case.

Next, we discuss a number of possible density models and nonlinearity choices for performing complex ICA and discuss their properties. Simple substitution of $u_r = (u + u^*)/2$ and $u_i = (u - u^*)/2j$ allows us to write a given pdf that is $p(u_r, u_i)$: $\mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}$ in terms of a function $f(u, u^*)$: $\mathbb{C} \times \mathbb{C} \mapsto \mathbb{R}$. Since all smooth functions that define a pdf can be shown to satisfy the real differentiability condition, they can be used in the development of ICA algorithms and in their analyses using Wirtinger calculus.

Generalized Gaussian Density Model A generalized Gaussian density of order c of the form given in [26] can be written as a function $\mathbb{C} \times \mathbb{C} \mapsto \mathbb{R}$ as

$$f_{\mathrm{GG}}(u, u^*; \sigma_r, \sigma_i, \rho, c) = \beta \exp(-[\gamma\alpha(u, u^*)]^c) \tag{1.69}$$

where

$$\alpha(u, u^*) = \frac{(u+u^*)^2}{4\sigma_r^2} + j\frac{\rho(u^2 - u^{*2})}{2\sigma_r\sigma_i} - \frac{(u-u^*)^2}{4\sigma_i^2},$$

$$\beta = \frac{c\gamma}{\pi\Gamma(1/c)\sigma_r\sigma_i\sqrt{1-\rho^2}}, \quad \text{and} \quad \gamma = \frac{\Gamma(2/c)}{2(1-\rho^2)\Gamma(1/c)}.$$

In the above expression, σ_r and σ_i are the standard deviations of the real and imaginary parts, $\rho = \sigma_{r,i}/\sigma_r\,\sigma_i$ is the correlation coefficient between the two variables, and the distribution is assumed to be zero mean. When the shape parameter $c = 1$, the pdf takes the form of the standard bivariate Gaussian and is super-Gaussian for $0 < c < 1$ and sub-Gaussian for $c > 1$.

The score function for the pdf given in (1.69) can be evaluated by using (1.50) as

$$\psi(u, u^*) = c\gamma^c[\alpha(u, u^*)]^{c-1}\frac{\partial\alpha(u, u^*)}{\partial u^*}.$$

When the sources are circular, that is, $\sigma_r = \sigma_i = \sigma$ and $\rho = 0$, we have $\alpha(u, u^*) = uu^*/\sigma^2$, and for circular Gaussian sources ($c = 1$), the score function is linear $\psi(u, u^*) = u/2\sigma^2$ as expected, since circular Gaussian sources cannot be separated using ICA. However, noncircular Gaussians can be separated, since in this case, the score function is given by $\psi(u, u^*) = (u+u^*)/4(1-\rho^2)\sigma_r^2 - j\rho u^*/2(1-\rho^2)\sigma_r\sigma_i + (u-u^*)/4(1-\rho^2)\sigma_i^2$, and thus is nonlinear with respect to u. A simple procedure for estimating noncircular Gaussian sources using ML is given in [20]. However, the second-order approach, strongly uncorrelating transform [32, 65] provides a more efficient procedure for estimating noncircular Gaussian sources as long as the sources have unique spectral coefficients.

For the Gaussian case, we can also write the score function as in [20]

$$\psi(u, u^*) = \frac{uE\{|u|^2\} - u^*E\{u^2\}}{2(E\{|u|^2\}^2 - |E\{u^2\}|^2)}$$

to note the *widely linear* nature of the score function for Gaussian sources.

In [84], the univariate form of the generalized Gaussian density is used to model circular source densities for deriving ICA algorithms through negentropy maximization and significant performance gain is noted when the shape parameter c is updated during the estimation. Such a scheme can be adopted for ICA through ML as well and would also require the estimation of the variances of the real and imaginary parts of the sources when used for noncircular source distributions.

Mixture Model Generalized Gaussian mixture model provides a flexible alternative to source density matching, especially in cases where the sources are not

unimodal. The mixture model using the generalized Gaussian kernels given in (1.69) can be written as

$$f_{\mathrm{GM}}(u, u^*) = \sum_{k=1}^{K} \pi_k f_{\mathrm{GG}}(u, u^*; \sigma_r, \sigma_i, \rho, c)$$

where π_k denotes the mixing proportions of the generalized Gaussian kernels. An example application of the model would be quadrature amplitude modulated (QAM) sources where the model simplifies to

$$f_{\mathrm{QAM}}(u, u^*) = \frac{1}{K2\pi\sigma^2} \sum_{k=1}^{K} f_{\mathrm{G}}(u, u^*; \sigma, \mu_k) \tag{1.70}$$

where

$$f_{\mathrm{G}}(u, u^*; \sigma, \mu_k) = \exp\left[-\frac{1}{2\sigma^2}(u - \mu_k)(u - \mu_k)^*\right]$$

since the π_ks are taken as equal and the Gaussian kernels ($c = 2$) are circular ($\sigma_r = \sigma_i = \sigma$). The parameters, μ_k are determined by the QAM scheme, which is a *prior* information, for example, are given by $\{\pm 1\}$ for 4-QAM sources, and the value of σ can be determined by the level of noise in the system, which is assumed to be Gaussian. The score function can be easily evaluated as

$$\psi_{\mathrm{QAM}}(u, u^*) = \frac{\sum_{k=1}^{K}(u - \mu_k) f_{\mathrm{G}}(u, u^*; \sigma, \mu_k)}{2\sigma^2 \sum_{k=1}^{K} f_{\mathrm{G}}(u, u^*; \sigma, \mu_k)}.$$

Linear Combinations of Basis Functions In [20], the adaptive score functions of Pham and Garat [89] are extended to the complex case through $\mathbb{C}^N \mapsto \mathbb{R}^{2N}$ mappings. We can directly evaluate and write the adaptive scores in the complex domain as follows: Approximate the "true" score function $\psi_o(u, u^*)$ as a linear combination of M basis functions $\phi_m(u, u^*)$, $m = 1, \ldots, M$ such that $\psi(u, u^*) = \sum_{m=1}^{M} \gamma_m^* \phi_m(u, u^*) = \boldsymbol{\gamma}^H \boldsymbol{\phi}$ where $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_M]^T$ and $\boldsymbol{\phi} = [\phi_1(u, u^*), \ldots, \phi_M(u, u^*)]^T$. Then, the problem is to determine the coefficient vector $\boldsymbol{\gamma}$ for each source such that $E\{|\psi_o(u, u^*) - \boldsymbol{\gamma}^H \boldsymbol{\phi}|^2\}$ is minimized. The solution is given by $\boldsymbol{\gamma} = (E\{\boldsymbol{\phi}\boldsymbol{\phi}^H\})^{-1} E\{\boldsymbol{\phi}\psi_o^*(u, u^*)\}$. The term $E\{\boldsymbol{\phi}\psi_o^*(u, u^*)\}$ requires that we know the true score function, which typically is not available. The clever trick introduced in [89] allows one to bypass this limitation, and can be extended to the complex case using Wirtinger calculus as follows. We substitute the expression for $\psi_o(u, u^*)$ given in (1.51) to the integral evaluation for the expectation $E\{\boldsymbol{\phi}\psi_o^*(u, u^*)\}$ to obtain

$$E\{\boldsymbol{\phi}\psi_o^*(u, u^*)\} = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \boldsymbol{\alpha}(u_r, u_i) du_r du_i \tag{1.71}$$

where $\boldsymbol{\alpha}(u_r, u_i) \triangleq \boldsymbol{\phi}(\partial f_o(u, u^*)/\partial u)$, f_o denotes the true (and unknown) source pdf and, we have used $(\partial \log f_o(u, u^*)/\partial u^*)^* = \partial \log f_o(u, u^*)/\partial u$ since $f_o(u, u^*)$ is a pdf and hence real valued. Wirtinger calculus enables us to directly write

$$\phi_m(u, u^*)\frac{\partial f_o(u, u^*)}{\partial u} = \frac{\partial}{\partial u}(\phi_m(u, u^*) f_o(u, u^*)) - f_o(u, u^*)\frac{\partial \phi_m(u, u^*)}{\partial u} \tag{1.72}$$

by using the chain rule. When (1.72) is substituted into (1.71), we obtain the important equality that shows how to evaluate the coefficients for adaptive scores using expectations without knowledge of the true source distributions

$$E\{\boldsymbol{\phi}\psi_o^*(u, u^*)\} = E\left\{\frac{\partial \boldsymbol{\phi}}{\partial u}\right\} \tag{1.73}$$

which holds when the product $f_o(u, u^*)\phi_m^*(u, u^*)$ vanishes at infinity for u_r and u_i. In the evaluation for this term, we used the integral formula given in (1.9) to write the symbolic integral given in terms of u and u^* as a contour integral of a single complex variable.

In the real case, it is shown that if the set of basis functions contains at least the identity function plus some other nonlinear function, then the stability of the separation is guaranteed [89]. For the real-valued generalized Gaussian density, a combination of three basis functions $\phi_{(1)}$, $\phi_{(0.75)}$, and $\phi_{(2)}$ that correspond to the score functions with shape parameters $c = 1$, 0.75, and 2, that is, an identity (linear Gaussian score), and one corresponding to a typical super- and one to a sub-Gaussian density have been used. In the complex case, to account for the additional dimensionality, we propose to use $\phi_1 = u$, $\phi_2 = u\,\alpha_{(0.75)}(u, u^*)$, $\phi_3 = u^*\alpha_{(0.75)}(u, u^*)$, $\phi_4 = u\,\alpha_{(2)}(u, u^*)$, $\phi_5 = u^*\alpha_{(2)}(u, u^*)$ where $\alpha(u, u^*)$ is defined in (1.69). An expansion that includes these basis functions accounts for all the terms present in the evaluation of the score function $\psi(u, u^*)$ for the generalized Gaussian density given in (1.69) along with a choice similar to those to in [89] for the shape parameters. It is worth noting that it is also possible to estimate coefficients of any nonlinear approximation to the score function such as those using splines or MLPs using a criterion such as least squares. However, the approach proposed here as in [89] has the advantage of leading to a unique solution that can be easily computed.

1.6.5 Numerical Examples

Since our focus in this section has primarily been on establishing a complete framework for complex ICA, and not on algorithm implementation and density matching mechanisms, in this section, we select examples to demonstrate the relationship between the two main classes of ICA approaches, complex ML (CML) and complex MN (CMN), to each other and to other main complex ICA approaches.

We test the performance of complex maximum likelihood using the relative gradient update in (1.54), which we refer to as the CML algorithm, and the version that constrains the demixing matrix to be unitary using the update in (1.67), the CML-unitary

algorithm. For maximization of non-Gaussianity, we use the modified Newton update shown in (1.64) as its performance matches that of the gradient update (1.63) when the stepsize is chosen correctly for the gradient approach [83]. We demonstrate the performance of the algorithms for three sets of sources, a set of 4-QAM and binary phase-shift keying (BPSK) sources using (1.70), and a circular set from a generalized Gaussian distribution (GGD) with different values for the shape parameter c as in (1.69). Hence, we have sources with all three types of circularity: GGDs that are strictly circular, QAM sources that are second-order circular, and noncircular BPSK sources. For the CML and CMN updates, the nonlinearity is matched to the form of the source distribution for each run, and for the 4-QAM and BPSK simulations, parameter σ in (1.70) is chosen as 0.25, which corresponds to 12 dB signal-to-noise ratio. The 4-QAM and BPSK sources are sub-Gaussian with a normalized kurtosis value of $-.885$ and $-.77$ respectively for the given σ. The GGD sources are super-Gaussian approaching to Gaussian when the shape parameter c approaches 1.

We include the performances of complex nonlinear decorrelations (C-ND) [3] using the $-\text{asinh}(u) + u$ nonlinearity for the sub-Gaussian sources, and the performances of complex FastICA [13] using the log nonlinearity, the kurtosis maximization (KM) algorithm [69], JADE with the version that uses simultaneous diagonalization of N cumulant matrices [22], and for the circular generalized Gaussian sources, complex Infomax using the nonlinear function that assumes circularity given in [9]. Since we have not considered density adaptation, all sources in a given run are generated from the same distribution, and as a result comparisons with SUT are not included since for SUT, all sources have to have distinct spectral coefficients. For CMN, we implemented symmetric orthogonalization such that all sources are estimated in parallel and the demixing matrix is orthogonalized using $\mathbf{W} \leftarrow (\mathbf{W}\mathbf{W}^H)^{1/2}\mathbf{W}$, which is noted to provide slightly better performance when all the source densities are the same [85].

As the performance index, we use the inter-symbol-interference (ISI)—or the positive separation index [73]—given by

$$\text{ISI} = \frac{1}{2N(N-1)}\left[\sum_{i=1}^{N}\left(\sum_{j=1}^{N}\frac{|p_{ij}|}{\max_k |p_{ik}|} - 1\right) + \sum_{j=1}^{N}\left(\sum_{i=1}^{N}\frac{|p_{ij}|}{\max_k |p_{kj}|} - 1\right)\right]$$

where p_{ik} are the elements of the matrix $\mathbf{P} = \mathbf{W}\mathbf{A}$, N is the number of sources, and the lower the ISI value the better the separation performance.

Figures 1.19 and 1.20 show the ISI values for six 4-QAM and six BPSK sources with increasing number of samples and Figure 1.21, the ISI values for six circular GGD sources as the shape parameter c varies from 0.125 to 0.75, from highly super-Gaussian to closely Gaussian for 5000 samples. For both cases, the results are the average of 10 independent runs with the least ISI out of 25, as we wanted to compare the approximate best performance of all algorithms. With this selection, the standard deviation for all the algorithms were in the range 10^{-8} for small number of samples and 10^{-11} when the number of samples are increased.

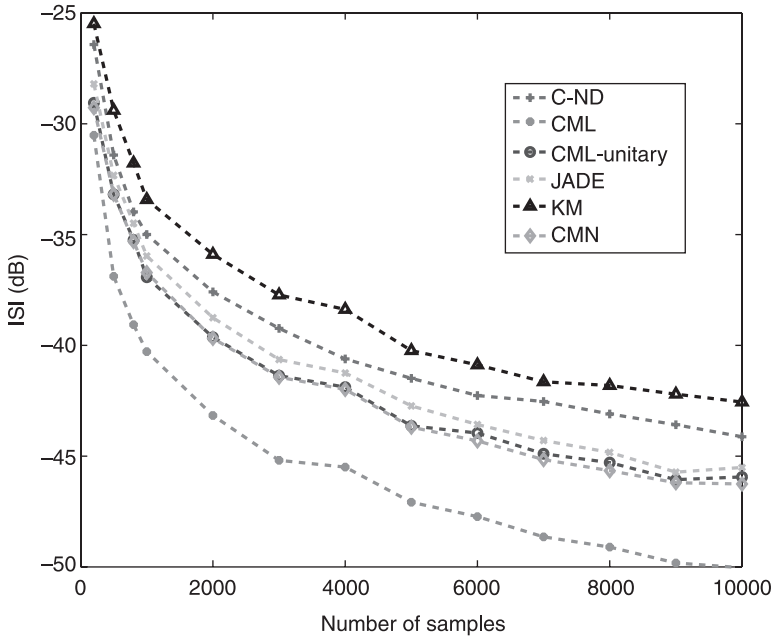

Figure 1.19 ISI as a function of number of samples for six 4-QAM sources.

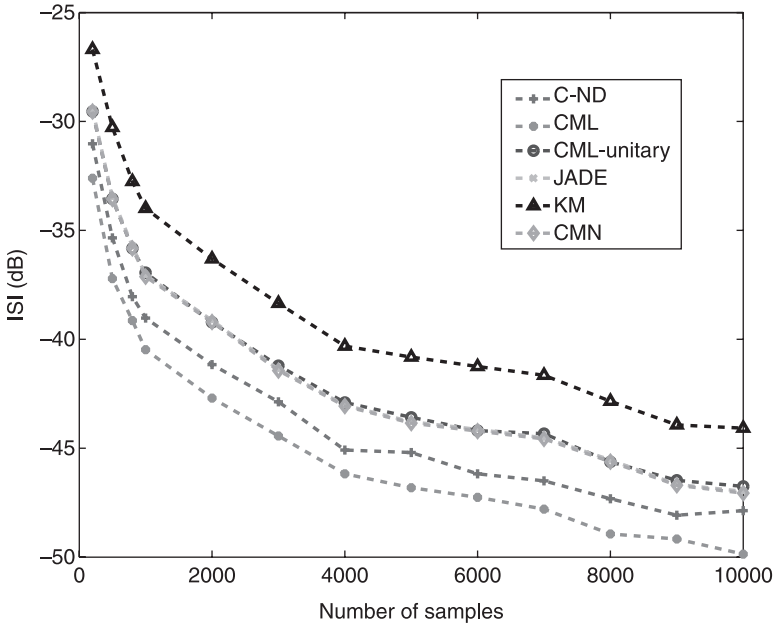

Figure 1.20 ISI as a function of number of samples for six BPSK sources.

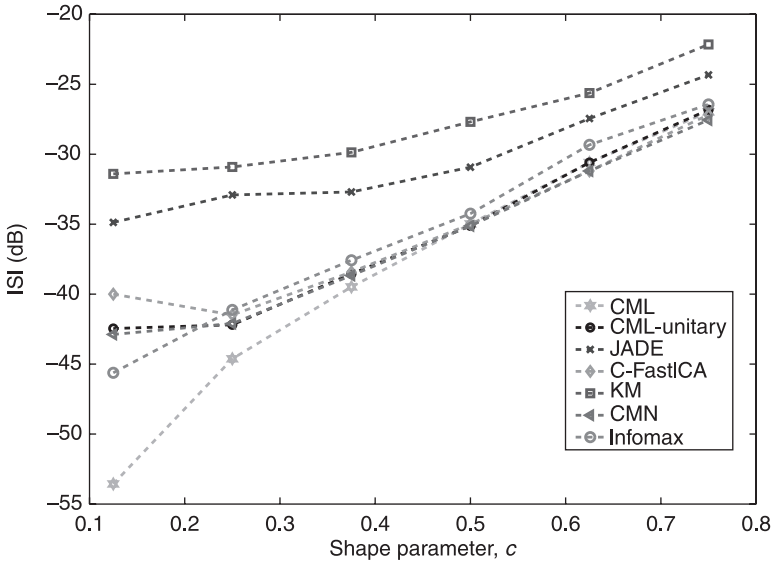

Figure 1.21 ISI as a function of the shape parameter c six GGD sources.

For all cases, we note the best performance by the CML algorithm, and the almost identical performance of the CML-unitary and CMN updates that have equivalent cost functions as discussed in Section 1.6.3. Other complex ICA approaches provide considerably satisfactory performance at a lower cost, in particular, JADE for the sub-Gaussian 4-QAM and BPSK sources, complex nonlinear decorrelations with $-\text{asinh}(u) + u$ nonlinearity for BPSK sources, and C-FastICA and complex Infomax that assume circularity for the circular GGD sources. The performance advantage of density matching comes at a computational cost as expected. The ML class of algorithms are computationally most costly when employed with density matching followed by the CMN algorithm. For example, the computational cost measured in terms of time for a single run of CML (and similarly for CML-unitary), without any optimization for implementation speed, is approximately 15 times that of KM, C-FastICA, and JADE, and three times that of CMN for six 4-QAM sources for 5000 samples. For the GGD sources, it is approximately 12 times that of KM, C-FastICA, JADE, CMN, and six times that of the ML/Infomax or nonlinear decorrelation approaches with a fixed nonlinearity.

1.7 SUMMARY

In this chapter, we provide the necessary tools for the development and analysis of algorithms in the complex domain and introduce their application to two important signal processing problems—filtering and independent component analysis. Complex-valued signal processing, we note, is not a simple extension of the real-valued case.

The definition of analyticity and commonly invoked assumptions such as circularity have been primarily motivated by this desire, such that the computations in the complex domain parallel those in the real domain. Wirtinger calculus, on the other hand, generalizes the definition of analyticity and enables development of a very convenient framework for complex-valued signal processing, which again, as desired, allows computations to be performed similar to the real case. Another important fact to note is that the framework based on Wirtinger calculus is a complete one, in the sense that the analytic case is included as a special case. Another attractive feature of the framework is that promising nonlinear structures such as fully complex (analytic) functions can be easily incorporated in algorithm development both for use within nonlinear filter structures such as MLPs and for the development of effective algorithms for performing independent component analysis. Commonly invoked assumptions such as circularity can be also easily avoided in the process, making the resulting algorithms applicable to a general class of signals, thus not limiting their usefulness. The only other reference besides this chapter—to the best of our knowledge—that fully develops the optimization framework including second-order relationships is [64], where the term $\mathbb{CR}$ calculus is used instead of Wirtinger calculus.

Though very limited in number, various textbooks have acknowledged the importance of complex-valued signals. In the most widely used book on adaptive filtering, [43], the complete development is given for complex signals starting with the 1991 edition of the book. In [43, 60, 97, 105], special sections are dedicated to complex signals, and optimization in the complex domain is introduced using the forms of the derivatives given in (1.5), also defined by Brandwood [15]. The simple trick that allows regarding the complex function as a function of two variables, $\mathbf{z}$ and $\mathbf{z}^*$, which significantly simplifies all computations, however, has not been noted in general. Even in the specific instance where it has been noted—a recent book [75] following [4, 6, 70, 71]—Wirtinger calculus is relegated to an afterthought as derivations are still given using the unnecessarily long and tedious split approach as in the previous work by the authors, for example, as in [38, 39, 41]. The important point to note is that besides simplifying derivations, Wirtinger calculus eliminates the need for many restrictive assumptions and extends the power of many convenient tools in analysis introduced for the real-valued case to the complex one. A simple example is the work in [72] where the second-order analysis of maximum likelihood independent component analysis is performed using a transformation introduced in [7] for the real-valued case while bypassing the need for any circularity assumption.

It is also interesting to note that the two forms for the derivatives given in [15], which are the correct forms and include the analytic case as well, have not been widely adopted. In a recent literature search, we noted that a significant portion of the papers published in the IEEE Transactions on Signal Processing and IEEE Transactions on Neural Networks within the past five years define the complex derivative differently than the one given in [15], which was published in 1983. The situation regarding contradictory statements and conflicting definitions in the complex domain unfortunately becomes more discouraging when we look at second-order expansions and algorithms. Even though the algorithms developed with derivative definitions other than those in (1.5) still provide reasonable—and in certain cases—equivalent

processing capability, these are ad-hoc solutions and typically do not fully take advantage of the complete information and processing capability offered by the complex domain. The development we present in this chapter, on the other hand, is a complete one. When conditions such as analyticity are satisfied, or when certain assumptions such as circularity are invoked, all the results we have derived simply reduce to the versions reported earlier in the literature.

Our hope in putting together this chapter has been to describe an effective framework for the complex domain, to present all the tools under a complete and consistent umbrella, and also to attract attention to two filtering solutions for complex-valued signal processing. Widely linear and fully complex nonlinearities promise to provide effective solutions for the challenging signal processing problems of next generation systems. Both of them also open new avenues for further research and deserve much attention.

1.8 ACKNOWLEDGMENT

The work of the authors on complex-valued adaptive filtering has been supported by the National Science Foundation (Awards NSF-CCF 0635129 and NSF-IIS 0612076).

1.9 PROBLEMS

1.1 Green's theorem can be stated as [1]:

For a function $f(z) = f(x, y) = u(x, y) + jv(x, y)$, let the real-valued functions $u(x, y)$ and $v(x, y)$ along with their partial derivatives u_x, u_y, v_x, and v_y, be continuous throughout a simply connected region $\mathcal{R}$ consisting of points interior to and on a simple closed contour (described in the counter-clockwise direction) $\mathcal{C}_{\mathcal{R}}$ in the x-y plane. We then have

$$\oint_{\mathcal{C}_{\mathcal{R}}} (udx + vdy) = \int\int_{\mathcal{R}} \left(\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right) dx\, dy.$$

Derive the integral formula given in (1.9) using Green's formula and the Wirtinger derivatives given in (1.5).

1.2 Verify the properties of complex-to-real mapping $\overline{(\cdot)}$: $\mathbb{C}^N \to \mathbb{R}^{2N}$ given in Section 1.2.3.

1.3 A simple way to generate samples from a circular distribution is to first generate real-valued nonnegative samples r from a selected pdf $p(r)$, and then to form the circular complex samples as

$$x + jy = rc^{j2\pi\theta} \tag{1.74}$$

where θ are samples from a uniform distribution in the range [0, 1].

We would like to generate samples from a circular generalized Gaussian distribution (GGD)—also called the exponential power distribution. We can use the procedure given in [57] to generate GGD samples with shape parameter c and scaling σ using the expression $[\texttt{gamrnd}(1/2\text{c}, \sigma)]^{1/2c}$ where the MATLAB (www.mathworks.com) function `gamrnd` generates samples from a gamma distribution with shape parameter $1/2c$ and scale parameter σ.

Explain why using this procedure directly to generate samples for the magnitude, r, will not produce samples with the same shape parameter as the bivariate case. How can you modify the expression $[\texttt{gamrnd}(1/2c, \sigma)]^{1/2c}$ so that the resulting samples will be circular-distributed GGDs with the shape parameter c when the expression given in (1.74) is used.

Hint: A simple way to check for the form of the resulting probability density function is to consider the case $c = 1$, that is, to consider the Gaussian special case.

1.4 Using the two mappings given in Proposition 1, Eqs. (1.25) and (1.26), and real-valued conjugate gradient algorithm given in Section 1.3.1, derive the complex conjugate gradient algorithm which is stated in Section 1.3.4.

1.5 Write the widely linear estimate given in (1.37) using the $\mathbb{C}^N$ notation by defining

$$\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}$$

and show that the optimum widely linear vector estimates can be written as

$$\mathbf{v}_{1,\text{opt}} = [\mathbf{C} - \mathbf{P}\mathbf{C}^{-*}\mathbf{P}^*]^{-1}[\mathbf{p} - \mathbf{P}\mathbf{C}^{-*}\mathbf{q}^*]$$

and

$$\mathbf{v}_{2,\text{opt}} = [\mathbf{C}^* - \mathbf{P}^*\mathbf{C}^{-1}\mathbf{P}]^{-1}[\mathbf{q}^* - \mathbf{P}^*\mathbf{C}^{-1}\mathbf{p}]$$

in $\mathbb{C}^N$, where $(\cdot)^{-*}$ is the complex conjugate of the inverse.

Use the forms given above for $\mathbf{v}_{1,\text{ opt}}$ and $\mathbf{v}_{2,\text{ opt}}$ to show that the mean-square error between a widely linear and linear filter J_{diff} is given by the expression in (1.38).

1.6 Given a finite impulse response system with the impulse response vector $\mathbf{w}_{\text{opt}}$ with coefficients $w_{\text{opt},n}$ for $n = 1, \ldots, N$.

Show that, if the desired response is written as

$$d(n) = \mathbf{w}_{\text{opt}}^H \mathbf{x}(n) + v(n)$$

where $\mathbf{x}(n) = [x(n)x(n-1) \cdots x(N-1)]^T$ and both the input $x(n)$ and the noise term $v(n)$ are zero mean and $x(n)$ is uncorrelated with both $v(n)$ and $v^*(n)$, then

the mean-square weight estimator is given by

$$\mathbf{w} = \mathbf{C}^{-1}\mathbf{p}$$

or by

$$\mathbf{w}^* = \mathbf{P}^{-1}\mathbf{q}$$

where the covariance and pseudo covariance matrices $\mathbf{C}$ and $\mathbf{P}$ as well as the cross covariance vectors $\mathbf{p}$ and $\mathbf{q}$ are defined in Section 1.4. Consequently, show that the mean-square error difference between a linear and a widely linear MSE filter ($J_{\text{diff}} = J_{L,\min} - J_{WL,\min}$) for this case is exactly zero, that is, using a widely linear filter does not provide any additional advantage even when the signal is noncircular.

1.7 The conclusion in Problem 1.6 can be extended to prediction of an autoregressive process given by

$$X(n) + \sum_{k=1}^{N} a_k X(n-k-1) = V(n)$$

where $V(n)$ is the white Gaussian noise. For simplicity, assume one-step ahead predictor and show that $J_{\text{diff}} = 0$ as long as $V(n)$ is a doubly white random process, that is, the covariance and the pseudo covariance functions of $V(n)$ satisfy $c(k) = c(0)\delta(k)$ and $p(k) = p(0)\delta(k)$ respectively.

1.8 For the widely linear weight vector error difference $\boldsymbol{\varepsilon}(n) = \mathbf{v}(n) - \mathbf{v}_{\text{opt}}$, show that we can write the expression for the modes of the widely linear LMS algorithm given in (1.41) as

$$E\{\varepsilon_k'(n)\} = \varepsilon_k'(0)(1 - \mu\bar{\lambda}_k)^n$$

and

$$E\{|\varepsilon_k'(n)|^2\} = \frac{\mu J_{WL,\min}}{2 - \mu\bar{\lambda}_k} + (1 - \mu\bar{\lambda}_k)^{2n}\left(|\varepsilon_k'(0)|^2 - \frac{\mu J_{WL,\min}}{2 - \mu\bar{\lambda}_k}\right)$$

as shown in [16, 43] for the linear LMS algorithm. Make sure you clearly identify all assumptions that lead to the expressions given above.

1.9 Explain the importance of the correlation matrix eigenvalues on the performance of the linear and widely linear LMS filter (λ and $\bar{\lambda}$). Let input $x(n)$ be a first order autoregressive process ($N = 1$ for the AR process given in Problem 1.7) but let the white Gaussian noise $v(n)$ be noncircular such that the pseudo-covariance $E\{v^2(n)\} \neq 0$. Show that when the pseudo-covariance matrix is nonzero, the

eigenvalue spread of the augmented covariance matrix $\bar{\mathbf{C}}$ will always be greater than or equal to that of the original covariance matrix $\mathbf{C}$ using the majorization theorem [49].

1.10 In real-valued independent component analysis, separation is possible as long as only one of the sources is Gaussian. In the complex case, however, as discussed in Section 1.6.4, Gaussian sources can be separated as long as they are noncircular with unique spectral coefficients.

Show that the score function for Gaussian sources can be reduced to

$$\psi_n(u) = \frac{u_r}{4\sigma_r^2} + j\frac{u_i}{4\sigma_i^2}$$

when we consider the scaling ambiguity for the complex case. Then, devise a procedure for density (score function) matching for the estimation of complex Gaussian sources.

REFERENCES

1. M. J. Ablowitz and A. S. Fokas, *Complex Variables*. Cambridge University Press, Cambridge, UK, (2003).
2. T. Adalı and V. D. Calhoun, Complex ICA of medical imaging data. *IEEE Signal Proc. Mag.*, 24(5):136–139, (2007).
3. T. Adalı, T. Kim, and V. D. Calhoun, Independent component analysis by complex nonlinearities. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, V, pp. 525–528, Montreal, QC, Canada, May (2004).
4. T. Adalı and H. Li, A practical formulation for computation of complex gradients and its application to maximum likelihood. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, II, pp. 633–636, Honolulu, HI, April (2007).
5. T. Adalı and H. Li, On properties of the widely linear MSE filter and its LMS implementation. In *Proc. Conf. Information Sciences and Systems*, Baltimore, MD, March (2009).
6. T. Adalı, H. Li, M. Novey, and J.-F. Cardoso, Complex ICA using nonlinear functions. *IEEE Trans. Signal Processing*, 56(9):4356–4544, (2008).
7. S.-I. Amari, T.-P. Chen, and A. Cichocki, Stability analysis of learning algorithms for blind source separation. *Neural Networks*, 10(8):1345–1351, (1997).
8. P. O. Amblard, M. Gaeta, and J. L. Lacoume, Statistics for complex variables and signals—Part 2: Signals. *Signal Processing*, 53(1):15–25, (1996).
9. J. Anemüller, T. J. Sejnowski, and S. Makeig, Complex independent component analysis of frequency-domain electroencephalographic data. *Neural Networks*, 16:1311–1323, (2003).
10. P. Arena, L. Fortuna, R. Re, and M. G. Xibilia, Multilayer perceptrons to approximate complex valued functions. *International Journal of Neural Systems*, 6:435–446, (1995).
11. A. Bell and T. Sejnowski, An information maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, (1995).

12. N. Benvenuto and F. Piazza, On the complex backpropagation algorithm. *IEEE Trans. Signal Processing*, 40(4):967–969, (1992).
13. E. Bingham and A. Hyvärinen, A fast fixed-point algorithm for independent component analysis of complex valued signals. *Int. J. Neural Systems*, 10:1–8, (2000).
14. D. L. Birx and S. J. Pipenberg, A complex mapping network for phase sensitive classification. *IEEE Trans. Neural Networks*, 4(1):127–135, (1993).
15. D. H. Brandwood, A complex gradient operator and its application in adaptive array theory. *Proc. Inst. Elect. Eng.*, 130(1):11–16, (1983).
16. H. J. Butterweck, A steady-state analysis of the LMS adaptive algorithm without the use of independence assumption. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, pp. 1404–1407, Detroit, (1995).
17. V. D. Calhoun, *Independent Component Analysis for Functional Magnetic Resonance Imaging*. Ph.D thesis, University of Maryland Baltimore County, Baltimore, MD, 2002.
18. V. D. Calhoun and T. Adalı, Complex ICA for fMRI analysis: Performance of several approaches. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, II, pp. 717–720, Hong Kong, China, April (2003).
19. J.-F. Cardoso, Blind signal separation: Statistical principles. *Proc. IEEE*, 86(10):2009–2025, (1998).
20. J.-F. Cardoso and T. Adalı, The maximum likelihood approach to complex ICA. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, V, pp. 673–676, Toulouse, France, May (2006).
21. J.-F. Cardoso and B. Laheld, Equivariant adaptive source separation. *IEEE Trans. Signal Processing*, 44:3017–3030, (1996).
22. J.-F. Cardoso and A. Souloumiac, Blind beamforming for non-Gaussian signals. *IEE Proc. Radar Signal Processing*, 140:362–370, (1993).
23. P. Chevalier and F. Pipon, New insights into optimal widely linear array receivers for the demodulation of BPSK, MSK, and GMSK signals corrupted by noncircular interferences—application to SAIC. *IEEE Trans. Signal Processing*, 54(3):870–883, (2006).
24. S. Choi, A. Cichocki, and S.-I. Amari, Flexible independent component analysis. *J. VLSI Signal Processing Systems for Signal, Image, and Video Technology*, 26(1/2):25–38, (2000).
25. A. Cichocki and R. Unbehauen, Robust neural networks with online learning for blind identification and blind separation of sources. *IEEE Trans. Circuits Syst. I: Fund. Theory Apps.*, 43:894–906, (1996).
26. M. Z. Coban and R. M. Mersereau, Adaptive subband video coding using bivariate generalized gaussian distribution model. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, IV, pp. 1990–1993, Atlanta, GA, May (1996).
27. P. Comon, Circularité et signaux aléatoires à temps discret. *Traitement du Signal*, 11(5):417–420, (1994).
28. P. Comon, Independent component analysis—a new concept? *Signal Processing*, 36:287–314, (1994).
29. N. Correa, T. Adalı, and V. D. Calhoun, Performance of blind source separation algorithms for fMRI analysis using a group ICA method. *Magnetic Resonance Imaging*, 25(5):684–694, (2007).

30. G. Cybenko, Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2:303–314, (1989).

31. J. P. D'Angelo, *Inequalities from Complex Analysis*, volume 28 of *Carus Mathematical Monographs*. Mathematical Association of America, (2002).

32. J. Eriksson and V. Koivunen, Complex-valued ICA using second order statistics. In *Proc. IEEE Int. Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 183–192, Saõ Luis, Brazil, Sept. (2004).

33. J. Eriksson and V. Koivunen, Complex random vectors and ICA models: Identifiability, uniqueness and separability. *IEEE Trans. Info. Theory*, 52(3):1017–1029, (2006).

34. J. Eriksson, A. Seppola, and V. Koivunen, Complex ICA for circular and non-circular sources. In *Proc. European Signal Process. Conf. (EUSIPCO)*, Antalya, Turkey, (2005).

35. K. Funahashi, On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2:183–192, (1989).

36. G. Georgiou and C. Koutsougeras, Complex back-propagation. *IEEE Trans. Circuits Syst. II*, 39(5):330–334, (1992).

37. S. L. Goh, M. Chen, D. H. Popovic, K. Aihara, D. Obradovic, and D. P. Mandic, Complex-valued forecasting of wind profile. *Renewable Energy*, 31:1733–1750, (2006).

38. S. L. Goh and D. P. Mandic, Nonlinear adaptive prediction of complex-valued signals by complex-valued PRNN. *IEEE Trans. Signal Processing*, 53(5):1827–1836, (2005).

39. S. L. Goh and D. P. Mandic, A general fully adaptive normalised gradient descent learning algorithm for complex-valued nonlinear adaptive filters. *IEEE Trans. Neural Networks*, 18(5):1511–1516, (2007).

40. N. R. Goodman, Statistical analysis based on a certain multivariate complex Gaussian distribution. *Annals Math. Stats.*, 34:152–176, (1963).

41. A. I. Hanna and D. P. Mandic, A fully adaptive normalized nonlinear gradient descent algorithm for complex-valued nonlinear adaptive filters. *IEEE Trans. Signal Processing*, 51(10):2540–2549, (2003).

42. S. Haykin, *Neural Networks: A Comprehensive Foundation (2nd Edition) Neural Networks: A Comprehensive Foundation*. Prentice-Hall, Inc., Second edition, (1999).

43. S. Haykin, *Adaptive Filter Theory*. Prentice-Hall, Inc., Upper Saddle River, NJ, fourth edition, (2002).

44. P. Henrici, *Applied and Computational Complex Analysis*, III, Wiley, New York, NY, (1986).

45. A. Hirose, Continuous complex-valued back-propagation learning. *Electronics Letts.*, 28(20):1854–1855, (1992).

46. A. Hjōrungnes and D. Gesbert, Complex-valued matrix differentiation: Techniques and key results. *IEEE Trans. Signal Processing*, 55(6):2740–2746, (2007).

47. F. G. C. Hoogenraad, P. J. W. Pouwels, M. B. M. Hofman, J. R. Reichenbach, M. Sprenger, and E. M. Haacke, Quantitative differentiation between bold models in FMRI. *Magnetic Resonance in Medicine*, 45:233–246, (2001).

48. L. Hörmander, *An Introduction to Complex Analysis in Several Variables*. North-Holland, Oxford, UK, (1990).

49. R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, New York, NY, (1999).

50. R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge University Press, New York, NY, (1999).
51. K. Hornik, M. Stinchcombe, and H. White, Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, (1989).
52. A. Hyvärinen, One-unit contrast functions for independent component analysis: A statistical analysis. In *Proc. IEEE Workshop on Neural Networks for Signal Processing (NNSP)*, pp. 388–397, Amelia Island, FL, Sept. (1997).
53. A. Hyvärinen, Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans. Neural Networks*, 10(3):626–634, (1999).
54. A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. Wiley, New York, NY, (2001).
55. J.-J. Jeon, RLS adaptation of widely linear minimum output energy algorithm for DS-CDMA systems. In *Proc. Adv. Industrial Conf. on Telecommunications*, pp. 98–102, (2005).
56. J.-J. Jeon, J. G. Andrews, and K.-M. Sung, The blind widely linear minimum output energy algorithm for DS-CDMA systems. *IEEE Trans. Signal Processing*, 54(5):1926–1931, (2006).
57. M. E. Johnson, Computer generation of the exponential power distributions. *Journal of Statistical Computation and Simulation*, 9:239–240, (1979).
58. C. Jutten and J. Hérault, Blind separation of sources, Part 1: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24(1):1–10, (1991).
59. J. Karvanen, J. Eriksson, and V. Koivunen, Pearson system based method for blind separation. In *Proc. Second Int. Workshop on ICA*, Helsinki, Finland, (2000).
60. S. M. Kay, *Fundamentals of Statistical Processing, Volume I: Estimation Theory*, Prentice Hall Signal Processing Series, Upper Saddl River, NJ (1993).
61. T. Kim and T. Adalı, Fully complex backpropagation for constant envelope signal processing. In *Proc. IEEE Workshop on Neural Networks for Signal Processing (NNSP)*, pp. 231–239. IEEE, Dec. (2000).
62. T. Kim and T. Adalı, Fully complex multi-layer perceptron network for nonlinear signal processing. *J. VLSI Signal Processing Systems for Signal, Image, and Video Technology*, 32:29–43, (2002).
63. T. Kim and T. Adalı, Approximation by fully complex multilayer perceptrons. *Neural Computation*, 15:1641–1666, (2003).
64. K. Kreutz-Delgado, ECE275A: Parameter Estimation I, Lecture Supplement on Complex Vector Calculus, New York, (2007).
65. L. De Lathauwer and B. De Moor, On the blind separation of non-circular sources. In *Proc. European Signal Process. Conf. (EUSIPCO)*, Toulouse, France, (2002).
66. T.-W. Lee, M. Girolami, and T. J. Sejnowski, Independent component analysis using an extended Infomax algorithm for mixed subgaussian and supergaussian sources. *Neural Computation*, 11:417–441, (1999).
67. H. Leung and S. Haykin, The complex backpropagation algorithm. *IEEE Trans. Signal Processing*, 39:2101–2104, (1991).
68. H. Li, *Complex-Valued Adaptive Signal Processing using Wirtinger Calculus and its Application to Independent Component Analysis*. Ph.D thesis, University of Maryland Baltimore County, Baltimore, MD, (2008).

69. H. Li and T. Adalı, Gradient and fixed-point complex ICA algorithms based on kurtosis maximization. In *Proc. IEEE Int. Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 85–90, Maynooth, Ireland, Sept. (2006).

70. H. Li and T. Adalı, Optimization in the complex domain for nonlinear adaptive filtering. In *Proc. 33rd Asilomar Conf. on Signals, Systems and Computers*, pp. 263–267, Pacific Grove, CA, Nov. (2006).

71. H. Li and T. Adalı, Complex-valued adaptive signal processing using nonlinear functions. *J. Advances in Signal Processing*, 2008 (Article ID 765615, 9 pages), (2008).

72. H. Li and T. Adalı, Stability analysis of complex maximum likelihood ICA using Wirtinger calculus. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, Las Vegas, NV, April (2008).

73. O. Macchi and E. Moreau, Self-adaptive source separation by direct or recursive networks. In *Proc. Int. Conf. Digital Signal Proc.*, pp. 122–129, Limasol, Cyprus, (1995).

74. J. R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Wiley, (1988).

75. D. Mandic and S. L. Goh, *Complex Valued Nonlinear Adaptive Filters*. Wiley, Chichester, (2009).

76. K.V. Mardia, *Families of Bivariate Distributions*. Griffen, London, (1970).

77. C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. SIAM, Philadephia, PA, (2000).

78. T. P. Minka, Old and new matrix algebra useful for statistics, http://research.microsoft.com/∼minka/papers/matrix, (2000).

79. M. F. Moller, A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6:525–533, (1993).

80. D. R. Morgan, Adaptive algorithms for a two-channel structure employing allpass filters with applications to polarization mode dispersion compensation. *IEEE Trans. Circuits Syst. I: Fund. Theory Apps.*, 51(9):1837–1847, (2004).

81. F. D. Neeser and J. L. Massey, Proper complex random processes with applications to information theory. *IEEE Trans. Info. Theory*, 39:1293–1302, (1993).

82. J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, New York, NY, (2000).

83. M. Novey and T. Adalı, ICA by maximization of nongaussianity using complex functions. In *Proc. IEEE Int. Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 21–26, Mystic, CT, Sept. (2005).

84. M. Novey and T. Adalı, Adaptable nonlinearity for complex maximization of nongaussianity and a fixed-point algorithm. In *Proc. IEEE Int. Workshop on Machine Learning for Signal Processing (MLSP)*, pages 79–84, Maynooth, Ireland, Sept. (2006).

85. M. Novey and T. Adalı, Complex fixed-point ICA algorithm for separation of QAM sources using Gaussian mixture model. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, volume II, pages 445–448, Honolulu, HI, April (2007).

86. M. Novey and T. Adalı, Complex ICA by negentropy maximization. *IEEE Trans. Neural Networks*, 19(4):596–609, (2008).

87. S. Olhede, On probability density functions for complex variables. *IEEE Trans. Info. Theory*, 52:1212–1217, (2006).

88. K. B. Petersen and M. S. Pedersen, *The Matrix cookbook*, Oct. (2008), Version (20081110).

89. D. Pham and P. Garat, Blind separation of mixtures of independent sources through a quasi maximum likelihood approach. *IEEE Trans. Signal Processing*, 45(7):1712–1725, (1997).
90. B. Picinbono, *Random Signals and Systems*, Prentice Hall Signal Processing Series, Englewood Cliffs, NJ, (1993).
91. B. Picinbono, On circularity. *IEEE Trans. Signal Processing*, 42:3473–3482, (1994).
92. B. Picinbono, Second-order complex random vectors and normal distributions. *IEEE Trans. Signal Processing*, 44(10):2637–2640, (1996).
93. B. Picinbono and P. Bondon, Second-order statistics of random signals. *IEEE Trans. Signal Processing*, 45(2):411–419, (1997).
94. B. Picinbono and P. Chevalier, Widely linear estimation with complex data. *IEEE Trans. Signal Processing*, 43:2030–2033, (1995).
95. R. Remmert, *Theory of Complex Functions*. Springer-Verlag, Harrisonburg, VA, (1991).
96. H. Sawada, R. Mukai, S. Araki, and S. Makino, A polar-coordinate based activation function for frequency domain blind source separation. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, pp. 1001–1004, May 2002.
97. A. Sayed, *Fundamentals of Adaptive Filtering*. Wiley-IEEE, Hoboken, NJ, (2003).
98. M. Scarpiniti, D. Vigliano, R. Parisi, and A. Uncini, Generalized splitting functions for blind separation of complex signals. *Neurocomputing*, (2008).
99. R. Schober, W. H. Gerstacker, and L. H.-J. Lampe, A widely linear LMS algorithm for MAI suppression for DS-CDMA. In *Proc. IEEE Int. Conf. on Communications*, pp. 2520–2525, (2003).
100. P. Schreier, Bounds on the degree of improperiety of complex random vectors. *IEEE Signal Proc. Letts.*, 15:190–193, (2008).
101. P. Schreier and L. Scharf, Second-order analysis of improper complex random vectors and processes. *IEEE Trans. Signal Processing*, 51(3):714–725, (2003).
102. P. Schreier, L. Scharf, and A. Hanssen, A generalized likelihood ratio test for impropriety of complex signals. *IEEE Signal Proc. Letts.*, 13(7):433–436, (2006).
103. P. Smaragdis, Blind separation of convolved mixtures in the frequency domain. *Neurocomputing*, 22:21–34, (1998).
104. A. Taleb and C. Jutten, Source separation in post-nonlinear mixtures. *IEEE Trans. Signal Processing*, 47(10):2807–2820, (1999).
105. C. W. Therrien, *Probability for Electrical and Computer Engineers*. CRC Press, Boca Raton, FL, (2004).
106. L. Tong, R.-W. Liu, V. C. Soon, and Y.-F. Huang, Indeterminacy and identifiability of blind identification. *IEEE Trans. Circuits Syst.*, 38(5):499–509, (1991).
107. A. Uncini and F. Piazza, Blind signal processing by complex domain adaptive spline neural networks. *IEEE Trans. Neural Networks*, 14(2):399–412, (2003).
108. A. Uncini, L. Vecci, P. Campolucci, and F. Piazza, Complex-valued neural networks with adaptive spline activation function for digital radio links nonlinear equalization. *IEEE Trans. Signal Processing*, 47(2):505–514, (1999).
109. N. N Vakhania and N. P. Kandelaki, Random vectors with values in complex Hilbert spaces. *Theory Probability Appl.*, 41(1):116–131, (1996).
110. A. van den Bos, Complex gradient and Hessian. *IEE Proc.: Vision, Image, and Signal Processing*, 141(6):380–382, (1994).

111. A. van den Bos, Estimation of complex parameters. In *10th IFAC Symp.*, volume 3, pp. 495–499, (1994).
112. N. Vlassis and Y. Motomura, Efficient source adaptivity in independent component analysis. *IEEE Trans. Neural Networks*, 12(3):559–566, (2001).
113. B. Widrow, J. Cool, and M. Ball, The complex LMS algorithm. *Proc. IEEE*, 63:719–720, (1975).
114. B. Widrow and Jr. M. E. Hopf, Adaptive switching circuits. In *IRE WESCON*, volume 4, pp. 96–104, (1960).
115. W. Wirtinger, Zur formalen theorie der funktionen von mehr komplexen veränderlichen. *Math. Ann.*, 97:357–375, (1927).
116. L. Xu, G. D. Pearlson, and V. D. Calhoun, Joint source based morphometry to identify sources of gray matter and white matter relative differences in schizophrenia versus healthy controls. In *Proc. ISMRM*, Toronto, ON, May (2008).
117. G. Yan and H. Fan, A newton-like algorithm for complex variables with application in blind equalization. *IEEE Trans. Signal Processing*, 48:553–556, (2000).
118. C.-C. Yang and N. K. Bose, Landmine detection and classification with complex-valued hybrid neural network using scattering parameters dataset. *IEEE Trans. Neural Networks*, 16(3):743–753, (2005).
119. C. You and D. Hong, Nonlinear blind equalization schemes using complex-valued multilayer feedforward neural networks. *IEEE Trans. Neural Networks*, 9(6):1442–1455, (1998).
120. L. Zhang, A. Cichocki, and S.-I. Amari, Self-adaptive blind source separation based on activation functions adaptation. *IEEE Trans. Neural Networks*, 15(2):233–244, (2004).