mathematics

MDPI

Article

State Estimation for Complex-Valued Inertial Neural Networks with Multiple Time Delays

Yaning Yu and Ziye Zhang *

College of Mathematics and Systems Science, Shandong University of Science and Technology,
Qingdao 266590, China; yyn19960227@163.com
* Correspondence: zhangziye@sdust.edu.cn

Abstract: In this paper, the problem of state estimation for complex-valued inertial neural networks with leakage, additive and distributed delays is considered. By means of the Lyapunov–Krasovskii functional method, the Jensen inequality, and the reciprocally convex approach, a delay-dependent criterion based on linear matrix inequalities (LMIs) is derived. At the same time, the network state is estimated by observing the output measurements to ensure the global asymptotic stability of the error system. Finally, two examples are given to verify the effectiveness of the proposed method.

Keywords: complex-valued inertial neural networks; state estimation; multiple time delays

MSC: 93D05

check for updates

Citation: Yu, Y.; Zhang, Z. State Estimation for Complex-Valued Inertial Neural Networks with Multiple Time Delays. *Mathematics* **2022**, *10*, 1725. https://doi.org/10.3390/math10101725

Academic Editors: Changhua Lien, Hamid Reza Karimi and Sundarapandian Vaidyanathan

Received: 18 March 2022
Accepted: 16 May 2022
Published: 18 May 2022

1. Introduction

In the past decades, people have performed much research on various types of neural network models. Because of the complex dynamic characteristics of the neurocyte, it is necessary to consider describing the complex dynamic properties of this neural response by using derivative information of the state variables. This makes neural networks have good application prospects in various fields [1,2]. Therefore, the dynamical behaviors of such systems have received considerable attention [3–6]. In addition, inertial neural networks are described by second-order differential equations, which are different from first-order differential models such as bidirectional associative memory neural networks and Cohen–Grossberg neural networks. The inertial neural network model is widely used in biology and engineering due to its more complex dynamics phenomena and superior characteristics, as in [7,8]. As a result, researchers have begun to focus on this system, and many instructive results have emerged [9–11]. For example, ref. [10] analyzed the global exponential stability problem of fuzzy inertial neural networks with time-varying delays, and [11] discussed the global exponential stabilization and lag synchronization control problems of inertial neural networks with time delays.

However, it is well known that in the design of neural network state feedback controllers, it is of great significance to obtain the state information of the networks. Unfortunately, in practice, due to the strong coupling of large-scale neural networks, it is difficult to obtain accurate and complete neural state information from the networks' output. Therefore, we need to adopt reasonable measurement methods and design various effective estimators to estimate the state of neurons so as to further describe and simulate the complex responses of neurons. Based on this, people have performed much research on the state estimation of neural networks [12–18]. It is worth emphasizing that in [12], the state estimation problem of neural networks was discussed for the first time. Subsequently, some important relevant results appeared, one after another [13–18]. However, there are few research works on state estimation of inertial neural networks, only appearing in [17,18]. As a result, it is worthwhile to explore the state estimation problem of inertial neural networks.

On the other hand, neural networks with real-valued weight, output, state, and activation functions are called real-valued neural networks. It is undeniable that in many application fields, real-valued neural networks have their excellent side, but also have certain limitations in some aspects. In order to surmount these limitations, researchers naturally put forward complex-valued neural networks, and more and more studies [19–32] on the dynamical behaviors of complex-valued neural networks have been carried out in recent years. At the same time, many scholars also pay great attention to the analysis of complex-valued inertial neural networks [33–38]. From the perspective of research methods, most of them transform the second-order differential systems into the first-order ones through variable transformation and then study the addressed systems based on the first-order ones [34,35]. However, this method can double the state variables and dimension of the systems. In order to maintain the original characteristics of the systems and make the theoretical analysis simple, via the method of non-reduced order, refs. [36–39] analyzed the exponential and adaptive synchronization and the finite/fixed time synchronization for complex-valued inertial neural networks, respectively.

In reality, the change trend of many systems is not only related to the current state, but also depends on the past state, which naturally involves time delays. As is known to all, as one of the main sources of system instability, oscillation, or performance deterioration, time delays inevitably affect many systems in different forms, including leakage time delay, additive time delays, distributed time delay, and so on. For bidirectional associative memory neural networks, the finite time stability problem with distributed delays [40] and the state estimation problem with additive delays [41] are considered, respectively. In [42–44], the state estimation problem of quaternion-valued neural networks and event-triggered exponential stabilization for inertial complex-valued neural networks with multiple time delays were studied. At present, based on the influence of time delays and the characteristics of inertial neural networks, some scholars have combined the inertia term, time delays, and the most classical first-order differential models to explore the synchronization and anti-synchronization of complex-valued inertial neural networks with time-varying delays, such as [45]. Unfortunately, for complex-valued inertial neural networks, there is no literature on the existence of the above three time delays at the same time, not to mention the research on state estimation, so we need to fill in the gaps in this aspect.

Inspired by the above reasons, the state estimation problem of a class of complex-valued inertial neural networks with multiple time delays is studied by using a non-reduced-order method. The main challenges and contributions of this paper are summarized below: (1) Different from [27,28,31], separating the complex-valued neural networks into two equivalent real-valued subsystems, the state estimation problem of complex-valued inertial neural networks with multiple time delays is studied via the nonseparable method. (2) To maintain the the characteristics of the original system and increase the flexibility and generality of the theoretical results, the considered second-order differential model is regarded as a whole instead of reducing the order based on variable substitution. (3) To achieve our aim, an appropriate Lyapunov–Krasovskii functional that fully considers the information of leakage delay, additive delays, and distributed delay is constructed, in which the inverted convex inequality in the complex domain is adopted.

2. Preliminaries and Model Descriptions

Consider the following complex-valued inertial neural networks with leakage and additive and distributed delays:

$$\ddot{u}(t) = -A\dot{u}(t) - Bu(t-\delta) + Cf(u(t)) + Df(u(t-\tau_1(t)-\tau_2(t))) + E\int_{t-\beta}^{t} f(u(s))ds \quad (1)$$

where $u(t) = (u_1(t), u_2(t), \ldots, u_n(t))^T \in C^n$ is the state of the neuron n, and its second derivative is called the term of inertia; $A = \text{diag}(a_1, a_2, \ldots, a_n) \in R^{n\times n}$ with $a_p > 0(p = 1,2,\ldots,n)$ and $B = \text{diag}(b_1, b_2, \ldots, b_n) \in R^{n\times n}$ with $b_p > 0$ indicate the self-feedback connection weight matrices; $C \in C^{n\times n}$, $D \in C^{n\times n}$ and $E \in C^{n\times n}$ are connection weight matrices; $f(u(t)) = (f_1(u_1(t)), f_2(u_2(t)), \ldots, f_n(u_n(t)))^T \in C^n$ represents the complex-valued activation functions; δ is the leakage delay satisfying $\delta \geq 0$; $\tau_q(t)(q = 1,2)$ denotes the time-varying delays with $0 \leq \tau_q(t) \leq \tau_q$, and β stands for the distributed delay, where τ_q are given real constants.

Let $\tau(t) = \tau_1(t) + \tau_2(t)$ and $\tau = \tau_1 + \tau_2$. The initial conditions of System (1) are

$$u(s) = \eta(s), \dot{u}(s) = \rho(s), -h \leq s \leq 0 \tag{2}$$

where $\eta(s)$ and $\rho(s)$ are continuous and $h = \max\{\delta, \tau, \beta\}$.

Moreover, the measurement outputs of System (1) are assumed as

$$m(t) = Fu(t) + Gg(u(t)) \tag{3}$$

where $m(t) \in C^m$ is the measurement output of System (1); $F \in C^{m\times n}$ and $G \in C^{m\times m}$ are the output weighting matrices; $g(u(t)) \in C^m$ denotes neuron-dependent nonlinear disturbance signals.

Further, Equations (1)–(3) can be integrated as

$$\begin{cases} \ddot{u}(t) = -A\dot{u}(t) - Bu(t-\delta) + Cf(u(t)) + Df(u(t-\tau_1(t)-\tau_2(t))) + E\int_{t-\beta}^{t} f(u(s))ds \\ m(t) = Fu(t) + Gg(u(t)) \\ u(s) = \eta(s), \dot{u}(s) = \rho(s), s \in [-h, 0]. \end{cases} \tag{4}$$

Next, we construct a full-order state estimator for System (4):

$$\begin{cases} \ddot{v}(t) = -A\dot{v}(t) - Bv(t-\delta) + Cf(v(t)) + Df(v(t-\tau_1(t)-\tau_2(t))) + E\int_{t-\beta}^{t} f(v(s))ds \\ \qquad + K(\dot{m}(t) - \dot{n}(t)) + H(m(t) - n(t)) \\ n(t) = Fv(t) + Gg(v(t)) \\ v(s) = \tilde{\eta}(s), \dot{v}(s) = \tilde{\rho}(s), s \in [-h, 0] \end{cases} \tag{5}$$

where $v(t)$ is the estimation of $u(t)$, $n(t)$ is the estimated output, and $K, H \in C^{n\times m}$ are the state estimator gains matrices to be designed.

Now, define $e(t) = u(t) - v(t)$, $\tilde{f}(e(t)) = f(u(t)) - f(v(t))$ and $\tilde{g}(e(t)) = g(u(t)) - g(v(t))$; the estimation error system can be derived from (4) and (5) as follows:

$$\begin{cases} \ddot{e}(t) = -(A+KF)\dot{e}(t) - HFe(t) - Be(t-\delta) + C\tilde{f}(e(t)) + D\tilde{f}(e(t-\tau_1(t)-\tau_2(t))) \\ \qquad + E\int_{t-\beta}^{t} \tilde{f}(e(s))ds - KG\tilde{g}(\dot{e}(t)) - HG\tilde{g}(e(t)) \\ e(s) = \varrho(s), \dot{e}(s) = \varsigma(s), s \in [-h, 0]. \end{cases} \tag{6}$$

The flow chart of complex-valued inertial networks networks and the design estimator are illustrated in Figure 1.

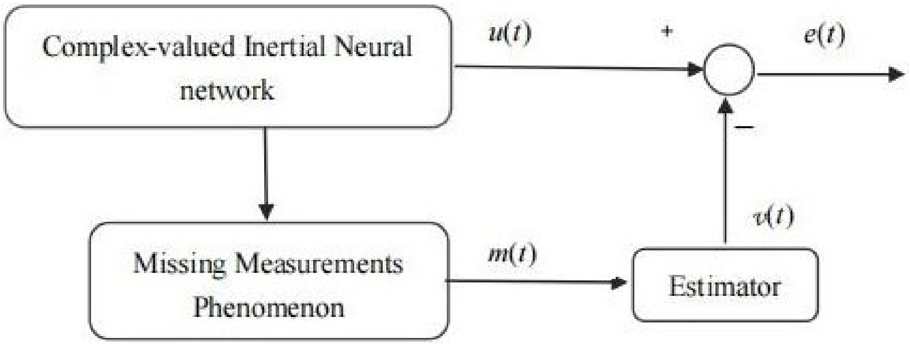
Complex-valued Inertial Neural network
u(t)
+
e(t)
−
v(t)
m(t)
Missing Measurements Phenomenon
Estimator

Figure 1. Flow chart of the estimator.

Remark 1. *In the real control networks, when the signal is transmitted from one point to the next, it will pass through physical equipment, controllers, sensors, and actuators. In this process, several different types of time delays may occur due to the changes of networks; transmission conditions, such as leakage time delay, additive time delays, and distributed time delay. On the other hand, in large-scale neural networks, there are only partial neuron states' information in the network output. In order to make better use of neural networks, it is usually necessary to estimate the state of neurons* *through output measurements and then use the estimated neuron state to complete some actual performance of the system. Hence, it is very important to discuss the state estimation problem for neural network models with multiple time delays. Here, we focus on this problem for complex-valued inertial neural networks with multiple time delays.*

The following preliminaries will be used in the derivation of the next section.

Assumption 1. *For any $\kappa \in \{1, 2, \ldots, n\}$, there exists a constant $l_\kappa \in R$ such that*

$$|f_\kappa(x) - f_i(y)| \leq l_\kappa |x - y| \tag{7}$$

for all $x, y \in C$.

Assumption 2. *For any $\kappa \in \{1, 2, \ldots, m\}$, there exists a constant $N_\kappa \in R^n$ such that*

$$|g_\kappa(u) - g_\kappa(v)| \leq |N_\kappa^T(u - v)| \tag{8}$$

for all $u, v \in C^n$.

For convenience, we define $L = \text{diag}(l_1, l_2, \ldots, l_n)$ and $N = (N_1, N_2, \ldots, N_m)$.

Lemma 1 ([21]). *For positive definite Hermitian matrix $P \in C^{n\times n}$, vector function $u(s) : [c, d] \to C^n$ with scalars $c < d$, then*

$$\left(\int_c^d u(s)ds\right)^* P\left(\int_c^d u(s)ds\right) \leq (d - c)\int_c^d u^*(s)Pu(s)ds. \tag{9}$$

Lemma 2 ([45]). *For any given vectors $\xi_1, \xi_2 \in C^n$, any scalar $0 < \rho < 1$, any positive definite Hermitian matrix $P \in C^{n\times n}$, and any matrix $Q \in C^{n\times n}$ such that $\begin{bmatrix} P & Q \\ Q^* & P \end{bmatrix} > 0$,*

The following inequality holds:

$$\frac{1}{\rho}\xi_1^* P\xi_1 + \frac{1}{1-\rho}\xi_2^* P\xi_2 \geq \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}^* \begin{bmatrix} P & Q \\ Q^* & P \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}. \tag{10}$$

3. Main Results

In this section, the state estimation problem of complex-valued inertial neural networks is analyzed, and a sufficient condition for the global asymptotic stability of the error state system (6) is proposed.

Theorem 1. *Suppose Assumptions 1 and 2 hold; the error system (6) is globally asymptotically stable if there exist real positive diagonal matrices $\Lambda_1, \Lambda_2 \in R^{n\times n}$, $\Lambda_3, \Lambda_4 \in R^{m\times m}$, positive definite Hermitian matrices $P_1, P_4, P_6, Q_1, Q_2, S_1, S_3, U_1, U_2, W_1, W_2, R \in C^{n\times n}$, and any matrices $P_2, P_3, P_5, S_2, \tilde{W}_1, \tilde{W}_2, \in C^{n\times n}$, $\bar{K}, \bar{H} \in C^{n\times m}$, such that the following LMIs hold:*

$$P = \begin{bmatrix} P_1 & P_2 & P_3 \\ * & P_4 & P_5 \\ * & * & P_6 \end{bmatrix} > 0 \tag{11}$$

$$S = \begin{bmatrix} S_1 & S_2 \\ * & S_3 \end{bmatrix} > 0 \tag{12}$$

$$\Theta_1 = \begin{bmatrix} W_1 & \tilde{W}_1 \\ * & W_1 \end{bmatrix} > 0 \tag{13}$$

$$\Theta_2 = \begin{bmatrix} W_2 & \tilde{W}_2 \\ * & W_2 \end{bmatrix} > 0 \tag{14}$$

$$\Xi = \begin{bmatrix} \tilde{\Xi}_1 & \tilde{\Xi}_2 \\ * & \tilde{\Xi}_3 \end{bmatrix} < 0 \tag{15}$$

where

$$\tilde{\Xi}_1 = \begin{bmatrix} \Delta_{11} & \Delta_{12} & \Delta_{13} & -P_3 - MB & \tilde{W}_1^* & \tilde{W}_2^* & W_1 - \tilde{W}_1^* & W_2 - \tilde{W}_2^* \\ * & \Delta_{22} & \Delta_{23} & -P_5 - MB & 0 & 0 & 0 & 0 \\ * & * & -M - M^* & -MB & 0 & 0 & 0 & 0 \\ * & * & * & -Q_1 & 0 & 0 & 0 & 0 \\ * & * & * & * & -W_1 & 0 & W_1 - \tilde{W}_1 & 0 \\ * & * & * & * & * & -W_2 & 0 & W_2 - \tilde{W}_2 \\ * & * & * & * & * & * & \Delta_{77} & 0 \\ * & * & * & * & * & * & * & \Delta_{88} \end{bmatrix}$$

$$\tilde{\Xi}_2 = \begin{bmatrix} P_6 & S_1 + S_2^* & S_2 + S_3 & MC & MD & ME & -\bar{H}G & -\bar{K}G \\ P_3 & 0 & 0 & MC & MD & ME & -\bar{H}G & -\bar{K}G \\ P_5 & 0 & 0 & MC & MD & ME & -\bar{H}G & -\bar{K}G \\ -P_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -S_1 & -S_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & -S_2^* & -S_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\tilde{\Xi}_3 = \begin{bmatrix} -Q_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ * & -U_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ * & * & -U_2 & 0 & 0 & 0 & 0 & 0 \\ * & * & * & \beta^2 R - \Lambda_1 & 0 & 0 & 0 & 0 \\ * & * & * & * & -\Lambda_2 & 0 & 0 & 0 \\ * & * & * & * & * & -R & 0 & 0 \\ * & * & * & * & * & * & -\Lambda_3 & 0 \\ * & * & * & * & * & * & * & -\Lambda_4 \end{bmatrix}$$

with $\Delta_{11} = P_3 + P_3^* + Q_1 + \delta^2 Q_2 + \tau_1 U_1 + \tau U_2 - W_1 - W_2 + L\Lambda_1 L + N\Lambda_3 N^T - \bar{H}F - F^*\bar{H}^*$, $\Delta_{12} = P_1 + P_5^* - MA - \bar{K}F - F^*\bar{H}^*$, $\Delta_{13} = P_2 - M - F^*\bar{H}^*$, $\Delta_{22} = P_2 + P_2^* + \tau_1^2 W_1 + \tau^2 W_2 - MA - \bar{K}F - F^*\bar{K}^* - A^T M^* + N\Lambda_4 N^T$, $\Delta_{23} = P_4 - M - F^*\bar{K}^* - A^T M^*$, $\Delta_{77} = -2W_1 + \tilde{W}_1 + \tilde{W}_1^*$, $\Delta_{88} = -2W_2 + \tilde{W}_2 + \tilde{W}_2^* + L\Lambda_2 L$. *In this case, the estimator gain matrix is given by* $K = M^{-1}\bar{K}, H = M^{-1}\bar{H}$.

Proof. Consider the following Lyapunov functional:

$$V(t) = \sum_{n=1}^{6} V_n(t) \tag{16}$$

where

$$\begin{aligned}
V_1(t) &= \begin{bmatrix} e(t) \\ \dot{e}(t) \\ \int_{t-\delta}^{t} e(s)ds \end{bmatrix}^* \begin{bmatrix} P_1 & P_2 & P_3 \\ * & P_4 & P_5 \\ * & * & P_6 \end{bmatrix} \begin{bmatrix} e(t) \\ \dot{e}(t) \\ \int_{t-\delta}^{t} e(s)ds \end{bmatrix} \\
V_2(t) &= \int_{t-\delta}^{t} e^*(s)Q_1e(s)ds + \delta \int_{-\delta}^{0}\int_{t+\epsilon}^{t} e^*(s)Q_2e(s)dsd\epsilon \\
V_3(t) &= \begin{bmatrix} \int_{t-\tau_1}^{t} e(s)ds \\ \int_{t-\tau}^{t} e(s)ds \end{bmatrix}^* \begin{bmatrix} S_1 & S_2 \\ * & S_3 \end{bmatrix} \begin{bmatrix} \int_{t-\tau_1}^{t} e(s)ds \\ \int_{t-\tau}^{t} e(s)ds \end{bmatrix} \\
V_4(t) &= \int_{-\tau_1}^{0}\int_{t+\epsilon}^{t} e^*(s)U_1e(s)dsd\epsilon + \int_{-\tau}^{0}\int_{t+\epsilon}^{t} e^*(s)U_2e(s)dsd\epsilon \\
V_5(t) &= \tau_1 \int_{-\tau_1}^{0}\int_{t+\epsilon}^{t} \dot{e}^*(s)W_1\dot{e}(s)dsd\epsilon + \tau \int_{-\tau}^{0}\int_{t+\epsilon}^{t} \dot{e}^*(s)W_2\dot{e}(s)dsd\epsilon \\
V_6(t) &= \beta \int_{-\beta}^{0}\int_{t+\epsilon}^{t} \tilde{f}^*(e(s))R\tilde{f}(e(s))dsd\epsilon.
\end{aligned} \tag{17}$$

Taking the derivative of $V(t)$ and based on Lemma 1, we can obtain

$$\begin{aligned}
\dot{V}_1(t) = {} & e^*(t)(P_3 + P_3^*)e(t) + e^*(t)(P_1 + P_5^*)\dot{e}(t) + e^*(t)P_2\ddot{e}(t) - e^*(t)P_3e(t-\delta) \\
& + \dot{e}^*(t)(P_1 + P_5)e(t) + \dot{e}^*(t)(P_2 + P_2^*)\dot{e}(t) + \dot{e}^*(t)P_4\ddot{e}(t) - \dot{e}^*(t)P_5e(t-\delta) \\
& + \ddot{e}^*(t)P_2^*e(t) + \ddot{e}^*(t)P_4\dot{e}(t) - e^*(t-\delta)P_3^*e(t) - e^*(t-\delta)P_5^*\dot{e}(t) \\
& + e^*(t)P_6\left[\int_{t-\delta}^{t} e(s)ds\right] + \dot{e}^*(t)P_3\left[\int_{t-\delta}^{t} e(s)ds\right] + \ddot{e}^*(t)P_5\left[\int_{t-\delta}^{t} e(s)ds\right] \\
& - e^*(t-\delta)P_6\left[\int_{t-\delta}^{t} e(s)ds\right] + \left[\int_{t-\delta}^{t} e(s)ds\right]^* P_3^*\dot{e}(t) + \left[\int_{t-\delta}^{t} e(s)ds\right]^* P_5^*\ddot{e}(t) \\
& + \left[\int_{t-\delta}^{t} e(s)ds\right]^* P_6e(t) - \left[\int_{t-\delta}^{t} e(s)ds\right]^* P_6e(t-\delta)
\end{aligned} \tag{18}$$

$$\begin{aligned}
\dot{V}_2(t) &= e^*(t)Q_1e(t) - e^*(t-\delta)Q_1e(t-\delta) + \delta^2e^*(t)Q_2e(t) - \delta\int_{t-\delta}^{t} e^*(s)Q_2e(s)ds \\
&\le e^*(t)(Q_1 + \delta^2Q_2)e(t) - e^*(t-\delta)Q_1e(t-\delta) - \left[\int_{t-\delta}^{t} e(s)ds\right]^* Q_2\left[\int_{t-\delta}^{t} e(s)ds\right]
\end{aligned} \tag{19}$$

$$\begin{aligned}
\dot{V}_3(t) = {} & e^*(t)(S_1 + S_2^*)\left[\int_{t-\tau_1}^{t} e(s)ds\right] + e^*(t)(S_2 + S_3)\left[\int_{t-\tau}^{t} e(s)ds\right] \\
& - e^*(t-\tau_1)S_1\left[\int_{t-\tau_1}^{t} e(s)ds\right] - e^*(t-\tau_1)S_2\left[\int_{t-\tau}^{t} e(s)ds\right] \\
& - e^*(t-\tau)S_2^*\left[\int_{t-\tau_1}^{t} e(s)ds\right] - e^*(t-\tau)S_3\left[\int_{t-\tau}^{t} e(s)ds\right] \\
& + \left[\int_{t-\tau_1}^{t} e(s)ds\right]^* (S_1 + S_2)e(t) + \left[\int_{t-\tau}^{t} e(s)ds\right]^* (S_2^* + S_3)e(t) \\
& - \left[\int_{t-\tau_1}^{t} e(s)ds\right]^* S_1e(t-\tau_1) - \left[\int_{t-\tau_1}^{t} e(s)ds\right]^* S_2e(t-\tau) \\
& - \left[\int_{t-\tau}^{t} e(s)ds\right]^* S_2^*e(t-\tau_1) - \left[\int_{t-\tau}^{t} e(s)ds\right]^* S_3e(t-\tau)
\end{aligned} \tag{20}$$

$$\begin{aligned}\dot{V}_4(t) &= e^*(t)(\tau_1 U_1 + \tau U_2)e(t) - \tau_1 \int_{t-\tau_1}^{t} e^*(s)U_1 e(s)ds - \tau \int_{t-\tau}^{t} e^*(s)U_2 e(s)ds \\ &\le e^*(t)(\tau_1 U_1 + \tau U_2)e(t) - \left[\int_{t-\tau_1}^{t} e(s)ds\right]^* U_1 \left[\int_{t-\tau_1}^{t} e(s)ds\right] \\ &\quad - \left[\int_{t-\tau}^{t} e(s)ds\right]^* U_2 \left[\int_{t-\tau}^{t} e(s)ds\right]\end{aligned} \tag{21}$$

$$\dot{V}_5(t) = \dot{e}^*(t)(\tau_1^2 W_1 + \tau^2 W_2)\dot{e}(t) - \tau_1 \int_{t-\tau_1}^{t} \dot{e}^*(s)W_1\dot{e}(s)ds - \tau \int_{t-\tau}^{t} \dot{e}^*(s)W_2\dot{e}(s)ds \tag{22}$$

$$\begin{aligned}\dot{V}_6(t) &= \beta^2 \tilde{f}^*(e(t))R\tilde{f}(e(t)) - \beta \int_{t-\beta}^{t} \tilde{f}^*(e(s))R\tilde{f}(e(s))ds \\ &\le \beta^2 \tilde{f}^*(e(t))R\tilde{f}(e(t)) - \left[\int_{t-\beta}^{t} \tilde{f}(e(s))ds\right]^* R \left[\int_{t-\beta}^{t} \tilde{f}(e(s))ds\right].\end{aligned} \tag{23}$$

On the basis of Lemmas 1 and 2, we further obtain that

$$\begin{aligned}&-\tau_1 \int_{t-\tau_1}^{t} \dot{e}^*(s)W_1\dot{e}(s)ds = -\tau_1 \int_{t-\tau_1}^{t-\tau_1(t)} \dot{e}^*(s)W_1\dot{e}(s)ds - \tau_1 \int_{t-\tau_1(t)}^{t} \dot{e}^*(s)W_1\dot{e}(s)ds \\ &\le -\frac{\tau_1}{\tau_1 - \tau_1(t)} \left[\int_{t-\tau_1}^{t-\tau_1(t)} \dot{e}(s)ds\right]^* W_1 \left[\int_{t-\tau_1}^{t-\tau_1(t)} \dot{e}(s)ds\right] \\ &\quad - \frac{\tau_1}{\tau_1(t)} \left[\int_{t-\tau_1(t)}^{t} \dot{e}(s)ds\right]^* W_1 \left[\int_{t-\tau_1(t)}^{t} \dot{e}(s)ds\right] \\ &= -\frac{\tau_1}{\tau_1 - \tau_1(t)} \begin{bmatrix} e(t) \\ e(t-\tau_1) \\ e(t-\tau_1(t)) \end{bmatrix}^* \begin{bmatrix} 0 \\ -I \\ I \end{bmatrix} W_1 \begin{bmatrix} 0 \\ -I \\ I \end{bmatrix}^* \begin{bmatrix} e(t) \\ e(t-\tau_1) \\ e(t-\tau_1(t)) \end{bmatrix} \\ &\quad - \frac{\tau_1}{\tau_1(t)} \begin{bmatrix} e(t) \\ e(t-\tau_1) \\ e(t-\tau_1(t)) \end{bmatrix}^* \begin{bmatrix} I \\ 0 \\ -I \end{bmatrix} W_1 \begin{bmatrix} I \\ 0 \\ -I \end{bmatrix}^* \begin{bmatrix} e(t) \\ e(t-\tau_1) \\ e(t-\tau_1(t)) \end{bmatrix} \\ &\le - \begin{bmatrix} e(t) \\ e(t-\tau_1) \\ e(t-\tau_1(t)) \end{bmatrix}^* \begin{bmatrix} 0 & I \\ -I & 0 \\ I & -I \end{bmatrix} \begin{bmatrix} W_1 & \tilde{W}_1 \\ * & W_1 \end{bmatrix} \begin{bmatrix} 0 & I \\ -I & 0 \\ I & -I \end{bmatrix}^* \begin{bmatrix} e(t) \\ e(t-\tau_1) \\ e(t-\tau_1(t)) \end{bmatrix} \\ &= - \begin{bmatrix} e(t) \\ e(t-\tau_1) \\ e(t-\tau_1(t)) \end{bmatrix}^* \begin{bmatrix} W_1 & -\tilde{W}_1^* & -W_1 + \tilde{W}_1^* \\ * & W_1 & -W_1 + \tilde{W}_1 \\ * & * & 2W_1 - \tilde{W}_1 - \tilde{W}_1^* \end{bmatrix} \begin{bmatrix} e(t) \\ e(t-\tau_1) \\ e(t-\tau_1(t)) \end{bmatrix}.\end{aligned} \tag{24}$$

Similar to (24), we have that

$$\begin{aligned}&-\tau \int_{t-\tau}^{t} \dot{e}^*(s)W_2\dot{e}(s)ds \\ &\quad \le - \begin{bmatrix} e(t) \\ e(t-\tau) \\ e(t-\tau(t)) \end{bmatrix}^* \begin{bmatrix} W_2 & -\tilde{W}_2^* & -W_2 + \tilde{W}_2^* \\ * & W_2 & -W_2 + \tilde{W}_2 \\ * & * & 2W_2 - \tilde{W}_2 - \tilde{W}_2^* \end{bmatrix} \begin{bmatrix} e(t) \\ e(t-\tau) \\ e(t-\tau(t)) \end{bmatrix}.\end{aligned} \tag{25}$$

Adding (24) and (25) to (22), then we can have

$$\begin{aligned}\dot{V}_5(t) &\le \dot{e}^*(t)(\tau_1^2 W_1 + \tau^2 W_2)\dot{e}(t) \\ &\quad - \begin{bmatrix} e(t) \\ e(t-\tau_1) \\ e(t-\tau_1(t)) \end{bmatrix}^* \begin{bmatrix} W_1 & -\tilde{W}_1^* & -W_1 + \tilde{W}_1^* \\ * & W_1 & -W_1 + \tilde{W}_1 \\ * & * & 2W_1 - \tilde{W}_1 - \tilde{W}_1^* \end{bmatrix} \begin{bmatrix} e(t) \\ e(t-\tau_1) \\ e(t-\tau_1(t)) \end{bmatrix}\end{aligned}$$

$$-\begin{bmatrix} e(t) \\ e(t-\tau) \\ e(t-\tau(t)) \end{bmatrix}^{*} \begin{bmatrix} W_2 & -\tilde{W}_2^{*} & -W_2+\tilde{W}_2^{*} \\ * & W_2 & -W_2+\tilde{W}_2 \\ * & * & 2W_2-\tilde{W}_2-\tilde{W}_2^{*} \end{bmatrix} \begin{bmatrix} e(t) \\ e(t-\tau) \\ e(t-\tau(t)) \end{bmatrix}. \tag{26}$$

In addition, by means of Assumptions 1 and 2, we can obtain

$$\begin{aligned} &0 \le e^*(t)L\Lambda_1 Le(t) - \tilde{f}^*(e(t))\Lambda_1\tilde{f}(e(t)) \\ &0 \le e^*(t-\tau(t))L\Lambda_2 Le(t-\tau(t)) - \tilde{f}^*(e(t-\tau(t)))\Lambda_2\tilde{f}(e(t-\tau(t))) \\ &0 \le e^*(t)N\Lambda_3 N^T e(t) - \tilde{g}^*(e(t))\Lambda_3\tilde{g}(e(t)) \\ &0 \le e^*(t)N\Lambda_4 N^T e(t) - \tilde{g}^*(\dot{e}(t))\Lambda_4\tilde{g}(\dot{e}(t)). \end{aligned} \tag{27}$$

Then, using the free weighting matrix, it follows from (6) that

$$0 = [M^*e(t) + M^*\dot{e}(t) + M^*\ddot{e}(t)]^*P + P^*[M^*e(t) + M^*\dot{e}(t) + M^*\ddot{e}(t)] \tag{28}$$

where

$$\begin{aligned} P = &-\ddot{e}(t) - (A+KF)\dot{e}(t) - HFe(t) - Be(t-\delta) + C\tilde{f}(e(t)) + D\tilde{f}(e(t-\tau_1(t)-\tau_2(t))) \\ &+ E\int_{t-\beta}^{t}\tilde{f}(e(s))ds - KG\tilde{g}(\dot{e}(t)) - HG\tilde{g}(e(t)). \end{aligned}$$

Nothing that $K = M^{-1}\bar{K}, H = M^{-1}\bar{H}$, from (19)–(21), (23), (26)–(28), we can obtain that

$$\dot{V}(t) \le \xi^*(t)\Xi\xi(t) \tag{29}$$

where

$$\begin{aligned} \xi(t) = \quad & [e^*(t)\dot{e}^*(t)\ddot{e}^*(t)e^*(t-\delta)e^*(t-\tau_1)e^*(t-\tau)e^*(t-\tau_1(t))e^*(t-\tau(t)) \\ & \textstyle\int_{t-\delta}^{t} e^*(s)ds \int_{t-\tau_1}^{t} e^*(s)ds \int_{t-\tau}^{t} e^*(s)ds\tilde{f}(e(t))\tilde{f}(e^*(t-\tau(t))) \int_{t-\beta}^{t}\tilde{f}(e^*(s))ds \\ & \tilde{g}^*(e(t))\tilde{g}^*(\dot{e}(t))]^*. \end{aligned}$$

Then, according to (15), one has

$$\dot{V}(t) \le 0. \tag{30}$$

Thus, the error system (6) is globally asymptotically stable. The proof is complete. □

Remark 2. *In the process of obtaining the desired result, instead of reducing the order, the original second-order system is considered as the subject. This results in that the constructed Lyapunov functional (16) includes the state derivatives, which is different from the existing ones in [42,43]. This is an important feature of this paper. Moreover, as far as we know, this is the first time that the state estimation of complex-valued inertial neural networks with multiple time delays has been discussed.*

Remark 3. *In order to maintain the original characteristics of the system and without increasing its complexity, besides adopting the non-reduced order, we also apply the nonseparable method. We regard the system (1) as a whole in the complex domain to study the state estimation problem. Thus, our result is more universal and flexible.*

4. Numerical Examples

In this section, we give two numerical examples to verify the effectiveness of the above theoretical result.

Example 1. *Consider System (4) with the following parameters:*

$$A = \begin{bmatrix} 0.8 & 0 \\ 0 & 2.8 \end{bmatrix}, B = \begin{bmatrix} 0.9 & 0 \\ 0 & 1.1 \end{bmatrix}, C = \begin{bmatrix} -0.5+i & -0.3+1.2i \\ -0.3-0.2i & 1-0.2i \end{bmatrix},$$

$$D = \begin{bmatrix} -0.3+0.7i & 1.1+0.5i \\ 0.3+0.2i & -0.5+0.5i \end{bmatrix}, E = \begin{bmatrix} -0.2+0.8i & 0.5-0.8i \\ 0.3+0.2i & -0.2-0.8i \end{bmatrix}.$$

In addition, we select the activation functions as $f_q(u_q(t)) = \frac{1-e^{-u_q}}{1+e^{-u_q}}$ $(q = 1,2)$, $\delta = 0.1$, $\tau_1(t) = 0.2\sin^2(t)$, $\tau_2(t) = 0.5\cos^2(t)$, *and* $\beta = 0.1$, *which mean that* $\tau_1 = 0.2$ *and* $\tau_2 = 0.5$. *It could be easily tested that Assumptions 1 and 2 are satisfied, and* $L = \mathrm{diag}(0.5, 0.5)$,

$$N = \begin{bmatrix} 0 & 0.1 \\ -0.2 & 0 \end{bmatrix}.$$

Moreover, choose the initial values as $\eta(s) = [4.7-3i; -9+8.9i]^T$, $\tilde{\eta}(s) = [-6+4.9i; 2.2+2.1i]^T$, $\rho(s) = [-2.1+7.2i; 2.9+5.1i]^T$, *and* $\tilde{\rho}(s) = [1.3+3.5i; 6.6-3.2i]^T$, *for* $s \in [-0.7, 0]$. *We select the output weighting matrices and neuron-dependent nonlinear disturbance signals as*

$$F = \begin{bmatrix} -0.2+0.5i & -0.2-0.4i \\ 0.1-0.2i & 0.4-0.3i \end{bmatrix}, G = \begin{bmatrix} 0.5-0.2i & 0.2-0.2i \\ -0.3+0.1i & 0.3+0.1i \end{bmatrix},$$

and $g_q(u_q(t)) = u_q$ $(q = 1,2)$.

Then, by means of the MATLAB LMI toolbox, the feasible solutions of (11)–(15) can be obtained, such as $\Lambda_1 = \mathrm{diag}(0.9615, 0.9615)$, $\Lambda_2 = \mathrm{diag}(0.6049, 0.6049)$, $\Lambda_3 = \mathrm{diag}(6.4005, 6.4005)$, $\Lambda_4 = \mathrm{diag}(10.9396, 10.9396)$.

$$P_1 = \begin{bmatrix} 1.3341+0.0000i & 0.0015+0.0015i \\ 0.0015-0.0015i & 4.0161+0.0000i \end{bmatrix}, \quad P_2 = \begin{bmatrix} 0.4473-0.0006i & 0.0008+0.0156i \\ 0.0008+0.0156i & 1.2160+0.0236i \end{bmatrix},$$

$$P_3 = \begin{bmatrix} -0.3097-0.0029i & -0.0226-0.0176i \\ -0.0226-0.0176i & -0.8992-0.0027i \end{bmatrix}, \quad P_4 = \begin{bmatrix} 0.8317+0.0000i & 0.0093+0.0156i \\ 0.0093-0.0156i & 2.4997+0.0000i \end{bmatrix},$$

$$P_5 = \begin{bmatrix} -0.1311-0.0024i & -0.0243-0.0361i \\ -0.0243-0.0361i & -0.4271-0.0211i \end{bmatrix}, \quad P_6 = \begin{bmatrix} 0.6252+0.0000i & 0.0058-0.0029i \\ 0.0058+0.0029i & 1.3612+0.0000i \end{bmatrix},$$

$$S_1 = \begin{bmatrix} 1.0883+0.0000i & -0.0019+0.8317i \\ -0.0019-0.8317i & 3.0778+0.0000i \end{bmatrix}, \quad S_2 = \begin{bmatrix} -0.0013-0.0018i & 0.0015-0.0027i \\ 0.0015-0.0027i & 0.0330+0.0213i \end{bmatrix},$$

$$S_3 = \begin{bmatrix} 0.1133+0.0000i & -0.0005-0.0243i \\ -0.0005+0.0243i & 0.4461+0.0000i \end{bmatrix}, \quad Q_1 = \begin{bmatrix} 0.3635+0.0000i & 0.0186-0.0361i \\ 0.0186+0.0361i & 1.1520+0.0000i \end{bmatrix},$$

$$Q_2 = \begin{bmatrix} 3.9982+0.0000i & 0.1264+0.0058i \\ 0.1264-0.0058i & 5.3500+0.0000i \end{bmatrix}, \quad W_1 = \begin{bmatrix} 3.1818+0.0000i & -0.0089+0.0330i \\ -0.0089-0.0330i & 4.8098+0.0000i \end{bmatrix},$$

$$W_2 = \begin{bmatrix} 0.7340+0.0000i & -0.0039+0.0213i \\ -0.0039-0.0213i & 1.8923+0.0000i \end{bmatrix}, \quad R = \begin{bmatrix} 3.3062+0.0000i & -0.4961+0.4461i \\ -0.4961-0.4461i & 3.8736+0.0000i \end{bmatrix},$$

$$\tilde{W}_1 = \begin{bmatrix} 0.9734+0.0002i & -0.0027+0.2611i \\ -0.0059-0.2518i & 1.7210+0.0003i \end{bmatrix}, \quad \tilde{W}_2 = \begin{bmatrix} 0.0745+0.0001i & -0.0009+0.0266i \\ -0.0011-0.0380i & 0.3236-0.0001i \end{bmatrix},$$

$$M = \begin{bmatrix} 0.1121+0.0007i & 0.0321-0.0026i \\ 0.0327+0.0113i & 0.3273+0.0046i \end{bmatrix}, \quad \bar{K} = \begin{bmatrix} -0.1284-1.3291i & 1.0282+0.0685i \\ 0.6352+1.0851i & 1.7107+2.4569i \end{bmatrix},$$

$$\bar{H} = \begin{bmatrix} -0.0382-0.7650i & 0.6966+0.0670i \\ 0.4376+0.7508i & 1.2550+1.7039i \end{bmatrix}.$$

Consequently, it can be obtained from $K = M^{-1}\bar{K}$ *and* $H = M^{-1}\bar{H}$ *that*

$$K = \begin{bmatrix} -1.8353 - 13.1449i & 7.7034 - 1.4274i \\ 1.7354 + 4.6670i & 4.5110 + 7.3204i \end{bmatrix}, \ H = \begin{bmatrix} -0.8037 - 7.6728i & 5.1293 - 0.8042i \\ 1.1953 + 3.0710i & 3.3657 + 5.0622i \end{bmatrix}.$$

The simulation results are shown in Figures 2 and 3. Among them, Figure 2 plots the trajectories of the real and imaginary parts of the true state $u(t)$ in System (4) and the estimation state $v(t)$. Figure 3 shows the state trajectory of the error system (6). Obviously, the simulation results are in agreement with our theoretical analysis.

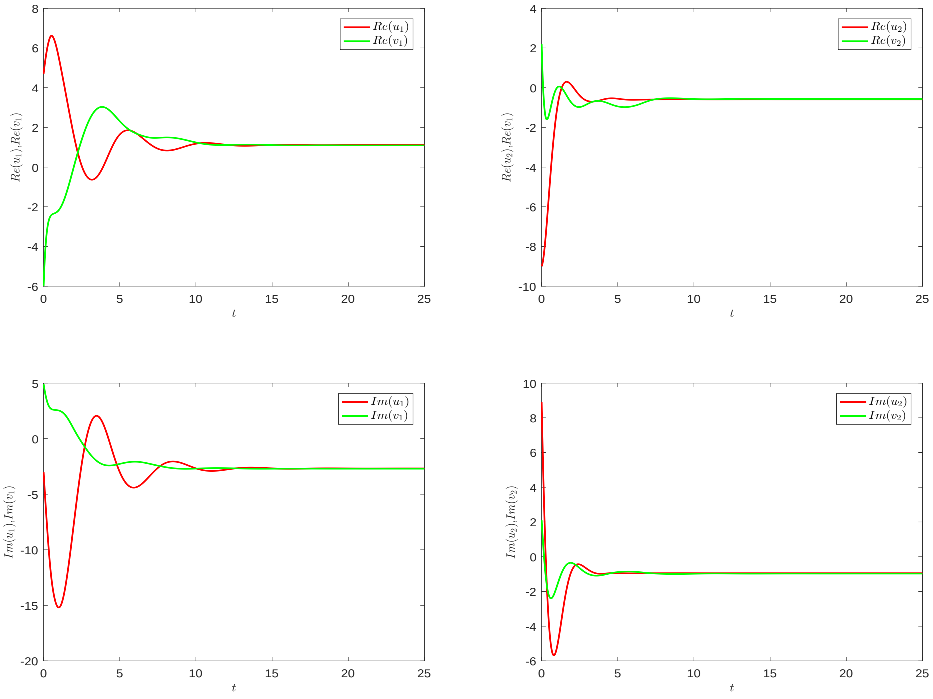

Figure 2. The curves of state $u(t)$ and its estimation $v(t)$ in Example 1.

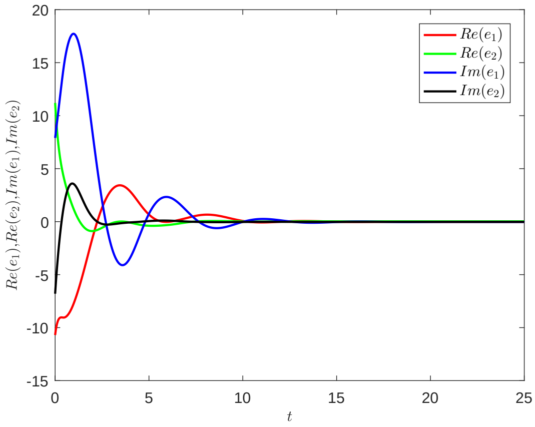

Figure 3. The curves dynamics of error system $e(t)$ in Example 1.

Example 2. *Consider System (4) with the following parameters:*

$$A = \begin{bmatrix} 0.7 & 0 \\ 0 & 2.6 \end{bmatrix}, \ B = \begin{bmatrix} 0.3 & 0 \\ 0 & 0.6 \end{bmatrix}, \ C = \begin{bmatrix} -0.5 - 0.2i & -0.3 - 0.4i \\ -0.3 + 0.2i & 1 - 0.2i \end{bmatrix},$$
$$D = \begin{bmatrix} -0.3 + 0.2i & 1.1 + 0.3i \\ 0.3 - 0.2i & -0.5 + 0.2i \end{bmatrix}, \ E = \begin{bmatrix} 0.3 + 0.8i & 0.8 - 0.8i \\ -0.5 + 0.2i & 1 - 0.8i \end{bmatrix}.$$

In addition, we select the same $f_q(u_q(t))$ as activation functions in Example 1, $\delta = 0.1$, $\tau_1(t) = 0.1|\sin(t)|$, $\tau_2(t) = 0.2|\cos(3t)|$, and $\beta = 0.1$, which mean that $\tau_1 = 0.1$ and $\tau_2 = 0.2$. It could be easily tested that Assumptions 1 and 2 are satisfied, and $L = \mathrm{diag}(0.5, 0.5)$,

$$N = \begin{bmatrix} 0 & 0.1 \\ -0.2 & 0 \end{bmatrix}.$$

Moreover, choose the initial values as $\eta(s) = [-5.6 - 1.1i; -1.9 + 1.2i]^T$, $\tilde{\eta}(s) = [0.2 + 0.3i; 0.3 + 0.9i]^T$, $\rho(s) = [-0.2 + 0.6i; -1.3 + 0.6i]^T$, and $\tilde{\rho}(s) = [2.4 - 1.1i; -1.6 - 1.1i]^T$, for $s \in [-0.3, 0]$. We select the output weighting matrices and neuron-dependent nonlinear disturbance signals as

$$F = \begin{bmatrix} 0.3 + 0.5i & 0.2 - 0.4i \\ 0.1 - 0.2i & 0.5 - 0.3i \end{bmatrix}, \; G = \begin{bmatrix} 0.5 - 0.2i & 0.2 - 0.2i \\ -0.3 + 0.1i & 0.3 + 0.1i \end{bmatrix},$$

and $g_q(u_q(t)) = u_q$ $(q = 1, 2)$.

Then, by means of the MATLAB LMI toolbox, the feasible solutions of (11)–(15) can be obtained, such as $\Lambda_1 = \mathrm{diag}(64.5684, 64.5684)$, $\Lambda_2 = \mathrm{diag}(55.5390, 55.5390)$, $\Lambda_3 = \mathrm{diag}(223.3662, 223.3662)$, $\Lambda_4 = \mathrm{diag}(227.1920, 227.1920)$.

$$P_1 = 1.0e+02 \begin{bmatrix} 0.6949 + 0.0000i & 0.0128 + 0.0128i \\ 0.0128 - 0.0128i & 1.6288 + 0.0000i \end{bmatrix}, \; P_2 = \begin{bmatrix} 23.8813 + 0.1538i & 1.5234 + 0.0078i \\ 1.5234 + 0.0078i & 50.3531 - 1.6239i \end{bmatrix},$$

$$P_3 = \begin{bmatrix} -14.7464 - 0.6364i & 0.8368 - 1.1136i \\ 0.8368 - 1.1136i & -30.9705 + 0.5937i \end{bmatrix}, \; P_4 = 1.0e+02 \begin{bmatrix} 0.4104 + 0.0000i & 0.0286 + 0.0001i \\ 0.0286 - 0.0001i & 1.0553 + 0.0000i \end{bmatrix},$$

$$P_5 = \begin{bmatrix} -8.1536 - 0.7565i & 0.0348 - 1.4125i \\ 0.0348 - 1.4125i & -16.2009 + 1.4115i \end{bmatrix}, \; P_6 = \begin{bmatrix} 31.3945 + 0.0000i & -1.2803 - 0.6364i \\ -1.2803 + 0.6364i & 49.7008 + 0.0000i \end{bmatrix},$$

$$S_1 = 1.0e+02 \begin{bmatrix} 0.7587 + 0.0000i & 0.0323 + 0.4104i \\ 0.0323 - 0.4104i & 1.2279 + 0.0000i \end{bmatrix}, \; S_2 = \begin{bmatrix} 0.0192 - 0.0506i & 0.1381 - 0.3017i \\ 0.1381 - 0.3017i & 2.5451 + 0.5451i \end{bmatrix},$$

$$S_3 = \begin{bmatrix} 19.7653 + 0.0000i & 1.1483 + 0.0348i \\ 1.1483 - 0.0348i & 29.9047 + 0.0000i \end{bmatrix}, \; Q_1 = \begin{bmatrix} 17.5145 + 0.0000i & -0.2291 - 1.4125i \\ -0.2291 + 1.4125i & 42.3430 + 0.0000i \end{bmatrix},$$

$$Q_2 = 1.0e+02 \begin{bmatrix} 1.6937 + 0.0000i & 0.0149 - 0.0128i \\ 0.0149 + 0.0128i & 1.8482 + 0.0000i \end{bmatrix}, \; W_1 = 1.0e+02 \begin{bmatrix} 1.3501 + 0.0000i & 0.0060 + 0.0255i \\ 0.0060 - 0.0255i & 1.4794 + 0.0000i \end{bmatrix},$$

$$W_2 = \begin{bmatrix} 93.4446 + 0.0000i & 1.0153 + 0.5451i \\ 1.0153 - 0.5451i & 96.3752 + 0.0000i \end{bmatrix}, \; R = 1.0e+02 \begin{bmatrix} 1.3938 + 0.0000i & 0.0229 + 0.2990i \\ 0.0229 - 0.2990i & 1.6984 + 0.0000i \end{bmatrix},$$

$$\tilde{W}_1 = \begin{bmatrix} 30.7114 - 47.2040i & 19.3940 + 27.8159i \\ 5.8570 + 24.8126i & 61.2309 + 33.2133i \end{bmatrix}, \; \tilde{W}_2 = \begin{bmatrix} 18.5470 - 41.1490i & 1.4340 + 39.2840i \\ -1.0151 + 26.5256i & 75.1281 + 35.4684i \end{bmatrix},$$

$$M = \begin{bmatrix} 14.5224 + 1.0902i & -1.8188 + 5.9497i \\ -0.3254 - 4.5011i & 22.3077 - 0.6569i \end{bmatrix}, \; \bar{K} = \begin{bmatrix} 39.3566 + 0.0305i & 0.2758 + 8.9574i \\ 0.2483 - 8.4581i & 45.8361 - 0.0401i \end{bmatrix},$$

$$\bar{H} = \begin{bmatrix} 17.4796 + 0.0089i & 0.3011 + 1.2179i \\ 0.3341 - 1.3773i & 17.7157 + 0.0188i \end{bmatrix}.$$

Consequently, it can be obtained from $K = M^{-1}\bar{K}$ and $H = M^{-1}\bar{H}$ that

$$K = \begin{bmatrix} 2.6277 - 3.6505i & 2.5831 + 0.9480i \\ 0.9898 + 1.6184i & 2.5294 + 2.0984i \end{bmatrix}, \; H = \begin{bmatrix} 1.7394 - 2.9958i & 1.4102 + 1.6175i \\ 0.5398 + 1.5122i & 3.0035 + 1.9865i \end{bmatrix}.$$

The simulation results are shown in Figures 4 and 5. Among them, Figure 4 plots the trajectories of the real and imaginary parts of the true state $u(t)$ in System (4) and the estimation state $v(t)$. Figure 5 shows the state trajectory of the error system (6). Obviously, the simulation results are in agreement with our theoretical analysis.

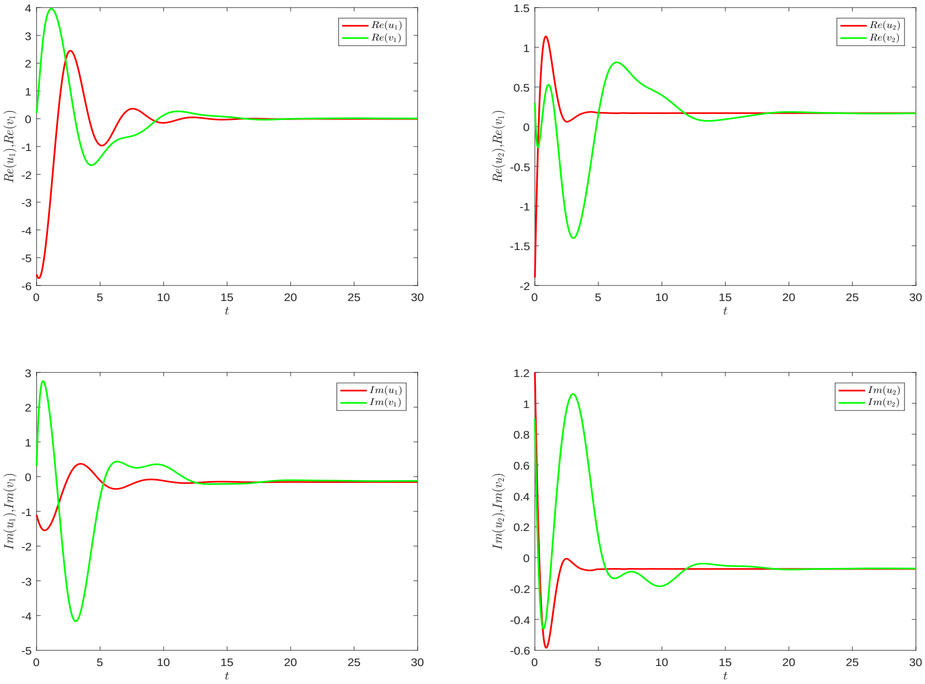

Figure 4. The curves of states $u(t)$ and its estimation $v(t)$ in Example 2.

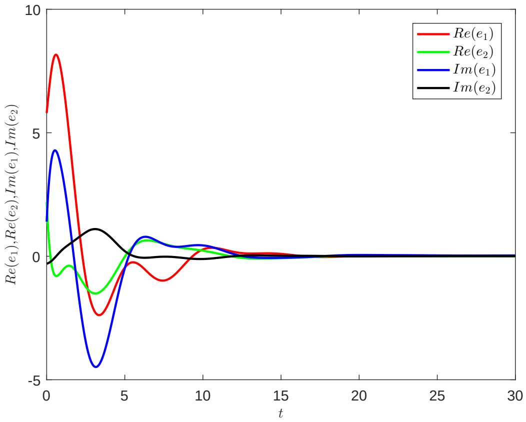

Figure 5. The curves dynamics of error system $e(t)$ in Example 2.

5. Conclusions

In this paper, the state estimation for complex-valued inertial neural networks with multiple time delays was studied. By constructing a suitable Lyapunov–Krasovskii functional and using some inequalities such as the Jensen inequality and the reciprocally convex inequality, a delay-dependent criterion based on linear matrix inequalities (LMIs) was obtained, which ensures the global asymptotic stability of the error system and that the network state can be estimated by observing the output measurements. Finally, two examples were given to verify the effectiveness of the proposed method. In the future, for various types of complex-valued neural network models with multiple delays, we will consider some synchronization phenomena, such as quasi-projective synchronization, polynomial synchronization, and so on.

Author Contributions: All authors contributed equally to the writing of the manuscript. All authors have read and approved the final manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (62173214), in part by the Natural Science Foundation of Shandong Province of China (ZR2021MF100), in part by the Research Fund for the Taishan Scholar Project of Shandong Province of China, in part by the Science and Technology Support Plan for Youth Innovation of Colleges and Universities of Shandong Province of China (2019KJI005), and in part by the SDUST Research Fund.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sharma, S.; Lingras, P.; Xu, F.; Kilburn, P. Application of neural networks to estimate AADT on low-volume roads. *J. Transp. Eng.* **2001**, *127*, 426–432. [CrossRef]
2. Balla, K.; Sevilla, R.; Hassan, O.; Morgan, K. An application of neural networks to the prediction of aerodynamic coefficients of aerofoils and wings. *Appl. Math. Model.* **2021**, *96*, 456–479. [CrossRef]
3. Jia, Q.; Mwanandiye, E.S.; Tang, W.K.S. Master-slave synchronization of delayed neural networks with time-varying control. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 2292–2298. [CrossRef]
4. Wang, X.; Cao, J.D.; Wang, J.T.; Qi, J. A novel fixed-time stability strategy and its application to fixed-time synchronization control of semi-Markov jump delayed neural networks. *Neurocomputing* **2021**, *452*, 284–293. [CrossRef]
5. Shi, C.Y.; Hoi, K.; Vong, S. Free-weighting-matrix inequality for exponential stability for neural networks with time-varying delay. *Neurocomputing* **2021**, *466*, 221–228. [CrossRef]
6. Yang, B.; Hao, M.N.; Cao, J.J.; Zhao, X. Delay-dependent global exponential stability for neural networks with time-varying delay. *Neurocomputing* **2019**, *338*, 172–180. [CrossRef]
7. Babcock, K.L.; Westervelt, R.M. Dynamics of simple electronic neural networks. *Phys. D Nonlinear Phenom.* **1987**, *28*, 305–316. [CrossRef]
8. Angelaki, D.; Correia, M. Models of membrane resonance in pigeon semicircular canal type II hair cells. *Biol. Cybern.* **1991**, *65*, 1–10. [CrossRef]
9. Wang, J.F.; Tian, L.X. Global Lagrange stability for inertial neural networks with mixed time-varying delays. *Neurocomputing* **2017**, *235*, 140–146. [CrossRef]
10. Chen, D.D.; Kong, F.C. Delay-dependent criteria for global exponential stability of time-varying delayed fuzzy inertial neural networks. *Neural Comput. Appl.* **2021**, *53*, 49–68. [CrossRef]
11. Shi, J.C.; Zeng, Z.G. Global exponential stabilization and lag synchronization control of inertial neural networks with time delays. *Neural Netw.* **2020**, *126*, 11–20. [CrossRef] [PubMed]
12. Wang, Z.D.; Ho, D.W.C.; Liu, X.H. State estimation for delayed neural networks. *IEEE Trans. Neural Netw.* **2005**, *16*, 279–284. [CrossRef] [PubMed]
13. Syed, A.M.; Saravanan, S.; Zhu, Q.X. Non-fragile finite-time H_∞ state estimation of neural networks with distributed time-varying delay. *J. Frankl. Inst.* **2017**, *354*,7566–7584. [CrossRef]
14. Ren, J.; Liu, X.; Zhu, H.; Zhong, S.; Shi, K. State estimation of neural networks with two Markovian jumping parameters and multiple time delays. *J. Frankl. Inst.* **2017**, *354*, 812–833. [CrossRef]
15. Tan, G.Q.; Wang, J.D.; Wang, Z.S. A new result on L_2–L_∞ performance state estimation of neural networks with time-varying delay: A new result on $L_2 - L_\infty$ performance state estimation of neural networks with time-varying delay. *Neurocomputing* **2020**, *398*, 166–171. [CrossRef]
16. Tan, G.Q.; Wang, J.D.; Wang, Z.S. Extended dissipativity state estimation for generalized neural networks with time-varying delay via delay-product-type functionals and integral inequality. *Neurocomputing* **2021**, *455*, 78–87. [CrossRef]
17. Sun, L.; Su, L.; Wang, J. Non-fragile dissipative state estimation for semi-Markov jump inertial neural networks with reaction-diffusion. *Appl. Math. Comput.* **2021**, *411*, 126404. [CrossRef]
18. Wang, J.; Hu, X.H.; Cao, J.D.; Park, J.H.; Shen, H. H_∞ state estimation for switched inertial neural networks with time-varying delays: A persistent dwell-time scheme. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *228*, 1–11. [CrossRef]
19. Gong, W.Q.; Liang, J.L.; Kan, X.; Nie, X. Robust state estimation for delayed complex-valued neural networks. *Neural Process. Lett.* **2017**, *46*, 1009–1029. [CrossRef]
20. Zhang, Z.Q.; Zheng, T. Global asymptotic stability of periodic solutions for delayed complex-valued Cohen–Grossberg neural networks by combining coincidence degree theory with LMI method. *Neurocomputing* **2018**, *289*, 220–230. [CrossRef]
21. Liang, J.; Li, K.L.; Song, Q.K.; Zhao, Z.; Liu, Y.; Alsaadi, F.E. State estimation of complex-valued neural networks with two additive time-varying delays. *Neurocomputing* **2018**, *309*, 54–61. [CrossRef]
22. Gong, W.Q.; Liang, J.L.; Kan, X.; Wang, L.; Dobaie, A.M. Robust state estimation for stochastic complex-valued neural networks with sampled-data. *Neural Comput. Appl.* **2019**, *31*, 523–542. [CrossRef]

23. Wan, P.; Jian, J.G. Global mittag-leffler boundedness for fractional-order complex-valued cohen-grossberg neural networks. *Neural Process. Lett.* **2019**, *49*, 121–139. [CrossRef]
24. Wang, S.Z.; Zhang, Z.Y.; Lin, C.; Chen, J. Fixed-time synchronization for complex-valued BAM neural networks with time-varying delays via pinning control and adaptive pinning control. *Chaos Soliton. Fractals* **2021**, *153*, 111583. [CrossRef]
25. Gunasekaran, N.; Zhai, G.S. Sampled-data state-estimation of delayed complex-valued neural networks. *Int. J. Syst. Sci* **2020**, *51*, 303–312. [CrossRef]
26. Hu, B.X.; Song, Q.K.; Zhao, Z.Q. Robust state estimation for fractional-order complex-valued delayed neural networks with interval parameter uncertainties: LMI approach. *Appl. Math. Comput.* **2020**, *373*, 1–12. [CrossRef]
27. Aouiti, C.; Bessifi, M. Sliding mode control for finite-time and fixed-time synchronization of delayed complex-valued recurrent neural networks with discontinuous activation functions and nonidentical parameters. *Eur. J. Control* **2021**, *59*, 109–122. [CrossRef]
28. Kumar, A.; Das, S.; Yadav, V.K. Global quasi-synchronization of complex-valued recurrent neural networks with time-varying delay and interaction terms. *Chaos Soliton. Frac.* **2021**, *152*, 111323. [CrossRef]
29. Ding, Z.X.; Zhang, H.; Zeng, Z.; Yang, L.; Li, S. Global dissipativity and quasi-mittag-leffler synchronization of fractional-order discontinuous complex-valued neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–14. [CrossRef]
30. Liu, X.; Yu, Y.G. Synchronization analysis for discrete fractional-order complex-valued neural networks with time delays. *Neural Comput. Appl.* **2021**, *33*, 10503–10514. [CrossRef]
31. Zhang, Z.Y.; Guo, R.N.; Liu, X.P.; Zhong, M.; Lin, C.; Chen, B. Fixed-time synchronization for complex-valued BAM neural networks with time delays. *Asian J. Control* **2021**, *23*, 298–314. [CrossRef]
32. Qiu, B.; Liao, X.F.; Zhou, B. State estimation for complex-valued neural networks with time-varying delays. In Proceedings of the Sixth International Conference on Intelligent Control and Information Processing (ICICIP), Wuhan, China, 26–28 November 2015; pp. 531–536.
33. Guo, R.N.; Lu, J.W.; Li, Y.M.; Lv, W. Fixed-time synchronization of inertial complex-valued neural networks with time delays. *Nonlinear Dyn.* **2021**, *105*, 1643–1656. [CrossRef]
34. Tang, Q.; Jian, J.G. Global exponential convergence for impulsive inertial complex-valued neural networks with time-varying delays. *Math. Comput. Simul.* **2019**, *159*, 39–56. [CrossRef]
35. Li, X.F.; Huang, T.W. Adaptive synchronization for fuzzy inertial complex-valued neural networks with state-dependent coefficients and mixed delays. *Fuzzy Sets Syst.* **2021**, *411*, 174–189. [CrossRef]
36. Yu, J.; Hu, C.; Jiang, H.J.; Wang, L. Exponential and adaptive synchronization of inertial complex-valued neural networks: A non-reduced order and non-separation approach. *Neural Netw.* **2021**, *124*, 50–59. [CrossRef]
37. Yu, Y.N.; Zhang, Z.Y.; Zhong, M.Y.; Wang, Z. Pinning synchronization and adaptive synchronization of complex-valued inertial neural networks with time-varying delays in fixed-time interval. *J. Frankl. Inst.* **2022**, *359*, 1434–1456. [CrossRef]
38. Guo, R.N.; Xu, S.Y.; Ma, Q.; Zhang, Z. Fixed-time synchronization of complex-valued inertial neural networks via nonreduced-order method. *IEEE Syst. J.* **2021**, 1–9. [CrossRef]
39. Long, C.Q.; Zhang, G.D.; Zeng, Z.G.; Hu, J. Finite-time stabilization of complex-valued neural networks with proportional delays and inertial terms: A non-separation approach. *Neural Netw.* **2022**, *148*, 86–95. [CrossRef]
40. Du, F.F.; Lu, J.G. New approach to finite-time stability for fractional-order BAM neural networks with discrete and distributed delays. *Chaos Soliton. Frac.* **2021**, *151*, 111225. [CrossRef]
41. Nagamani, G.; Rajan, G.S.; Zhu, Q.X. Exponential state estimation for memristor-based discrete-time BAM neural networks with additive delay components. *IEEE Trans. Cybern.* **2020**, *50*, 4281–4292. [CrossRef]
42. Chen, X.F.; Song, Q.K. State estimation for quaternion-valued neural networks with multiple time delays. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 2278–2287. [CrossRef]
43. Liu, L.B.; Chen, X.F. State estimation of quaternion-valued neural networks with leakage time delay and mixed two additive time-varying delays. *Neural Process. Lett.* **2020**, *51*, 2155–2178. [CrossRef]
44. Li, X.F.; Fang, J.A.; Huang, T.W. Event-triggered exponential stabilization for state-based switched inertial complex-valued neural networks with multiple delays. *IEEE Trans. Cybern.* **2020**. [CrossRef] [PubMed]
45. Wei, X.F.; Zhang, Z.Y.; Lin, C.; Chen, J. Synchronization and anti-synchronization for complex-valued inertial neural networks with time-varying delays. *Appl. Math. Comput.* **2021**, *403*, 126194. [CrossRef]