Neural Networks 61 (2015) 59–67

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

A complex-valued neural dynamical optimization approach and its stability analysis☆

Songchuan Zhang [a], Youshen Xia [b,*], Weixing Zheng [c]

[a] *Center for Discrete Mathematics and Theoretical Computer Science, Fuzhou University, Fuzhou, China*

[b] *College of Mathematics and Computer Science, Fuzhou University, China*

[c] *School of Computing, Engineering and Mathematics, University of Western Sydney, Australia*

A R T I C L E   I N F O

Article history:
Received 25 April 2014
Received in revised form 4 September 2014
Accepted 2 October 2014
Available online 19 October 2014

Keywords:
Complex-valued neural dynamical system
$\mathbb{CR}$ Calculus
Nonlinear convex programming
Complex variables
Global stability analysis

A B S T R A C T

In this paper, we propose a complex-valued neural dynamical method for solving a complex-valued nonlinear convex programming problem. Theoretically, we prove that the proposed complex-valued neural dynamical approach is globally stable and convergent to the optimal solution. The proposed neural dynamical approach significantly generalizes the real-valued nonlinear Lagrange network completely in the complex domain. Compared with existing real-valued neural networks and numerical optimization methods for solving complex-valued quadratic convex programming problems, the proposed complex-valued neural dynamical approach can avoid redundant computation in a double real-valued space and thus has a low model complexity and storage capacity. Numerical simulations are presented to show the effectiveness of the proposed complex-valued neural dynamical approach.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

In many engineering problems, the unknown variables are complex-valued vectors and a main task is to find these variables by minimization of a complex-valued optimization problem with constraints. Applications of the complex-valued optimization problem can be found in communications, adaptive filtering, medical imaging, and robot control, etc. (Loesch & Yang, 2013; Lustig, Donoho, & Pauly, 2007; Vakhania & Kandelaki, 1996; Xia & Feng, 2006; Zhang & Ma, 1997). Complex domain has distinctive advantages. For example, the XOR problem in the real domain cannot be solved with a single real-valued neuron, but for the same task in the complex domain, only one complex-valued is sufficient (Aizenberg, 2008). By using complex variables to estimate derivatives of real functions, the accuracy is increased by an order of magnitude (Squire & Trapp, 1998). In communications, array signal processing, beamforming, etc., complex domain not only provides a convenient representation for these signals, but also a natural way to preserve the phase information of the signals. While the phase information in 2-D signals such as images, the role of the phase of a signal is more obvious (Oppenheim & Lim, 1981). For example, the phase information can be used to improve the perceptual side of modeling (Mandic, Souretis, Javidi, Goh, & Tanaka, 2007). The simultaneous modeling of the "intensity" and "direction" components of vector field processes in the complex domain $\mathbb{C}$, (such as radar, sonar, vector fields, wind modeling) where the phase information can be accounted for naturally, has advantages over the so called "dual univariate" modeling (where the components of such processes are treated as independent random processes). For example, in modeling of wind profile (Goh et al., 2006), the fully complex approach (simultaneously speed and direction as a complex vector) has major benefits over the dual univariate approach (speed and direction as independent processes). In addition, signal processing in complex domain possesses several distinctive features which are not present in real domain, such as more powerful statistics and complex nonlinearity. A systematic view of the duality between the processing in $\mathbb{R}$ and $\mathbb{C}$ for several classes of real world processes has been provided in a recent book by Mandic and Goh (2009). Therefore, complex-valued optimization has been widely used for general signal processing with Fourier synthesis or in frequency-domain treatment through Fourier transform. The complex domain not only provides a convenient representation but also a natural way to preserve the physical characteristics of the original problem. On the other hand, it is more difficult to deal with the complex-valued problem than the real-valued problem

☆ This work was supported by the National Natural Science Foundation of China under Grant Nos. 61179037 and 61473330, and the Doctoral Project of the Ministry of Education of China under Grant No. 20133514110010, and in part by a research grant from the University of Western Sydney, Australia.

* Corresponding author.
E-mail addresses: ysxia@fzu.edu.cn (Y.S. Xia), w.zheng@uws.edu.au (W.X. Zheng).

http://dx.doi.org/10.1016/j.neunet.2014.10.003
0893-6080/© 2014 Elsevier Ltd. All rights reserved.

in analytic function theory (Brandwood, 1983; Van den Bos, 1994). Traditional numerical optimization methods are used to solve real-valued optimization problems and cannot be directly applied to complex-valued optimization problems (Boyd & Vandenberghe, 2006). In order to solve the optimization problems over the complex field, it is required to convert a complex-valued optimization problem into a real-valued one by splitting the complex numbers into their real and imaginary parts. However, the major disadvantage of this method is that the resulting algorithm will double the dimension compared with the original problem and may break the special data structure. Moreover, they will suffer from high computational complexity and have a slow convergence when the problem size is large.

During recent decades, real-valued neural dynamical methods for solving optimization problems have been a major area in neural network research. For example, Zhang and Constantinides (1992) proposed a Lagrange programming neural network for nonlinear program with equality constants; Cichocki and Unbehauen (1993) presented several neural networks for linear and nonlinear program; Chong, Hui, and Zak (1999) developed a linear programming neural network; Perfetti and Todisco (2008) presented a quasi-Lagrangian neural network for convex quadratic optimization; Forti, Nistri, and Quincampoix (2004, 2006) analyzed a generalized neural network for nonsmooth nonlinear programming problems; Xia, Feng, and Kamel (2007) developed a neural dynamical approach to solving nonlinear programming problems. Mathematically, the neural dynamical approach converts an optimization problem into a dynamical system so that whose state output will give the optimal solution of the optimization problem and then the optimal solution can be obtained by tracking the state trajectory of the designed dynamical system based on hardware and software implementation. The neural dynamical approach has an advantage over the traditional ordinary differential equation method (Chong et al., 1999; Cichocki & Unbehauen, 1993; Xia et al., 2007) in that the designed neural dynamical system is suitable for the recurrent neural model with input, output, artificial neurons, and activation functions. Therefore, it has a low computational complexity and is suitable for parallel implementation. Compared with the conventional numerical optimization method, the computation path of the neural dynamical approach is continuous instead of discrete, although they share some fundamental optimization formulations. Moreover, because of the inherent dynamical nature, the neural dynamical approach has a potential capacity in solving the optimization problems with weak condition for the global convergence (Hu & Wang, 2012; Lu & Wang, 2008; Wang, Hu, & Jiang, 1999; Zeng & Wang, 2008). Although complex-valued neural networks (CVNN for short) have been developed for engineering applications (Aizenberg, Dmitriy, & Zurada, 2007; Goh & Mandic, 2004, 2007; Hanna & Mandic, 2003; Hirose, 2004; Hirose & Yoshida, 2012; Nitta, 2009) in recent years, there are only several papers on dealing with complex-valued quadratic convex programming problems. A discrete-time complex-valued neural network model for quadratic optimization problems with bounded variables was studied by Nitta (2009) and a complex-valued algorithm for unconstrained optimization of real functions in complex variables was presented by Sorber, Barel, and Lathauwer (2012). Two real-valued neural networks (Tan, Shi, & Tan, 2010; Xia & Feng, 2006) were presented for solving complex-valued quadratic convex programming problems. The complex-valued quadratic convex programming problem is first converted into a real-valued one by splitting the complex numbers into their real and imaginary parts and then they are mapped into a real-valued neural network with larger model size, respectively. As a result, the two neural networks have high model complexity and slow convergence for some high-dimensional optimization problems. So, it is very desirable to develop complex-valued algorithms for solving constrained optimization problems of real functions in complex variables.

In order to reduce computational model complexity, we propose a complex-valued neural dynamical method for solving complex-valued nonlinear convex programming problems. The proposed neural dynamical approach significantly generalizes the real-valued nonlinear Lagrange network completely in the complex domain. Compared with existing real-valued neural networks and numerical optimization methods for solving complex-valued quadratic convex programming problems, the proposed complex-valued neural dynamical approach can avoid redundant computation in a double real-valued space and thus has a low model complexity and storage capacity. Theoretically, we prove that the proposed complex-valued dynamical approach is globally stable and convergent to the optimal solution. Compared with the classic result on the stability of nonlinear systems, we obtain the results of global stability of nonlinear systems, instead of local stability. Compared with stability results of the real-valued neural dynamical approach, our results establish in the complex domain, instead of the real domain. Numerical simulations confirm the computational efficiency of the proposed complex-valued dynamical approach.

The rest of this paper is organized as follows: Section 2 describes complex-valued nonlinear programming problems, provides mathematical foundations for the $\mathbb{CR}$ calculus, and introduces a complex-valued neural dynamical approach. In Section 3, we give some complex preliminaries. In Section 4, we analyze the global convergence of the proposed complex-valued neural dynamical approach. In Section 4, simulation comparisons are given to show computational efficiency of the proposed complex-valued neural dynamical algorithm. Finally, the paper is concluded in Section 5.

2. Complex-valued nonlinear programming and neural dynamical approach

We are concerned with the following complex-valued nonlinear convex programming problem:

$$\begin{array}{ll} \min & g(\mathbf{z}) \\ \text{s.t.} & \mathbf{z} \in \boldsymbol{\Omega} \end{array} \tag{1}$$

where $g(\mathbf{z}) : \Omega \subset \mathbb{C}^n \to \mathbb{R}$, $\Omega = \{\mathbf{z} \in \mathbb{C}^n | \mathbf{A}\mathbf{z} = \mathbf{b}\}$, $\mathbf{b} \in \mathbb{C}^m$, and $\mathbf{A} \in \mathbb{C}^{m\times n}$. A point $\mathbf{z}$ is feasible if $\mathbf{z} \in \Omega$. A feasible point $\hat{\mathbf{z}}$ is locally optimal if there exists an open ball around $\hat{\mathbf{z}}$, $B_\epsilon(\hat{\mathbf{z}}) \subseteq \Omega$ such that

$$g(\mathbf{z}) \geq g(\hat{\mathbf{z}}), \quad \forall \mathbf{z} \in B_\epsilon(\hat{\mathbf{z}}) \cap \Omega$$

and $\hat{\mathbf{z}}$ is globally optimal if $g(\mathbf{z}) \geq g(\hat{\mathbf{z}})$, $\forall \mathbf{z} \in \Omega$. For our discussion, we assume that (1) has a global optimal solution.

In complex-variables optimization problems, the functions of interest, such as $g(\mathbf{z})$ in problem (1), are not analytic. That is, the Cauchy–Riemann (C–R) conditions are not satisfied. To overcome this difficulty, a relaxed $\mathbb{CR}$ Calculus (also known as Wirtinger Calculus (Remmert, 1991) and Brandwood's result (Brandwood, 1983)) is introduced by using the duality between the space $\mathbb{C}$ and $\mathbb{R}^2$ and the conjugate coordinate $\bar{\mathbf{z}}$ (Kreutz-Delgado, 2006; Mandic, Still, & Douglas, 2009). Let $\mathbf{z} = \mathbf{x} + i\mathbf{y}$ where $i = \sqrt{-1}$ is the imaginary unit. Then $g(\mathbf{z})$ can be viewed as a real bivariate function of its real and imaginary components $f(\mathbf{x}, \mathbf{y}) : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$. On the other side, from $x = (\mathbf{z} + \bar{\mathbf{z}})/2$ and $y = -i(\mathbf{z} - \bar{\mathbf{z}})/2$ we see that $g(\mathbf{z}) = g(\mathbf{z}, \bar{\mathbf{z}})$ where $\bar{\mathbf{z}}$ represents the complex conjugate. Assume that the partial derivatives of $f(\mathbf{x}, \mathbf{y})$, $\frac{\partial f}{\partial \mathbf{x}}$ and $\frac{\partial f}{\partial \mathbf{y}}$, exist. Then $\mathbb{R}$-derivative of a real function of a complex variable $g(\mathbf{z}) = g(\mathbf{z}, \bar{\mathbf{z}})$ is given by

$$\frac{\partial g}{\partial \mathbf{z}} = \frac{1}{2}\left(\frac{\partial f}{\partial \mathbf{x}} - i\frac{\partial f}{\partial \mathbf{y}}\right),$$

and the conjugate $\mathbb{R}$-derivative ($\mathbb{R}^*$-derivative) of a function $g(\mathbf{z}) = g(\mathbf{z}, \bar{\mathbf{z}})$ is given by

$$\frac{\partial g}{\partial \bar{\mathbf{z}}} = \frac{1}{2}\left(\frac{\partial f}{\partial \mathbf{x}} + i\frac{\partial f}{\partial \mathbf{y}}\right).$$

For a complex analytic function, the $\mathbb{R}$-derivative is equivalent to the standard $\mathbb{C}$-derivative and the $\mathbb{R}^*$-derivative vanishes. For further introduction about $\mathbb{CR}$ Calculus, see an excellent overview by Kreutz-Delgado (2006), and Mandic and Goh (2009). A comprehensive account of complex vector and matrix differentiation is given in Hjorungnes and Gesbert (2007) and Javidi, Mandic, and Kuh (2010).

Throughout this paper we suppose that the objective function $g(\mathbf{z})$ is analytic with respect to $\mathbf{z}$ and $\bar{\mathbf{z}}$ independently. For real functions of complex variables, the complex gradient is defined as Brandwood (1983)

$$\nabla g = \frac{\partial g}{\partial \bar{\mathbf{z}}}. \tag{2}$$

Similar to the Lagrangian technique of the real-valued optimization method (Perfetti & Todisco, 2008; Zhang & Constantinides, 1992), we define a complex-valued Lagrangian function of (1) as

$$L(\mathbf{z}, \mathbf{u}) = g(\mathbf{z}) - \mathbf{Re}(2\mathbf{u}^H(\mathbf{A}\mathbf{z} - \mathbf{b}))$$

where $\mathbf{u} \in \mathbb{C}^n$ is Lagrange multiplier and the superscript $(.)^H$ represents the Hermitian transpose. By (2) we have for any $\mathbf{d} \in C^n$

$$\begin{aligned}\frac{\partial}{\partial \bar{\mathbf{z}}}\mathbf{Re}(\mathbf{d}^H\mathbf{z}) &= \frac{\partial}{\partial \bar{\mathbf{z}}}((\mathbf{d}_R - i\mathbf{d}_I)^T(\mathbf{x} + i\mathbf{y})) = \frac{\partial}{\partial \bar{\mathbf{z}}}(\mathbf{d}_R^T\mathbf{x} + \mathbf{d}_I^T\mathbf{y}) \\ &= \frac{1}{2}\left(\frac{\partial}{\partial \mathbf{x}}(\mathbf{d}_R^T\mathbf{x} + \mathbf{d}_I^T\mathbf{y}) + i\frac{\partial}{\partial \mathbf{y}}(\mathbf{d}_R^T\mathbf{x} + \mathbf{d}_I^T\mathbf{y})\right) \\ &= \frac{1}{2}(\mathbf{d}_R + i\mathbf{d}_I) = \mathbf{d}/2.\end{aligned}$$

Then

$$\begin{aligned}&\frac{\partial}{\partial \bar{\mathbf{z}}}(g(\mathbf{z}) - \mathbf{Re}(2\mathbf{u}^H(\mathbf{A}\mathbf{z} - \mathbf{b}))) = \nabla g(\mathbf{z}) - \frac{\partial}{\partial \bar{\mathbf{z}}}\mathbf{Re}(2\mathbf{u}^H(\mathbf{A}\mathbf{z} - \mathbf{b})) \\ &= \nabla g(\mathbf{z}) - \frac{\partial}{\partial \bar{\mathbf{z}}}\mathbf{Re}(2(\mathbf{A}^H\mathbf{u})^H\mathbf{z}) = \nabla g(\mathbf{z}) - \mathbf{A}^H\mathbf{u}.\end{aligned}$$

Similarly,

$$\begin{aligned}\frac{\partial}{\partial \bar{\mathbf{u}}}(g(\mathbf{z}) - \mathbf{Re}(2\mathbf{u}^H(\mathbf{A}\mathbf{z} - \mathbf{b}))) &= -\frac{\partial}{\partial \bar{\mathbf{u}}}\mathbf{Re}(2\mathbf{u}^H(\mathbf{A}\mathbf{z} - \mathbf{b})) \\ &= -\mathbf{A}\mathbf{z} + \mathbf{b}.\end{aligned}$$

Then

$$\begin{aligned}&\nabla L_{\mathbf{z}}(\mathbf{z}, \mathbf{u}) = \nabla g(\mathbf{z}) - \mathbf{A}^H\mathbf{u} \\ &\nabla L_{\mathbf{u}}(\mathbf{z}, \mathbf{u}) = -\mathbf{A}\mathbf{z} + \mathbf{b}.\end{aligned} \tag{3}$$

Based on (3), we propose a complex-valued neural dynamical approach as follows

State equation

$$\frac{d}{dt}\begin{pmatrix}\mathbf{z}(t) \\ \mathbf{u}(t)\end{pmatrix} = -\lambda\begin{pmatrix}\nabla g(\mathbf{z}(t)) - \mathbf{A}^H\mathbf{u}(t) \\ \mathbf{A}\mathbf{z}(t) - \mathbf{b}\end{pmatrix}. \tag{4}$$

Output equation

$$\mathbf{z}_{\text{out}}(t) = \mathbf{z}(t)$$

where $(\mathbf{z}(t), \mathbf{u}(t)) \in \mathbb{C}^{n+m}$ is the state vector, $\mathbf{z}_{\text{out}}(t)$ is the output vector, and $\lambda > 0$ is a design constant. Under the condition that $g(\mathbf{z})$ is analytic with respect to $\mathbf{z}$ and $\bar{\mathbf{z}}$ independently, the real and imaginary part of $\nabla g(\mathbf{z})$ are differentiable. Thus there exists one solution of the dynamical equation in (4) with any given initial point.

We now give a comparison. First, when the state vector $(\mathbf{z}(t), \mathbf{u}(t))$, the vector $\mathbf{b}$, and the matrix $\mathbf{A}$ defined in (4) are all real, the proposed complex-valued neural dynamical system in (4) becomes the real-valued neural dynamical system, called the Lagrangian neural network, given in Perfetti and Todisco (2008) and Zhang and Constantinides (1992). Thus the proposed complex-valued neural dynamical approach is a significant extension of the real-valued neural dynamical approach from the real domain to the complex domain. Next, two real-valued neural network approaches were presented to solving a complex-valued quadratic convex programming problem. Because they have to split the complex numbers into their real and imaginary parts and then are mapped into an argument real-valued neural network (Tan et al., 2010; Xia & Feng, 2006), they require $2(n+m)$ neurons and $2(m+n) \times 2(m+n)$ connection weight. By contrast, the proposed complex-valued neural dynamical approach requires $(n+m)$ neurons and $(m+n) \times (m+n)$ complex connection weight only. So, the proposed complex-valued neural dynamical approach has a lower model complexity and storage capacity than one of the existing real-valued neural dynamical approach.

Finally, for analysis discussion in next section, we give preliminaries on functions of complex variables. A set Ω is convex if for any two points $\mathbf{z}_1$ and $\mathbf{z}_2$ in Ω and any θ with $0 \le \theta \le 1$, we have

$$\theta\mathbf{z}_1 + (1-\theta)\mathbf{z}_2 \in \Omega.$$

A real-valued function of a complex vector $\mathbf{z}$ $g(\mathbf{z}) : \mathbb{C}^n \mapsto \mathbb{R}$ defined on convex set $\Omega \subset \mathbb{C}^n$ is called convex, if for any two points $\mathbf{z}_1$ and $\mathbf{z}_2$ in Ω and any t in [0, 1],

$$g(t\mathbf{z}_1 + (1-t)\mathbf{z}_2) \le tg(\mathbf{z}_1) + (1-t)g(\mathbf{z}_2).$$

A function is called strictly convex if the above inequality holds strictly. The following two linear mappings from the complex space to the real space are needed in our analysis:

(1) For $\forall \mathbf{z} \in \mathbb{C}^n$, let

$$\varphi_1(\mathbf{z}) = \begin{pmatrix}\mathbf{Re}(\mathbf{z}) \\ \mathbf{Im}(\mathbf{z})\end{pmatrix} \in \mathbb{R}^{2n}.$$

(2) For $\forall \mathbf{A} \in \mathbb{C}^{m\times n}(n > 1)$, let

$$\varphi_2(\mathbf{A}) = \begin{pmatrix}\mathbf{Re}(\mathbf{A}) & -\mathbf{Im}(\mathbf{A}) \\ \mathbf{Im}(\mathbf{A}) & \mathbf{Re}(\mathbf{A})\end{pmatrix} \in \mathbb{R}^{2m\times 2n}$$

where $\mathbf{Re}(\,)$ and $\mathbf{Im}(\,)$ denote the real and imaginary parts of a complex matrix or vector. It is easy to see that the two mappings φ_1 and φ_2 are the one-to-one mapping from the complex space to the real matrix space, respectively. Moreover,

$$\varphi_1(\nabla g) = \frac{1}{2}\begin{pmatrix}\dfrac{\partial f}{\partial \mathbf{x}} \\ \dfrac{\partial f}{\partial \mathbf{y}}\end{pmatrix} = \frac{1}{2}\nabla f(\mathbf{w})$$

where $\mathbf{w} = \binom{\mathbf{x}}{\mathbf{y}} = \varphi_1(\mathbf{z})$ and $\nabla f(\mathbf{w}) = \frac{\partial f}{\partial \mathbf{w}}$. Furthermore, in the Appendix we will show that they have the following useful properties: for any $\mathbf{A} \in \mathbb{C}^{m\times n}$, $\mathbf{B} \in \mathbb{C}^{n\times k}$, $\mathbf{D} \in \mathbb{C}^{n\times n}$ where $\mathbf{D}$ is invertible matrix, $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{C}^n$, $\sigma \in \mathbb{R}$, then

(1) $\varphi_1(\mathbf{z}_1 \pm \mathbf{z}_2) = \varphi_1(\mathbf{z}_1) \pm \varphi_1(\mathbf{z}_2)$; $\varphi_1(\sigma\mathbf{z}) = \sigma\varphi_1(\mathbf{z})$

(2) $\varphi_2(\mathbf{A} \pm \mathbf{B}) = \varphi_2(\mathbf{A}) \pm \varphi_2(\mathbf{B})$; $\varphi_2(\sigma\mathbf{A}) = \sigma\varphi_2(\mathbf{A})$; $\varphi_2(\mathbf{I}_n) = \mathbf{I}_{2n}$

(3) $\varphi_2(\mathbf{A}\cdot\mathbf{B}) = \varphi_2(\mathbf{A})\cdot\varphi_2(\mathbf{B})$; $\varphi_1(\mathbf{A}\mathbf{z}) = \varphi_2(\mathbf{A})\varphi_1(\mathbf{z})$;

(4) $\varphi_2(\mathbf{A}^H) = (\varphi_2(\mathbf{A}))^T$; $\varphi_2(\mathbf{D}^{-1}) = (\varphi_2(\mathbf{D}))^{-1}$;

(5) $\|\mathbf{z}\|_2^2 = \|\varphi_1(\mathbf{z})\|_2^2$; $\mathbf{Re}(\mathbf{z}_1^H\mathbf{z}_2) = \varphi_1(\mathbf{z}_1)^T\varphi_1(\mathbf{z}_2)$

where the superscript $(.)^T$ represents the transpose and $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ is a unit matrix.

3. Main results

The analysis results of the real-valued Lagrangian dynamical system have been obtained by Perfetti and Todisco (2008), Xia (2003) and Zhang and Constantinides (1992), respectively. But there is no theoretical analysis result of the complex-valued neural dynamical system. In this section, we will fill this gap by analyzing stability and convergence of the complex-valued neural dynamical system.

Proposition 1. *If $g(\mathbf{z})$ is convex on convex set $\Omega \subset \mathbb{C}^n$, then $f(\mathbf{w})$ is convex on convex set $\tilde{\Omega} \overset{\text{def}}{=} \{\mathbf{w} \in \mathbb{R}^{2n} : \mathbf{w} = \varphi_1(\mathbf{z}), \mathbf{z} \in \Omega\}$.*

Proof. First, we prove the set $\tilde{\Omega}$ is convex in $\mathbb{R}^{2n}$. For any two points $\mathbf{w}_1, \mathbf{w}_2$ in $\tilde{\Omega}$ and any θ with $0 \le \theta \le 1$, by definition of φ_1, we can obtain unique corresponding $\mathbf{z}_1$ and $\mathbf{z}_2$ which satisfy $\mathbf{w}_1 = \varphi_1(\mathbf{z}_1)$ and $\mathbf{w}_2 = \varphi_1(\mathbf{z}_2)$. Let

$$\mathbf{w} = \theta\mathbf{w}_1 + (1-\theta)\mathbf{w}_2,$$

then taking φ_1^{-1} on both sides of the above equalities, we have

$$\varphi_1^{-1}(\mathbf{w}) = (\theta\mathbf{z}_1 + (1-\theta)\mathbf{z}_2) \overset{\text{def}}{=} \mathbf{z} \in \Omega.$$

Thus, we can easily know $\mathbf{w} \in \tilde{\Omega}$ and the convexity of $\tilde{\Omega}$ is proved.

Next, we prove that $f(\mathbf{w})$ is convex.

We know

$$f(\theta\mathbf{w}_1 + (1-\theta)\mathbf{w}_2) = f(\mathbf{w}) = g(\mathbf{z}),$$
$$\theta f(\mathbf{w}_1) + (1-\theta)f(\mathbf{w}_2) = \theta g(\mathbf{z}_1) + (1-\theta)g(\mathbf{z}_2).$$

Since $g(\mathbf{z})$ is convex on convex set Ω, we have

$$g(\mathbf{z}) = g(\theta\mathbf{z}_1 + (1-\theta)\mathbf{z}_2) \le \theta g(\mathbf{z}_1) + (1-\theta)g(\mathbf{z}_2).$$

We can further know

$$f(\theta\mathbf{w}_1 + (1-\theta)\mathbf{w}_2) \le \theta f(\mathbf{w}_1) + (1-\theta)f(\mathbf{w}_2).$$

Hence, $f(\mathbf{w})$ is convex on convex set $\tilde{\Omega}$. □

Proposition 2. *The equilibrium point of the proposed complex-valued neural dynamical system in (4) can correspond to the optimal solution of (1).*

Proof. We need to prove that (1) has the following Lagrange optimality conditions:

$$\nabla g(\mathbf{z}) - \mathbf{A}^H\mathbf{u} = \mathbf{0}, \qquad \mathbf{A}\mathbf{z} = \mathbf{b}. \tag{5}$$

By the properties of the introduced two mappings we see that (1) is equivalent to the following real-valued nonlinear programming problem:

$$\begin{aligned} \min \quad & f(\mathbf{w}) \\ \text{s.t.} \quad & \varphi_2(\mathbf{A})\mathbf{w} = \varphi_1(\mathbf{b}) \end{aligned} \tag{6}$$

where $\mathbf{w} \in \mathbb{R}^{2n}$, $\mathbf{w} = \varphi_1(\mathbf{z})$, and $f(\mathbf{w}) = g(\mathbf{z})$.

Now, by Proposition 1 we know that $f(\mathbf{w})$ is convex and thus the following Lagrange condition of (6) is optimal:

$$\frac{\partial f(\mathbf{w})}{\partial \mathbf{w}} - \varphi_2(\mathbf{A})^T\lambda = \mathbf{0}, \qquad \varphi_2(\mathbf{A})\mathbf{w} = \varphi_1(\mathbf{b}) \tag{7}$$

where $\lambda \in \mathbb{R}^{2m}$ is the optimal Lagrange multiplier. Since

$$\varphi_1\left(\frac{\partial g}{\partial \bar{\mathbf{z}}}\right) = \frac{1}{2}\frac{\partial f}{\partial \mathbf{w}},$$

we have

$$2\varphi_1\left(\frac{\partial g}{\partial \bar{\mathbf{z}}}\right) - \varphi_2(\mathbf{A}^H)\lambda = \mathbf{0}, \qquad \varphi_2(\mathbf{A})\mathbf{w} = \varphi_1(\mathbf{b}).$$

Taking φ_1^{-1} and φ_2^{-1} on both sides of the above equalities, respectively, we obtain that

$$\frac{\partial g}{\partial \bar{\mathbf{z}}} - \frac{1}{2}\mathbf{A}^H\varphi_1^{-1}(\lambda) = \mathbf{0}, \qquad \mathbf{A}\mathbf{z} = \mathbf{b}. \tag{8}$$

Let $\mathbf{u} = \frac{1}{2}\varphi_1^{-1}(\lambda) \in \mathbb{C}^n$. Then (8) becomes the form of (5). It implies that (5) is the Lagrange optimality condition of (1). Therefore, the equilibrium point of the proposed complex-valued neural dynamical system corresponds to the optimal solution of (1). □

Proposition 3. *The proposed complex-valued neural dynamical system is equal to*

$$\frac{d}{dt}\begin{pmatrix}\mathbf{w}\\ \mathbf{k}\end{pmatrix} = -\lambda\begin{pmatrix}\frac{1}{2}\nabla f - \mathbf{\Lambda}^T\mathbf{k}\\ \mathbf{\Lambda}\mathbf{w} - \boldsymbol{\beta}\end{pmatrix} \tag{9}$$

where $\mathbf{\Lambda} = \varphi_2(\mathbf{A})$, $\boldsymbol{\beta} = \varphi_1(\mathbf{b})$ and $\mathbf{k} = \varphi_1(\mathbf{u})$.

Proof. Let $\mathbf{u} = \mathbf{u}_R + i\mathbf{u}_I$ where $\mathbf{u}_R \in \mathbb{R}^m$ and $\mathbf{u}_I \in \mathbb{R}^m$. Since

$$\frac{d\mathbf{z}}{dt} = \frac{d\mathbf{x}}{dt} + i\frac{d\mathbf{y}}{dt} = -\lambda(\nabla g(\mathbf{z}) - \mathbf{A}^H\mathbf{u})$$
$$\frac{d\mathbf{u}}{dt} = \frac{d\mathbf{u}_R}{dt} + i\frac{d\mathbf{u}_I}{dt} = -\lambda(\mathbf{A}\mathbf{z} - \mathbf{b}).$$

Taking φ_1 on both sides of the above equalities, we obtain that

$$\frac{d}{dt}\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} = -\lambda\left(\frac{1}{2}\frac{\partial f}{\partial \mathbf{w}} - (\varphi_2(\mathbf{A}))^T\varphi_1(\mathbf{u})\right)$$
$$\frac{d}{dt}\begin{pmatrix}\mathbf{u}_R\\ \mathbf{u}_I\end{pmatrix} = -\lambda(\varphi_2(\mathbf{A})\mathbf{w} - \varphi_1(\mathbf{b})).$$

Let $\mathbf{w} = \begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} = \varphi_1(\mathbf{z})$, $\mathbf{k} = \begin{pmatrix}\mathbf{u}_R\\ \mathbf{u}_I\end{pmatrix} = \varphi_1(\mathbf{u})$, $\mathbf{\Lambda} = \varphi_2(\mathbf{A})$ and $\boldsymbol{\beta} = \varphi_1(\mathbf{b})$. Then the proposed complex-valued neural dynamical system can be expressed as (9). □

Assume that $g(\mathbf{z})$ is twice differentiable with respect to $\mathbf{z}$ and $\bar{\mathbf{z}}$. The complex Hessian matrix of g with respect to complex vectors $\mathbf{z}$ and $\bar{\mathbf{z}}$ is defined as Hualiang and Adali (2006) and Kreutz-Delgado (2006)

$$\nabla^2 g = \frac{\partial^2 g}{\partial \bar{\mathbf{v}}\partial \mathbf{v}^T} \in \mathbb{C}^{2n\times 2n}$$

where $\mathbf{v} = \begin{pmatrix}\mathbf{z}\\ \bar{\mathbf{z}}\end{pmatrix} \in \mathbb{C}^{2n}$. The definition may be viewed as a variation of the definition proposed by Van den Bos (1994). Furthermore, the complex Hessian matrix of g can be expressed as:

$$\nabla^2 g = \begin{pmatrix}\mathbf{H}_1 & \mathbf{H}_2\\ \bar{\mathbf{H}}_2 & \bar{\mathbf{H}}_1\end{pmatrix}$$

where

$$\mathbf{H}_1 = \frac{\partial^2 g}{\partial \bar{\mathbf{z}}\partial \mathbf{z}^T} \in \mathbb{C}^{n\times n}, \qquad \mathbf{H}_2 = \frac{\partial^2 g}{\partial \bar{\mathbf{z}}\partial \mathbf{z}^H} \in \mathbb{C}^{n\times n}.$$

Similar to Kreutz-Delgado (2006), the following result reveals the relationship between $\nabla^2 g(\mathbf{z})$ and the Hessian matrix of f, $\nabla^2 f(\mathbf{w})$.

Proposition 4. *Assume that $g(\mathbf{z})$ is twice differentiable with respect to the complex-valued variable $\mathbf{z}$ and $\bar{\mathbf{z}}$, independently. Then*

$$\nabla^2 f = \mathbf{J}^H\begin{pmatrix}\mathbf{H}_1 & \mathbf{H}_2\\ \bar{\mathbf{H}}_2 & \bar{\mathbf{H}}_1\end{pmatrix}\mathbf{J} \tag{10}$$

where

$$\mathbf{J} = \begin{pmatrix}\mathbf{I}_n & i\mathbf{I}_n\\ \mathbf{I}_n & -i\mathbf{I}_n\end{pmatrix} \in \mathbb{C}^{2n\times 2n}.$$

Proof. First, note that $\mathbf{w} = \begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix} = \varphi_1(\mathbf{z})$, $\mathbf{v} = \mathbf{J}\mathbf{w}$, $\mathbf{J}^{-1} = \frac{1}{2}\mathbf{J}^H$, and $\mathbf{w} = \frac{1}{2}\mathbf{J}^T\bar{\mathbf{v}}$. We have

$$\nabla f = \mathbf{J}^H\frac{\partial g}{\partial \bar{\mathbf{v}}} \tag{11}$$

and then

$$\nabla^2 f = \frac{\partial}{\partial \mathbf{w}^T}(\nabla f) = \frac{\partial}{\partial \mathbf{w}^T}\left(\mathbf{J}^H \frac{\partial g}{\partial \bar{\mathbf{v}}}\right)$$
$$= J^H \frac{\partial}{\partial \mathbf{v}^T}\left(\frac{\partial g}{\partial \bar{\mathbf{v}}}\right)\frac{\partial \mathbf{v}}{\partial \mathbf{w}^T} = \mathbf{J}^H \nabla^2 g \mathbf{J}.$$

Next, since

$$\frac{\partial g}{\partial \bar{\mathbf{v}}} = \begin{pmatrix} \frac{\partial g}{\partial \bar{\mathbf{z}}} \\ \frac{\partial g}{\partial \mathbf{z}} \end{pmatrix},$$

we have

$$\frac{\partial^2 g}{\partial \bar{\mathbf{v}} \partial \mathbf{v}^T} = \begin{pmatrix} \frac{\partial^2 g}{\partial \bar{\mathbf{z}} \partial \mathbf{z}^T} & \frac{\partial^2 g}{\partial \bar{\mathbf{z}} \partial \mathbf{z}^H} \\ \frac{\partial^2 g}{\partial \mathbf{z} \partial \mathbf{z}^T} & \frac{\partial^2 g}{\partial \mathbf{z} \partial \mathbf{z}^H} \end{pmatrix}.$$

Based on the $\mathbb{CR}$-derivatives (Brandwood, 1983; Van den Bos, 1994), the following rules of complex differentiation hold:

$$\overline{\left(\frac{\partial g}{\partial \mathbf{z}}\right)} = \left(\frac{\partial \bar{g}}{\partial \bar{\mathbf{z}}}\right), \qquad \left(\frac{\partial g}{\partial \bar{\mathbf{z}}}\right) = \overline{\left(\frac{\partial \bar{g}}{\partial \mathbf{z}}\right)}.$$

Since $g(\mathbf{z})$ is a real function of complex variable, we also have

$$\overline{\left(\frac{\partial g}{\partial \mathbf{z}}\right)} = \frac{\partial g}{\partial \bar{\mathbf{z}}}.$$

Hence,

$$\frac{\partial^2 g}{\partial \bar{\mathbf{z}} \partial \mathbf{z}^T} = \frac{\partial}{\partial \mathbf{z}^T}\left(\frac{\partial g}{\partial \bar{\mathbf{z}}}\right) = \frac{\partial}{\partial \mathbf{z}^T}\overline{\left(\frac{\partial g}{\partial \mathbf{z}}\right)}$$
$$= \overline{\left(\frac{\partial}{\partial \mathbf{z}^H}\left(\frac{\partial g}{\partial \mathbf{z}}\right)\right)} = \overline{\left(\frac{\partial^2 g}{\partial \mathbf{z} \partial \mathbf{z}^H}\right)}.$$

Similarly, we also have

$$\frac{\partial^2 g}{\partial \bar{\mathbf{z}} \partial \mathbf{z}^H} = \frac{\partial}{\partial \mathbf{z}^H}\left(\frac{\partial g}{\partial \bar{\mathbf{z}}}\right) = \frac{\partial}{\partial \mathbf{z}^H}\overline{\left(\frac{\partial g}{\partial \mathbf{z}}\right)}$$
$$= \overline{\left(\frac{\partial}{\partial \mathbf{z}^T}\left(\frac{\partial g}{\partial \mathbf{z}}\right)\right)} = \overline{\left(\frac{\partial^2 g}{\partial \mathbf{z} \partial \mathbf{z}^T}\right)}.$$

Let

$$\mathbf{H}_1 = \frac{\partial^2 g}{\partial \bar{\mathbf{z}} \partial \mathbf{z}^T} \in \mathbb{C}^{n\times n}, \qquad \mathbf{H}_2 = \frac{\partial^2 g}{\partial \bar{\mathbf{z}} \partial \mathbf{z}^H} \in \mathbb{C}^{n\times n}.$$

Then

$$\nabla^2 f = \mathbf{J}^H \begin{pmatrix} \mathbf{H}_1 & \mathbf{H}_2 \\ \bar{\mathbf{H}}_2 & \bar{\mathbf{H}}_1 \end{pmatrix} \mathbf{J}.$$

It follows (10). □

We now establish our main results as follows.

Theorem 1. *Assume that $\nabla^2 g$ is positive definite. Then the proposed complex-valued neural dynamical system in (4) is stable in the Lyapunov sense and its state trajectory is globally convergent to an equilibrium point of (4). Moreover, its output trajectory will converge globally to the optimal solution of (1).*

Proof. First, by Proposition 4 we know that

$$\frac{\partial^2 f}{\partial \mathbf{w} \partial \mathbf{w}^T} = \mathbf{J}^H \frac{\partial^2 g}{\partial \bar{\mathbf{v}} \partial \mathbf{v}^T} \mathbf{J}.$$

Note that J is non-singular. Then that $\nabla^2 g$ is positive definite implies that $\nabla^2 f$ is positive definite too. Next, Proposition 3 shows that the complex-valued neural dynamical system in (4) is equally converted into the following real-valued Lagrange network:

$$\frac{d}{dt}\begin{pmatrix} \mathbf{w}(t) \\ \mathbf{k}(t) \end{pmatrix} = -\lambda \begin{pmatrix} \frac{1}{2}\nabla f(\mathbf{w}(t)) - \mathbf{\Lambda}^T \mathbf{k}(t) \\ \mathbf{\Lambda}\mathbf{w}(t) - \boldsymbol{\beta} \end{pmatrix} \tag{12}$$

where $(\mathbf{w}, \mathbf{k}) \in R^{2n} \times R^{2m}$ is a state vector, $\mathbf{\Lambda} = \varphi_2(\mathbf{A}) \in R^{2m\times 2n}$, $\boldsymbol{\beta} = \varphi_1(\mathbf{b}) \in R^{2m}$, $\mathbf{w} = \varphi_1(\mathbf{z})$, and $\mathbf{k} = \varphi_1(\mathbf{u})$. From the result given in Xia (2003), we know that if $\nabla^2 f(\mathbf{w})$ is positive definite, then the real-valued Lagrange network above is stable in the Lyapunov sense and is globally convergent to its equilibrium point. Let $(\mathbf{w}^*, \mathbf{k}^*)$ be the equilibrium point. Then

$$\begin{cases} \frac{1}{2}\nabla f(\mathbf{w}^*) - \mathbf{\Lambda}^T \mathbf{k}^* = 0 \\ \mathbf{\Lambda}\mathbf{w}^* - \boldsymbol{\beta} = 0. \end{cases}$$

Then

$$\begin{cases} \varphi_1(\nabla g(\mathbf{z}^*)) - \mathbf{\Lambda}^T \varphi_1(\mathbf{u}^*) = 0 \\ \mathbf{\Lambda}\varphi_1(\mathbf{z}^*) - \boldsymbol{\beta} = 0 \end{cases}$$

where $\mathbf{z}^* = \varphi_1^{-1}(\mathbf{w}^*)$ and $\mathbf{u}^* = \varphi_1^{-1}(\mathbf{k}^*)$. It follows that $(\mathbf{z}^*, \mathbf{u}^*)$ satisfies

$$\nabla g(\mathbf{z}^*) - \mathbf{A}^H \mathbf{u}^* = \mathbf{0}, \qquad \mathbf{A}\mathbf{z}^* = \mathbf{b}. \tag{13}$$

On the other side, let $(\mathbf{w}(t), \mathbf{k}(t))$ be the state trajectory of (12) and let $(\mathbf{z}(t), \mathbf{u}(t))$ be the state trajectory of (4). Because $(\mathbf{w}(t), \mathbf{k}(t)) \to (\mathbf{w}^*, \mathbf{k}^*)$ as $t \to \infty$, $(\mathbf{z}(t), \mathbf{u}(t)) = (\varphi_1^{-1}(\mathbf{w}(t)), \varphi_1^{-1}(\mathbf{k}(t))) \to (\mathbf{z}^*, \mathbf{u}^*)$ as $t \to \infty$. Therefore, we can conclude that the proposed complex-valued neural dynamical system in (4) is stable in the Lyapunov sense and its state trajectory is globally convergent to an equilibrium point of (4). Moreover, the output trajectory will converge globally to the optimal solution of (1). □

Theorem 2. *Assume that $\nabla^2 g$ is positive definite at the optimal solution of (1) and is positive semi-definite at others. Then the proposed complex-valued neural dynamical system in (4) is stable in the Lyapunov sense and its state trajectory is globally convergent to an equilibrium point of (4). Moreover, its output trajectory will converge globally to the optimal solution of (1).*

Proof. By Proposition 4 we first know that $\nabla^2 g$ is positive semi-definite at the optimal solution of (1) implies that $\nabla^2 f$ is positive semi-definite, and $\nabla^2 g$ is positive definite at the optimal solution of (1) implies that $\nabla^2 f$ is positive definite at the optimal solution of the following minimization problem:

$$\begin{array}{ll} \min & f(\mathbf{w}) \\ \text{s.t.} & \varphi_2(\mathbf{A})\mathbf{w} = \varphi_1(\mathbf{b}) \end{array} \tag{14}$$

where $\mathbf{w} \in \mathbb{R}^{2n}$, $\mathbf{w} = \varphi_1(\mathbf{z})$, and $f(\mathbf{w}) = g(\mathbf{z})$. Next, Proposition 3 shows that the complex-valued neural dynamical system in (4) is equally converted into the following real-valued Lagrange network in (12). From the result given in Xia et al. (2007), we know that if $\nabla^2 f(\mathbf{w})$ is positive definite at the optimal solution of (14) and is positive semi-definite at others, then the real-valued Lagrange network in (12) is stable in the Lyapunov sense and is globally convergent to its equilibrium point. The rest of proofs is similar to the proof of Theorem 1. □

Following Theorem 2, we have the following further result.

Corollary 1. *Assume that $\nabla^2 g$ is positive definite at the optimal solution of (1) and is positive semi-definite at others. If $\operatorname{rank}(\mathbf{A}) = m$, then the complex-valued neural dynamical system in (4) is globally asymptotically stable.*

Proof. By the definition of φ, the solution set of $\mathbf{Az} = \mathbf{b}$ is equivalent to that of $\mathbf{\Lambda w} = \boldsymbol{\beta}$ in the sense of $\mathbf{z} = \varphi^{-1}(\mathbf{w})$. Since $\text{rank}(\mathbf{A}) = m$, $\text{rank}(\mathbf{\Lambda}) = 2m$. That is, $\mathbf{\Lambda}$ is of full row rank. From the analysis given in Xia (2003), we see that the equilibrium point of (4) is unique. Therefore, from Theorem 2 it follows the conclusion. □

Finally, we consider the following complex-valued quadratic programming problem:

$$\begin{array}{ll} \min & g(\mathbf{z}) = \mathbf{z}^H\mathbf{Q}\mathbf{z} - \mathbf{c}^H\mathbf{z} - \mathbf{z}^H\mathbf{c} \\ \text{s.t.} & \mathbf{Az} = \mathbf{b} \end{array} \tag{15}$$

where $\mathbf{Q} \in \mathbb{C}^{n\times n}$ is positive definite and $\mathbf{c} \in \mathbb{C}^n$. It is easy to see that (15) is a special case of (1). In this case, the dynamical equation in (4) becomes

$$\frac{d}{dt}\begin{pmatrix}\mathbf{z}(t)\\ \mathbf{u}(t)\end{pmatrix} = -\lambda\begin{pmatrix}\mathbf{Q}\mathbf{z}(t) - \mathbf{c} - \mathbf{A}^H\mathbf{u}(t)\\ \mathbf{A}\mathbf{z}(t) - \mathbf{b}\end{pmatrix} \tag{16}$$

where $\lambda > 0$ is a design constant and $\mathbf{z}(t) \in \mathbb{C}^n$ and $\mathbf{u}(t) \in \mathbb{C}^m$ are state vectors.

Note that

$$\mathbf{H}_1 = \frac{\partial^2 g}{\partial\bar{\mathbf{z}}\partial\mathbf{z}^T} = \mathbf{Q}, \qquad \mathbf{H}_2 = \frac{\partial^2 g}{\partial\bar{\mathbf{z}}\partial\mathbf{z}^H} = \mathbf{0}.$$

Then

$$\frac{\partial^2 g}{\partial\bar{\mathbf{v}}\partial\mathbf{v}^T} = \begin{pmatrix}\mathbf{Q} & \mathbf{0}\\ \mathbf{0} & \bar{\mathbf{Q}}\end{pmatrix}$$

which is positive definite. As for the complex-valued dynamical system in (16), from Theorem 2 and Corollary 2 we have the following results:

Corollary 2. *The complex-valued neural dynamical system in (16) is stable in the Lyapunov sense. Moreover, its state trajectory is globally convergent to an equilibrium point of (16) and its output trajectory will converge globally to the optimal solution of (15). If* $\text{rank}(\mathbf{A}) = m$*, the complex-valued neural dynamical system in (16) is globally asymptotically stable.*

Remark 1. Compared with the classic result on the stability of nonlinear systems, we obtain the results of global stability of nonlinear systems, instead of local stability. Thus the obtained results significantly extend the classic stability results of nonlinear systems.

Remark 2. To apply the proposed results to inequality constraints, a possible way is to use the relax variable technique (Cichocki & Unbehauen, 1993) such that inequality constraints are converted into equality constraints.

4. Simulations

In this section, we give two illustrative examples to demonstrate the effectiveness of the obtained results. The simulation is conducted in MATLAB.

Example 1. Consider the complex quadratic programming problem (15) where

$$\mathbf{Q} = \begin{pmatrix}1.4268 & 1.31081 - 0.4296i\\ 1.31081 + 0.4296i & 2.0951\end{pmatrix}$$

$$\mathbf{A} = \begin{pmatrix}1 + i & 0.6 - 0.4i\\ -1 - i & -0.6 + 0.4i\end{pmatrix}$$

$$\mathbf{c} = \begin{pmatrix}1.1127 - 1.0015i\\ 1.8418 - 0.7951i\end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix}0.2 + 0.1i\\ 0.5 - 0.8i\end{pmatrix}.$$

The problem has the following optimal solution:

$(\hat{\mathbf{z}}_1, \hat{\mathbf{z}}_2) = (0.5348 + 0.6611i, 0.5464 - 0.8788i)$.

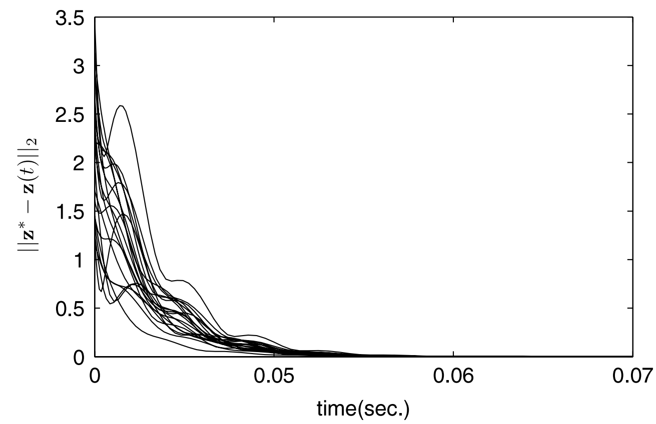

Fig. 1. Transient behavior of $\|\mathbf{z}^* - \mathbf{z}(t)\|_2$ based on the complex-valued dynamical system (16) with 20 random initial points in Example 1.

Table 1
Comparison of the running times (second) of three algorithms in Example 2.

Case	Proposed	RRNN	CVX
1: $n = 100, m = 50$	0.5641	1.2149	0.9507
2: $n = 150, m = 100$	1.6845	4.4453	2.6625
3: $n = 200, m = 150$	4.2022	10.3986	5.9211
4: $n = 250, m = 200$	7.4797	20.0102	11.7986

Because $\mathbf{Q}$ is positive definite and $\text{rank}(\mathbf{A}) = 1$, by Corollary 2 we know that for any given initial point, the complex-valued dynamical system in (16) is stable in the Lyapunov sense and its state trajectory will converge globally to an equilibrium point of (16). Fig. 1 displays transient behavior of $\|\mathbf{z}^* - \mathbf{z}(t)\|_2$ based on the complex-valued dynamical system (16) with 20 random initial points under $\lambda = 120$. It confirms the results of Corollary 2.

Example 2. Consider the complex quadratic programming problem (15) where $\mathbf{Q}$ and $\mathbf{A}$ are random matrices, $\mathbf{b}$ and $\mathbf{c}$ are random vectors, $\mathbf{Q}$ is a positive definite Hermitian matrix, and $\mathbf{A}$ is row-full rank. We study the following cases:

case 1 : $n = 100$, $m = 50$;   case 2 : $n = 150$, $m = 100$

case 3 : $n = 200$, $m = 150$;   case 4 : $n = 250$, $m = 200$.

Because $\mathbf{Q}$ is positive definite and $\text{rank}(\mathbf{A}) = m$, by Corollary 1 we first know that for any given initial point, the complex-valued neural dynamical system in (16) is globally asymptotically stable. For comparative purpose, we now perform the CVX convex optimization method (Grant & Boyd, 2014), the proposed complex-valued dynamical approach, and the real-valued recurrent neural network (RRNN) (Tan et al., 2010). In order to show the numerical stability of the proposed dynamical approach, the computed results were conducted by averaging 50 independent Monte Carlo simulations. Table 1 lists the averaging results of the running times (second) of the three algorithms, where $\lambda = 120$. It is seen that the proposed dynamical approach runs faster than other two algorithms. In order to give further evidence of a statistically significant improvement, let $\lambda = 60$. Fig. 2 displays the statistical outcomes of three box-plots given by the three algorithms for four cases, respectively. It is seen that the distributions of running times of the proposed dynamical approach are more stable and lower than other two algorithms.

Our final example is going to study the complex-valued neural dynamical approach in digital channel equalization applications. Because the digital channel equalization can be well interpreted as a problem of nonlinear classification (Cha & Kassam, 1995), the complex radial basis function network (RBFN) and the complex single-hidden-layer feedforward network (SLFN) named the

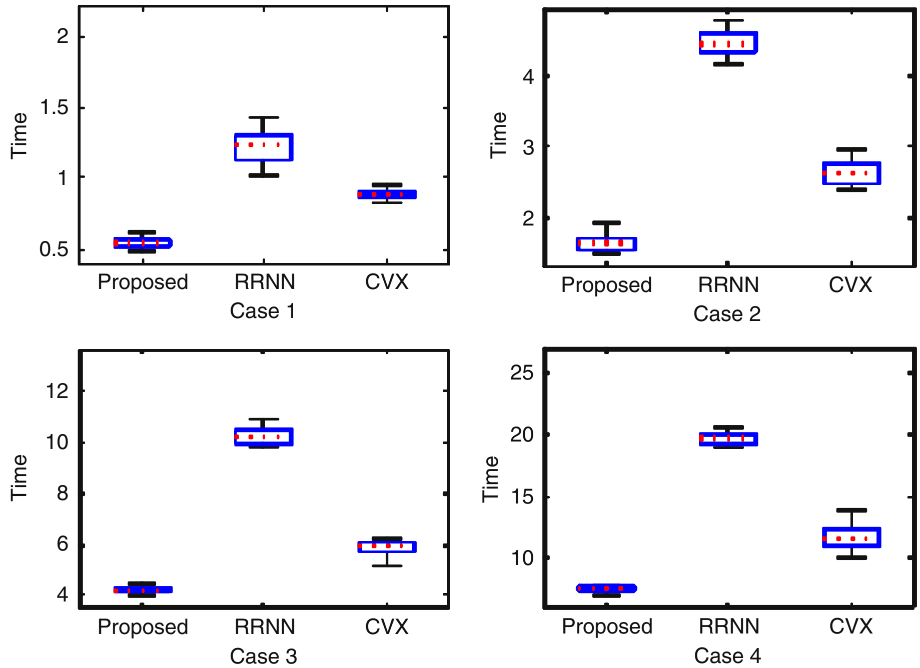

Fig. 2. Box plots of distributions of running times given by three algorithms for four cases.

complex extreme learning machine (C-ELM) were developed for approximating nonlinear mappings (Huang, Zhou, Ding, & Zhang, 2012; Jianping, Sundararajan, & Saratchandran, 2002; Li, Huang, Saratchandran, & Sundararajan, 2005), respectively. By contrast, the C-ELM avoids tuning control parameters and reaches better solutions.

Given a series of complex-valued training samples $(\mathbf{z}_i, \mathbf{y}_i)$, $i = 1, 2, \ldots, N$, where $\mathbf{z}_i \in \mathbb{C}^n$ and $\mathbf{y}_i \in \mathbb{C}^m$, the actual outputs of the single-hidden-layer feedforward network (SLFN) with complex activation function $g_c(z)$ for these N training data is given by Li et al. (2005):

$$\sum_{k=1}^{\tilde{N}} \boldsymbol{\beta}_k g_c(\mathbf{w}_k \cdot \mathbf{z}_i + b_k) = \mathbf{y}_i, \quad i = 1, \ldots, N,$$

where column vector $\mathbf{w}_k \in \mathbb{C}^n$ is the complex input weight vector connecting the input layer neurons to the kth hidden neuron, $\boldsymbol{\beta}_k = [\beta_{k1}, \beta_{k2}, \ldots, \beta_{km}]^T \in \mathbb{C}^m$ the complex output weight vector connecting the kth hidden neuron and the output neurons, and $b_k \in \mathbb{C}$ is the complex bias of the kth hidden neuron. $\mathbf{w}_k \cdot \mathbf{z}_i$ denotes the inner product of column vectors $\mathbf{w}_k$ and $\mathbf{z}_i$. g_c is a fully complex activation function. The above N equations can be written compactly as a complex linear system:

$$\mathbf{A}\boldsymbol{\beta} = \mathbf{Y}$$

where

$$\mathbf{A} = \begin{bmatrix} g_c(\mathbf{w}_1 \cdot \mathbf{z}_1 + b_1) & \cdots & g_c(\mathbf{w}_{\tilde{N}} \cdot \mathbf{z}_1 + b_{\tilde{N}}) \\ \vdots & \cdots & \vdots \\ g_c(\mathbf{w}_1 \cdot \mathbf{z}_N + b_1) & \cdots & g_c(\mathbf{w}_{\tilde{N}} \cdot \mathbf{z}_N + b_{\tilde{N}}) \end{bmatrix}_{N\times\tilde{N}},$$
$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^T \\ \vdots \\ \boldsymbol{\beta}_{\tilde{N}}^T \end{bmatrix}_{\tilde{N}\times m}, \qquad \mathbf{Y} = \begin{bmatrix} \mathbf{y}_1^T \\ \vdots \\ \mathbf{y}_N^T \end{bmatrix}_{N\times m}. \tag{17}$$

Because the complex matrix $\mathbf{A}$ may be rank-deficient matrix, the complex linear system can be solved by a complex generalized inverse solution $\mathbf{A}^{+}\mathbf{Y}$ where $\mathbf{A}^{+}$ is the generalized inverse of complex matrix $\mathbf{A}$. According to the minimality of the Frobenius norm of the pseudo-inverse (Wei & Wu, 2003), the generalized inverse problem can be equivalent to the complex-valued quadratic convex programming problem with equality constraints:

$$\begin{aligned} &\min \quad \|\mathbf{X}\|_F^2 \\ &\text{s.t.} \quad \mathbf{A}^H\mathbf{A}\mathbf{X} = \mathbf{A}^H, \end{aligned} \tag{18}$$

where $\mathbf{A} \in \mathbb{C}^{m\times n}$ is the hidden layer output matrix, $\mathbf{X} \in \mathbb{C}^{n\times m}$ and $\|\mathbf{X}\|_F$ is the Frobenius norm of $\mathbf{X}$. Applying the complex-valued neural dynamical system (4), we can obtain the optimal solution denoted by $\mathbf{X}^*$. Then the generalized inverse solution is given by $\mathbf{X}^*\mathbf{Y}$.

Example 3. Consider a well-known complex nonminimum-phase channel model introduced by Cha and Kassam (1995). This equalization model is of order 3 with nonlinear distortion for 4-QAM signaling. The channel output z_n (which is also the input of the equalizer) is given by

$$\begin{aligned} z_n &= o_n + 0.1o_n^2 + 0.05o_n^3 + v_n, \quad v_n \sim \mathcal{N}(0, 0.05) \\ o_n &= (0.34 - i0.27)s_n + (0.87 + i0.43)s_{n-1} + (0.34 - i0.21)s_{n-2} \end{aligned}$$

where $\mathcal{N}(0, 0.05)$ means the white Gaussian noise (of the nonminimum-phase channel) with mean 0 and variance 0.05. The equalizer input dimension is chosen as 3 and the decision delay τ is set to 1. 4-QAM symbol sequence s_n is passed through the channel and the real and imaginary parts of the symbol are valued from the set $\{\pm 0.7\}$. The hidden neuron numbers of the equalizer C-ELM is set to 4. The fully complex activation function is chosen as $\operatorname{arcsin} h(z) = \int_0^z dt/[(1+t^2)^{1/2}]$, where $z = \mathbf{w} \cdot \mathbf{z} + b$. Both the input weight vectors $\mathbf{w}_k$ and biases b_k are randomly chosen from a complex area centered at the origin with the radius set as 0.1. Fig. 3 shows the 4-QAM test set channel output distribution. We perform the complex-valued neural dynamical system (4). Fig. 4 shows the eye diagram of the output of the C-ELM equalizer obtained by the complex-valued neural dynamical system (4). From Fig. 4 we see that the obtained C-ELM can separate the output into four regions clearly.

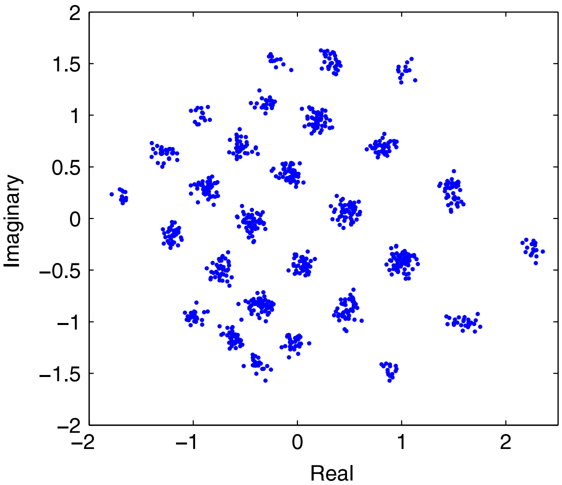

Fig. 3. The distribution of the 4-QAM test set channel output.

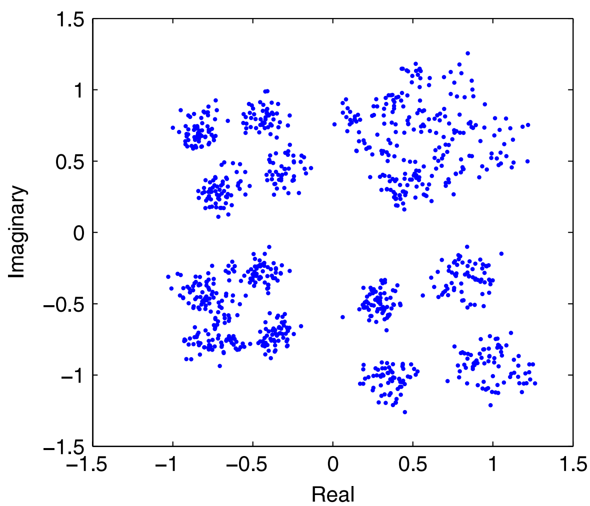

Fig. 4. Eye diagram of the output of the C-ELM equalizer.

5. Conclusion

In this paper, we have developed a complex-valued neural dynamical method for solving complex-valued nonlinear convex programming problems. The proposed neural dynamical approach significantly generalizes the real-valued nonlinear Lagrange network completely in the complex domain. Compared with existing real-valued neural networks and numerical optimization methods for solving complex-valued quadratic convex programming problems, the proposed complex-valued neural dynamical approach has a low model complexity and storage capacity. Theoretically, the proposed complex-valued dynamical approach is shown to be globally stable and convergent to the optimal solution. Compared with the classic result on the stability of nonlinear systems, our stability results is global, instead of local. Compared with stability results of the real-valued neural dynamical approach, our results establish in the complex domain, instead of the real domain. Two illustrative examples have demonstrated the effectiveness of the obtained results. Further study will focus on the complex-valued dynamical approaches to more general cases involving inequality constraints and their applications.

Acknowledgments

The authors thank the associate editor and reviewers for their encouragement and valuable comments, which helped in improving the quality of the paper.

Appendix

A.1. Properties of φ_1 and φ_2

In this paper, the linear invertible mappings φ_1 and φ_2 play a key role in our discussion.

Proposition A.1. *$\forall \mathbf{z}_1, \mathbf{z}_2 \in \mathbb{C}^n, \sigma \in \mathbb{R}, \mathbf{A} \in \mathbb{C}^{m\times n}, \mathbf{B} \in \mathbb{C}^{n\times l}, \mathbf{D} \in \mathbb{C}^{n\times n}$ and $\mathbf{D}$ is invertible, then:*

(1) $\varphi_1(\mathbf{z}_1 \pm \mathbf{z}_2) = \varphi_1(\mathbf{z}_1) \pm \varphi_1(\mathbf{z}_2)$; $\quad \varphi_1(\sigma\mathbf{z}) = \sigma\varphi_1(\mathbf{z})$
(2) $\varphi_2(\mathbf{A} \pm \mathbf{B}) = \varphi_2(\mathbf{A}) \pm \varphi_2(\mathbf{B})$; $\quad \varphi_2(\sigma\mathbf{A}) = \sigma\varphi_2(\mathbf{A})$; $\varphi_2(\mathbf{I}_n) = \mathbf{I}_{2n}$
(3) $\varphi_2(\mathbf{A}\cdot\mathbf{B}) = \varphi_2(\mathbf{A})\cdot\varphi_2(\mathbf{B})$; $\quad \varphi_1(\mathbf{Az}) = \varphi_2(\mathbf{A})\varphi_1(\mathbf{z})$;
(4) $\varphi_2(\mathbf{A}^H) = (\varphi_2(\mathbf{A}))^T$; $\quad \varphi_2(\mathbf{D}^{-1}) = (\varphi_2(\mathbf{D}))^{-1}$;
(5) $\|\mathbf{z}\|_2^2 = \|\varphi_1(\mathbf{z})\|_2^2$; $\quad \mathbf{Re}(\mathbf{z}_1^H\mathbf{z}_2) = \varphi_1(\mathbf{z}_1)^T\varphi_1(\mathbf{z}_2)$.

Proof. The proofs of (1) and (2) are straightforward consequence of the definition of φ. Now we prove the rest of conclusions.

(3) Since

$$\begin{aligned}\mathbf{A}\cdot\mathbf{B} = &(\mathbf{Re}(\mathbf{A})\cdot\mathbf{Re}(\mathbf{B}) - \mathbf{Im}(\mathbf{A})\cdot\mathbf{Im}(\mathbf{B})) \\ &+ i(\mathbf{Im}(\mathbf{A})\cdot\mathbf{Re}(\mathbf{B}) + \mathbf{Re}(\mathbf{A})\cdot\mathbf{Im}(\mathbf{B})),\end{aligned}$$

by the definition of φ_2, then

$$\begin{aligned}\varphi_2(\mathbf{A}\cdot\mathbf{B}) &= \begin{pmatrix}\mathbf{Re}(\mathbf{A})\cdot\mathbf{Re}(\mathbf{B}) - \mathbf{Im}(\mathbf{A})\cdot\mathbf{Im}(\mathbf{B}) & -(\mathbf{Im}(\mathbf{A})\cdot\mathbf{Re}(\mathbf{B}) + \mathbf{Re}(\mathbf{A})\cdot\mathbf{Im}(\mathbf{B})) \\ \mathbf{Im}(\mathbf{A})\cdot\mathbf{Re}(\mathbf{B}) + \mathbf{Re}(\mathbf{A})\cdot\mathbf{Im}(\mathbf{B}) & \mathbf{Re}(\mathbf{A})\cdot\mathbf{Re}(\mathbf{B}) - \mathbf{Im}(\mathbf{A})\cdot\mathbf{Im}(\mathbf{B})\end{pmatrix} \\ &= \varphi_2(\mathbf{A})\cdot\varphi_2(\mathbf{B}).\end{aligned}$$

Since

$$\begin{aligned}\mathbf{Az} &= (\mathbf{Re}(\mathbf{A}) + i\mathbf{Im}(\mathbf{A}))(\mathbf{Re}(\mathbf{z}) + i\mathbf{Im}(\mathbf{z})) \\ &= (\mathbf{Re}(\mathbf{A})\mathbf{Re}(\mathbf{z}) - \mathbf{Im}(\mathbf{A})\mathbf{Im}(\mathbf{z})) \\ &\quad + i(\mathbf{Re}(\mathbf{A})\mathbf{Im}(\mathbf{z}) + \mathbf{Im}(\mathbf{A})\mathbf{Re}(\mathbf{z})),\end{aligned}$$

then

$$\begin{aligned}\varphi_1(\mathbf{Az}) &= \begin{pmatrix}\mathbf{Re}(\mathbf{A})\mathbf{Re}(\mathbf{z}) - \mathbf{Im}(\mathbf{A})\mathbf{Im}(\mathbf{z}) \\ \mathbf{Re}(\mathbf{A})\mathbf{Im}(\mathbf{z}) + \mathbf{Im}(\mathbf{A})\mathbf{Re}(\mathbf{z})\end{pmatrix} \\ &= \begin{pmatrix}\mathbf{Re}(\mathbf{A}) & -\mathbf{Im}(\mathbf{A}) \\ \mathbf{Im}(\mathbf{A}) & \mathbf{Re}(\mathbf{A})\end{pmatrix}\begin{pmatrix}\mathbf{Re}(\mathbf{z}) \\ \mathbf{Im}(\mathbf{z})\end{pmatrix} \\ &= \varphi_2(\mathbf{A})\varphi_1(\mathbf{z}).\end{aligned}$$

(4) Since

$$\mathbf{A}^H = \mathbf{Re}(\mathbf{A}^T) - i\mathbf{Im}(\mathbf{A}^T),$$

then

$$\begin{aligned}\varphi_2(\mathbf{A}^H) &= \begin{pmatrix}\mathbf{Re}(\mathbf{A}^T) & \mathbf{Im}(\mathbf{A}^T) \\ -\mathbf{Im}(\mathbf{A}^T) & \mathbf{Re}(\mathbf{A}^T)\end{pmatrix} \\ &= (\varphi_2(\mathbf{A}))^T.\end{aligned}$$

Next, we prove $\varphi_2(\mathbf{D}^{-1}) = (\varphi_2(\mathbf{D}))^{-1}$. Since

$$\begin{aligned}\mathbf{D}\cdot\mathbf{D}^{-1} &= (\mathbf{Re}(\mathbf{D}) + i\mathbf{Im}(\mathbf{D}))(\mathbf{Re}(\mathbf{D}^{-1}) + i\mathbf{Im}(\mathbf{D}^{-1})) \\ &= (\mathbf{Re}(\mathbf{D})\mathbf{Re}(\mathbf{D}^{-1}) - \mathbf{Im}(\mathbf{D})\mathbf{Im}(\mathbf{D}^{-1})) \\ &\quad + i(\mathbf{Re}(\mathbf{D})\mathbf{Im}(\mathbf{D}^{-1}) + \mathbf{Im}(\mathbf{D})\mathbf{Re}(\mathbf{D}^{-1})) = \mathbf{I}_n,\end{aligned}$$

it follows that

$$\begin{aligned}&\mathbf{Re}(\mathbf{D})\mathbf{Re}(\mathbf{D}^{-1}) - \mathbf{Im}(\mathbf{D})\mathbf{Im}(\mathbf{D}^{-1}) = \mathbf{I}_n, \\ &\mathbf{Re}(\mathbf{D})\mathbf{Im}(\mathbf{D}^{-1}) + \mathbf{Im}(\mathbf{D})\mathbf{Re}(\mathbf{D}^{-1}) = \mathbf{0}.\end{aligned}$$

Then further derive

$$\varphi_2(\mathbf{D}^{-1}) \cdot \varphi_2(\mathbf{D}) = \begin{pmatrix} \mathbf{Re}(\mathbf{D}^{-1}) & -\mathbf{Im}(\mathbf{D}^{-1}) \\ \mathbf{Im}(\mathbf{D}^{-1}) & \mathbf{Re}(\mathbf{D}^{-1}) \end{pmatrix} \cdot \begin{pmatrix} \mathbf{Re}(\mathbf{D}) & -\mathbf{Im}(\mathbf{D}) \\ \mathbf{Im}(\mathbf{D}) & \mathbf{Re}(\mathbf{D}) \end{pmatrix}$$
$$= \begin{pmatrix} \mathbf{Re}(\mathbf{D}^{-1})\mathbf{Re}(\mathbf{D}) - \mathbf{Im}(\mathbf{D})\mathbf{Im}(\mathbf{D}^{-1}) & -\mathbf{Re}(\mathbf{D}^{-1})\mathbf{Im}(\mathbf{D}) - \mathbf{Re}(\mathbf{D})\mathbf{Im}(\mathbf{D}^{-1}) \\ \mathbf{Re}(\mathbf{D}^{-1})\mathbf{Im}(\mathbf{D}) + \mathbf{Re}(\mathbf{D})\mathbf{Im}(\mathbf{D}^{-1}) & \mathbf{Re}(\mathbf{D}^{-1})\mathbf{Re}(\mathbf{D}) - \mathbf{Im}(\mathbf{D})\mathbf{Im}(\mathbf{D}^{-1}) \end{pmatrix}$$
$$= \begin{pmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_n \end{pmatrix} = \mathbf{I}_{2n}.$$

Because of the uniqueness of $\mathbf{D}^{-1}$, $\varphi_2(\mathbf{D}^{-1}) = (\varphi(\mathbf{D})_2)^{-1}$ holds.

(5)

$$\begin{aligned} \|\mathbf{z}\|_2^2 &= \mathbf{z}^H\mathbf{z} \\ &= (\mathbf{Re}(\mathbf{z})^T\mathbf{Re}(\mathbf{z}) + \mathbf{Im}(\mathbf{z})^T\mathbf{Im}(\mathbf{z})) \\ &= \|\varphi(\mathbf{z})\|_2^2. \end{aligned}$$

The first conclusion holds.

Since

$$\begin{aligned} \mathbf{z}_1^H\mathbf{z}_2 &= (\mathbf{Re}(\mathbf{z}_1)^T - i\mathbf{Im}(\mathbf{z}_1)^T) \cdot (\mathbf{Re}(\mathbf{z}_2) + i\mathbf{Im}(\mathbf{z}_2)) \\ &= (\mathbf{Re}(\mathbf{z}_1)^T\mathbf{Re}(\mathbf{z}_2) + \mathbf{Im}(\mathbf{z}_1)^T\mathbf{Im}(\mathbf{z}_2)) \\ &\quad + i(\mathbf{Re}(\mathbf{z}_1)^T\mathbf{Im}(\mathbf{z}_2) - \mathbf{Im}(\mathbf{z}_1)^T\mathbf{Re}(\mathbf{z}_2)) \end{aligned}$$

then,

$$\begin{aligned} \mathbf{Re}(\mathbf{z}_1^H\mathbf{z}_2) &= \begin{pmatrix} \mathbf{Re}(\mathbf{z}_1)^T & \mathbf{Im}(\mathbf{z}_1)^T \end{pmatrix} \begin{pmatrix} \mathbf{Re}(\mathbf{z}_2) \\ \mathbf{Im}(\mathbf{z}_2) \end{pmatrix} \\ &= \varphi_1(\mathbf{z}_1)^T\varphi_2(\mathbf{z}_2). \end{aligned}$$

The second conclusion holds too. □

References

Aizenberg, I. N. (2008). Solving the XOR and parity N problems using a single universal binary neuron. *Soft Computing*, *12*(3), 215–222.

Aizenberg, I., Dmitriy, P. V., & Zurada, J. M. (2007). Blur identification by multilayer neural network based on multivalued neurons. *IEEE Transactions on Neural Networks*, *19*, 883–898.

Boyd, S., & Vandenberghe, L. (2006). *Convex optimization*. Cambridge: Cambridge University Press.

Brandwood, D. H. (1983). A complex gradient operator and its application in adaptive array theory. *Proceedings of the Institution of Electrical Engineers*, *130*, 11–16.

Cha, I., & Kassam, S. A. (1995). Channel equalization using adaptive complex radial basis function networks. *IEEE Journal on Selected Areas in Communications*, *13*, 122–131.

Chong, E. K. P., Hui, S., & Zak, S. H. (1999). An analysis of a class of neural networks for solving linear programming problems. *IEEE Transactions on Automatic Control*, *44*(11), 1995–2006.

Cichocki, A., & Unbehauen, R. (1993). *Neural networks for optimization and signal processing*. Chichester: Wiley.

Forti, M., Nistri, P., & Quincampoix, M. (2004). Generalized neural network, for nonsmooth nonlinear programming problems. *IEEE Transactions on Circuits and Systems. I. Regular Papers*, *51*, 1741–1754.

Forti, M., Nistri, P., & Quincampoix, M. (2006). Convergence of neural networks for programming problems via a nonsmooth Lojasiewicz inequality. *IEEE Transactions on Neural Networks*, *17*, 1471–1486.

Goh, S. L., Chen, M., Popovic, D. H., Aihara, K., Obradovic, D., & Mandic, D. P. (2006). Complex-valued forecasting of wind profile. *Renewable Energy*, *31*(11), 1733–1750.

Goh, S. L., & Mandic, D. P. (2004). A general complex RTRL algorithm for nonlinear adaptive filters. *Neural Computation*, *16*, 2699–2713.

Goh, S. L., & Mandic, D. P. (2007). An augmented extended Kalman filter algorithm for complex-valued recurrent neural networks. *Neural Computation*, *19*(4), 1–17.

Grant, M., & Boyd, S. (2014). CVX:MATLAB software for disciplined convex programming (Web page and software). http://cvxr.com/cvx/.

Hanna, A. I., & Mandic, D. P. (2003). A general fully adaptive normalised gradient descent learning algorithm for complex-valued nonlinear adaptive filters. *IEEE Transactions on Signal Processing*, *51*, 2540–2549.

Hirose, A. (2004). *Complex-valued neural networks: theories and applications*. World Scientific.

Hirose, A., & Yoshida, S. (2012). Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence. *IEEE Transactions on Neural Networks and Learning Systems*, *23*, 541–551.

Hjorungnes, A., & Gesbert, D. (2007). Complex-valued matrix differentiation: techniques and key results. *IEEE Transactions on Signal Processing*, *55*(6), 2740–2746.

Hu, J., & Wang, J. (2012). Global stability of complex-valued recurrent neural networks with time-delays. *IEEE Transactions on Neural Networks and Learning Systems*, *23*, 853–865.

Hualiang, L., & Adali, T. (2006). Optimization in the complex domain for nonlinear adaptive filtering. In *Signals, systems and computers, 2006. ACSSC'06 Fortieth Asilomar conference on IEEE* (pp. 263–267).

Huang, G. B., Zhou, H., Ding, X., & Zhang, R. (2012). Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *42*(2), 513–529.

Javidi, S., Mandic, D. P., & Kuh, A. (2010). Optimisation of real functions of complex matrices for the adaptive estimation of complex sources. In *Green circuits and systems (ICGCS), 2010 international conference on* (pp. 30–35). IEEE.

Jianping, D., Sundararajan, N., & Saratchandran, P. (2002). Communication channel equalization using complex-valued minimal radial basis function neural networks. *IEEE Transactions on Neural Networks*, *13*, 687–696.

Kreutz-Delgado, K. (2006). The complex gradient operator and the CR calculus. In *Lecture supplement ECE275A* (pp. 1–74).

Li, M. B., Huang, G. B., Saratchandran, P., & Sundararajan, N. (2005). Fully complex extreme learning machine. *Neurocomputing*, *68*, 306–314.

Loesch, B., & Yang, B. (2013). Cramer–Rao bound for circular and noncircular complex independent component analysis. *IEEE Transactions on Signal Processing*, *61*, 365–379.

Lu, W., & Wang, J. (2008). Convergence analysis of a class of nonsmooth gradient systems. *IEEE Transactions on Circuits and Systems. I. Regular Papers*, *55*(11), 3514–3527.

Lustig, M., Donoho, D., & Pauly, J. M. (2007). Sparse MRI: the application of compressed sensing for rapid MR imaging. *Magnetic Resonance in Medicine*, *6*, 1182–1195.

Mandic, D. P., & Goh, S. L. (2009). *Complex valued nonlinear adaptive filters: noncircularity, widely linear and neural models*. New York: Wiley.

Mandic, D. P., Souretis, G., Javidi, S., Goh, S. L., & Tanaka, T. (2007). Why a complex valued solution for a real domain problem. In *Proceedings of the international conference on machine learning for signal processing, MLSP'07* (pp. 384–389).

Mandic, D. P., Still, S., & Douglas, S. C. (2009). Duality between widely linear and dual channel adaptive filtering. In *Acoustics, speech and signal processing, 2009. IEEE international conference on*, ICASSP 2009. (pp. 1729–1732). IEEE.

Nitta, T. (2009). *Complex-valued neural networks: utilizing high-dimensional parameters*. IGI Global.

Oppenheim, A. V., & Lim, J. S. (1981). The importance of phase in signals. *Proceedings of the IEEE*, *69*(5), 529–541.

Perfetti, R., & Todisco, M. (2008). Quasi-Lagrangian neural network for convex quadratic optimization. *IEEE Transactions on Neural Networks*, *19*, 1804–1809.

Remmert, R. (1991). *Theory of complex functions*. New York: Springer-Verlag.

Sorber, L., Barel, M. V., & Lathauwer, L. D. (2012). Unconstrained optimization of real functions in complex variables. *SIAM Journal on Optimization*, *22*(3), 879–898.

Squire, W., & Trapp, G. (1998). Using complex variables to estimate derivatives of real functions. *SIAM Review*, *40*(1), 110–112.

Tan, Y., Shi, J., & Tan, K. C. (2010). A recurrent neural network for solving complex-valued quadratic programming problems with equality constraints. In *LNCS: Vol. 6146. ICSI 2010, Part II* (pp. 321–326).

Vakhania, N. N., & Kandelaki, N. P. (1996). Random vectors with values in complex Hilbert spaces. *Theory of Probability and Its Applications*, *41*(1), 116–131.

Van den Bos, A. (1994). Complex gradient and Hessian. *IEE Proceedings. Vision, Image & Signal Processing*, *141*, 380–382.

Wang, J., Hu, Q., & Jiang, D. (1999). A Lagrangian network for kinematic control of redundant robot manipulators. *IEEE Transactions on Neural Networks*, *10*(5), 1123–1132.

Wei, Y., & Wu, H. (2003). The representation and approximation for the generalized inverse $\mathbf{A}_{T,S}^{(2)}$. *Applied Mathematics and Computation*, 263–276.

Xia, Y. S. (2003). Global convergence analysis of Lagrangian networks. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, *50*, 818–822.

Xia, Y. S., & Feng, G. (2006). A neural network for robust LCMP beamforming. *Signal Processing*, *35*, 54–64.

Xia, Y. S., Feng, G., & Kamel, M. S. (2007). Development and analysis of neural dynamical approaches to solving nonlinear programming problems. *IEEE Transactions on Automatic Control*, *52*, 2154–2159.

Zeng, Z., & Wang, J. (2008). Analysis and design of high-capacity associative memories based on class of discrete-time recurrent neural networks. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *38*(6), 1525–1536.

Zhang, S., & Constantinides, A. G. (1992). Lagrange programming neural networks. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, *39*(7), 441–452.

Zhang, Y. W., & Ma, Y. L. (1997). CGHA for principal component extraction in the complex domain. *IEEE Transactions on Neural Networks*, *8*, 1031–1036.