Adaptive Dynamic Programming for a Class of Complex-Valued Nonlinear Systems

Ruizhuo Song, Member, IEEE, Wendong Xiao, Senior Member, IEEE, Huaguang Zhang, Senior Member, IEEE, and Changyin Sun, Member, IEEE

Abstract—In this brief, an optimal control scheme based on adaptive dynamic programming (ADP) is developed to solve infinite-horizon optimal control problems of continuous-time complex-valued nonlinear systems. A new performance index function is established on the basis of complex-valued state and control. Using system transformations, the complex-valued system is transformed into a real-valued one, which overcomes Cauchy–Riemann conditions effectively. With the transformed system and the performance index function, a new ADP method is developed to obtain the optimal control law by using neural networks. A compensation controller is developed to compensate the approximation errors of neural networks. Stability properties of the nonlinear system are analyzed and convergence properties of the weights for neural networks are presented. Finally, simulation results demonstrate the performance of the developed optimal control scheme for complex-valued nonlinear systems.

Index Terms—Adaptive critic designs, adaptive dynamic programming (ADP), approximate complex-valued systems, dynamic programming, optimal control, neural networks.

I. INTRODUCTION

In many science problems and engineering applications, the parameters and signals are complex-valued [1], [2], such as quantum systems [3] and complex-valued neural networks [4]. In [5], complex-valued filters were proposed for complex signals and systems. In [6], a complex-valued B-spline neural network was proposed to model the complex-valued Wiener system. In [7], a complex-valued pipelined recurrent neural network for nonlinear adaptive prediction of complex nonlinear and nonstationary signals was proposed. In [8], the output feedback stabilization of complex-valued reaction-advection-diffusion systems was studied. In [4], the global asymptotic stability of delayed complex-valued recurrent neural networks was studied. In [9], a reinforcement learning algorithm with complex-valued functions was proposed. In the investigations of complex-valued systems, many system designers want to find the optimal value of the complex-valued parameters or controlled variable, by optimizing a chosen performance index function [10].

During last decades, adaptive dynamic programming (ADP), proposed by Werbos [11], [12], has demonstrated the powerful capability to find the optimal control law and solve the Hamilton-Jacobi-Bellman (HJB) equation forward-in-time [13]–[21]. There were several synonyms used for ADP, including adaptive critic designs [22]–[25], adaptive dynamic programming [26], [27], approximate dynamic programming [28], [29], neuro-dynamic programming [30], [31], and reinforcement learning [32], [33]. Until now, ADP has successfully solved nonlinear zero-sum/nonzero-sum differential games [34], [35], optimal tracking control problems [36], [37], multiagent control problems [38], and so on. From the previous discussions, it can be seen that the optimal control schemes based on ADP are constrained to real-valued systems. In many real-world systems, however, the system states and controls are complex values [39]. As there exist inherent differences between real-valued systems and complex-valued ones, the ADP methods for real-valued systems cannot solve the optimal control problems of complex-valued systems, directly. To the best of our knowledge, there are no discussions on ADP for complex-valued systems. Therefore, a novel ADP method for complex-valued systems is eagerly anticipated. This motivates our research.

In this brief, for the first time an ADP-based optimal control scheme for complex-valued systems is developed. First, a new performance index function is defined based on the complex-valued state and control. Second, using system transformations, the complex-valued system is transformed into a real-valued one, where the Cauchy–Riemann conditions can effectively be avoided. Then, a new ADP method is developed to obtain the optimal control law of the nonlinear systems. Neural networks, including critic and action networks, are employed to implement the developed ADP method. A compensation control method is established to overcome the approximation errors of neural networks. It is proven that the developed control scheme makes the closed-loop system uniformly ultimately bounded (UUB). It is also shown that the weights of neural networks will converge to a finite neighborhood of the optimal weights. Finally, the simulation studies are given to show the effectiveness of the developed control scheme.
II. Motivations and Preliminaries

Consider the following complex-valued nonlinear system:

$$\dot{z} = f(z) + g(z)u$$  \hspace{1cm} (1)

where $z \in \mathbb{C}^n$ is the system state, $f(z) \in \mathbb{C}^n$, and $f(0) = 0$. Let $g(z) \in \mathbb{C}^{n \times n}$ be a bounded input gain, that is, $\|g(z)\| \leq \bar{g}$, where $\bar{g}$ is a positive constant, and $\| \cdot \|$ is the two-norm, unless special declaration is given. Let $u \in \mathbb{C}^n$ be the control vector. Let $z_0$ be the initial state. For system (1), the infinite-horizon performance index function is defined as

$$J(z) = \int_0^\infty \bar{r}(z(\tau), u(\tau))d\tau$$  \hspace{1cm} (2)

where the utility function $\bar{r}(z, u) = z^H Q_1 z + u^H R_1 u$. Let $Q_1$ and $R_1$ be diagonal positive definite matrices. Let $z^H$ and $u^H$ denote the complex conjugate transpose of $z$ and $u$, respectively.

The aim of this brief is to obtain the optimal control of the complex-valued nonlinear system (1). To achieve this purpose, the following assumptions are necessary.

**Assumption 1** [4]: Let $i^2 = -1$, and then $z = x + iy$, where $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$. If $C(z) \in \mathbb{C}^n$ is the complex-valued function, then it can be expressed as $C(z) = C^R(x, y) + i C^I(x, y)$, where $C^R(x, y) \in \mathbb{R}^n$, and $C^I(x, y) \in \mathbb{R}^n$.

**Assumption 2**: Let $f(z) = f_R(z) + if_I(z)$, and $f_R(z) = f_R(x, y) + if_I(x, y)$, $j = 1, \ldots, n$. The partial derivatives of $f_j(z)$ with respect to $x$ and $y$ satisfy

$$\left\| \frac{\partial f_j}{\partial x} \right\|_1 \leq \dot{x}_j^R \hspace{1cm}, \hspace{1cm} \left\| \frac{\partial f_j}{\partial y} \right\|_1 \leq \dot{x}_j^I$$

and

$$\left\| \frac{\partial f_j}{\partial y} \right\|_1 \leq \dot{y}_j^R \hspace{1cm}, \hspace{1cm} \left\| \frac{\partial f_j}{\partial x} \right\|_1 \leq \dot{y}_j^I$$

where $\dot{x}_j^R$, $\dot{x}_j^I$, $\dot{y}_j^R$, and $\dot{y}_j^I$ are positive constants. Let $\| \cdot \|_1$ stand for one-norm.

According to above preparations, the system transformation for system (1) will be given. Let $f(z) = f_R(x, y) + if_I(x, y)$, $g(z) = g_R(x, y) + ig_I(x, y)$, and $u = u_R + iu_I$. Then, system (1) can be written as

$$\dot{x} + i \dot{y} = f_R(x, y) + if_I(x, y)$$

$$+(g_R(x, y) + ig_I(x, y))(u_R + iu_I).$$  \hspace{1cm} (3)

Let

$$\eta = \begin{bmatrix} x \\ y \end{bmatrix}, \hspace{0.5cm} \nu = \begin{bmatrix} u_R \\ u_I \end{bmatrix}, \hspace{0.5cm} F(\eta) = \begin{bmatrix} f_R(x, y) \\ f_I(x, y) \end{bmatrix}$$

and

$$G(\eta) = \begin{bmatrix} g_R(x, y) - g_I(x, y) \\ g_I(x, y) + g_R(x, y) \end{bmatrix}.$$  \hspace{1cm} (4)

Then, we can obtain

$$\dot{\eta} = F(\eta) + G(\eta)\nu$$  \hspace{1cm} (5)

where $\eta \in \mathbb{R}^{2n}$, $F(\eta) \in \mathbb{R}^{2n}$, $G(\eta) \in \mathbb{R}^{2n \times 2n}$, and $\nu \in \mathbb{R}^{2n}$. From (4) we can see that $F(0) = 0$.  

**Remark 1**: The system transformations between system (1) and system (4) are equivalent and reversible, which can be seen in the following equations:

$$\dot{\eta} = F(\eta) + G(\eta)\nu$$

$$\Leftrightarrow \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} f_R(x, y) + g_R(x, y)u_R - g_I(x, y)u_I \\ f_I(x, y) + g_I(x, y)u_R + g_R(x, y)u_I \end{bmatrix}$$

$$\Leftrightarrow \dot{x} + i \dot{y} = f_R(x, y) + if_I(x, y)$$

$$+(g_R(x, y) + ig_I(x, y))(u_R + iu_I)$$

$$\Leftrightarrow \dot{x} + i \dot{y} = f_R(x, y) + i f_I(x, y)$$

$$+(g_R(x, y) + ig_I(x, y))(u_R + iu_I)$$

$$\Leftrightarrow \dot{\eta} = F(\eta) + G(\eta)\nu.$$  \hspace{1cm} (6)

According to (5), the performance index function (2) can be expressed as

$$J(\eta) = \int_0^\infty r(\eta(\tau), \nu(\tau))d\tau.$$  \hspace{1cm} (7)

For an arbitrary admissible control law $\nu$, if the associated performance index function $J(\eta)$ is given in (6), then an infinitesimal version of (6) is the so-called nonlinear Lyapunov equation [40]

$$0 = J^T(\eta) F(\eta) + G(\eta)\nu + r(\eta, \nu).$$  \hspace{1cm} (8)

where $J_\eta = (\partial J / \partial \eta)$ is the partial derivative of the performance index function $J$. Define the optimal performance index function as

$$J^*(\eta) = \min_{\nu \in U} \int_0^\infty r(\eta(\tau), \nu(\eta(\tau)))d\tau$$  \hspace{1cm} (9)

where $U$ is a set of admissible control laws. Defining the Hamiltonian function as

$$H(\eta, \nu, J_\eta) = J^T(\eta) F(\eta) + G(\eta)\nu + r(\eta, \nu)$$  \hspace{1cm} (10)

the optimal performance index function $J^*(\eta)$ satisfies

$$0 = \min_{\nu \in U} \{H(\eta, \nu, J_\eta)\}. $$  \hspace{1cm} (11)

According to (9) and (10), the optimal control law can be expressed as

$$\nu^*(\eta) = -\frac{1}{2} R^{-1} G^T(\eta) J^*_\eta(\eta).$$  \hspace{1cm} (12)

**Remark 2**: In this brief, the system transformations between (1) and (4) are necessary. We should say that the optimal control cannot be obtained from (1) and (2) directly. For example, if the optimal control is calculated from (1) and (2), according to Bellman optimality principle, we have
where $\varepsilon_H$ is the residual error, which is expressed as $\varepsilon_H = -SV^T(F + Gv)$. Let $\hat{W}_c$ be the estimate of $W_c$, and then the output of the critic network is

$$J(\eta) = \hat{W}_c^T \varphi_c(\eta).$$  \hspace{1cm} (16)

Hence, Hamiltonian function (9) can be approximated by

$$H(\eta, v, \hat{W}_c) = \hat{W}_c^T \nabla \varphi_c(F + Gv) + r(\eta, v).$$  \hspace{1cm} (17)

Then, we can define the weight estimation error of the critic network as

$$\bar{W}_c = W_c - \hat{W}_c.$$  \hspace{1cm} (18)

Note that, for a fixed admissible control law $v$, Hamiltonian function (9) becomes $H(\eta, v, J_\eta) = 0$, which means $H(\eta, v, \bar{W}_c) = 0$. Therefore, from (15), we have

$$\varepsilon_\eta = W_c^T \nabla \varphi_c(F + Gv) + r(\eta, v).$$  \hspace{1cm} (19)

Let $e_c = H(\eta, v, \hat{W}_c) - H(\eta, v, W_c)$. We can obtain

$$e_c = -\bar{W}_c^T \nabla \varphi_c(\eta)(F(\eta) + G(\eta)v) + \varepsilon_\eta.$$  \hspace{1cm} (20)

It is desired to select $\bar{W}_c$ to minimize the squared residual error $E_c = 1/2e_c^T e_c$. Normalized gradient descent algorithm is used to train the critic network [40]. Then, the weight update rule of the critic network $\hat{W}_c$ is derived as

$$\begin{align*}
\dot{\hat{W}}_c &= -\alpha_c \frac{\partial E_c}{\partial \hat{W}_c} = -\alpha_c \frac{\delta_1(\xi_1^T \hat{W}_c + r(\eta, v))}{(\xi_1^T \xi_1 + 1)^2} \\hspace{1cm} (21)
\end{align*}
$$

where $\alpha_c > 0$ is the learning rate of the critic network and $\xi_1 = \nabla \varphi_c(F + Gv)$. It is a modified Levenberg-Marquardt algorithm, that is, $(\xi_1^T \xi_1 + 1)$ is replaced by $(\xi_1^T \xi_1 + 1)^2$, which is used for normalization, and it will be required in the proofs [40]. Let $\hat{\xi}_2 = (\xi_1^T \xi_2)$ and $\bar{\xi}_2 = \bar{\xi}_1^T \bar{\xi}_1 + 1$. We have

$$\begin{align*}
\dot{\hat{W}}_c &= \alpha_c \frac{\delta_1(\xi_1^T \hat{W}_c + r)}{\xi_1^2} \\
&= -\alpha_c \frac{\delta_1^2 \hat{\xi}_2}{\xi_3} \hat{W}_c + \alpha_c \frac{\delta_1^2 \xi_2^T \hat{W}_c + r}{\xi_3} \\
&= -\alpha_c \frac{\delta_1^2 \hat{\xi}_2}{\xi_3} \hat{W}_c + \alpha_c \frac{\delta_1 \xi_2^T W_c + r}{\xi_3} \quad (22)
\end{align*}
$$

### B. Action Network

The action network is used to obtain the control law $u$. The ideal expression of the action network is $u = \tilde{W}_a^T \bar{\varphi}_a(z) + \tilde{\epsilon}_a$, where $\tilde{W}_a \in \mathbb{C}^{n_a \times 1}$ is the ideal weight matrix of the action network. Let $\bar{\varphi}_a(\eta) \in \mathbb{C}^{n_a}$ be the activation function and let $\tilde{e}_a \in \mathbb{C}^n$ be the approximation error of the action network.

Let $\tilde{W}_a = \tilde{W}_a^R + i \tilde{W}_a^I, \bar{\varphi}_a = \bar{\varphi}_a^R + i \bar{\varphi}_a^I$ and $\tilde{\epsilon}_a = \tilde{\epsilon}_a^R + i \tilde{\epsilon}_a^I$. Let

$$\begin{align*}
W_a^T &= \begin{bmatrix}
\tilde{W}_a^T & -\tilde{W}_a^I \\
\tilde{W}_a^I & \tilde{W}_a^R
\end{bmatrix}, \quad \varphi_a = \begin{bmatrix}
\bar{\varphi}_a^R \\
\bar{\varphi}_a^I
\end{bmatrix} \\
\epsilon_a &= \begin{bmatrix}
\tilde{\epsilon}_a^R \\
\tilde{\epsilon}_a^I
\end{bmatrix}
\end{align*}
$$

$$W_a = \begin{bmatrix}
\tilde{W}_a^R & -\tilde{W}_a^I \\
\tilde{W}_a^I & \tilde{W}_a^R
\end{bmatrix}, \quad \varphi_a = \begin{bmatrix}
\bar{\varphi}_a^R \\
\bar{\varphi}_a^I
\end{bmatrix} \quad (23)
$$
We have
\[ v = W^T_a \varphi_a(\eta) + \epsilon_a. \] (23)
The output of the action network is
\[ \hat{v}(\eta) = \hat{W}^T_a \varphi_a(\eta) \] (24)
where \( \hat{W}_a \) is the estimation of \( W_a \). We can define the output error of the action network as
\[ e_a = \hat{W}^T_a \varphi_a + \frac{1}{2} R^{-1} G^T \nabla \varphi_c^T \hat{W}_c. \] (25)
The objective function to be minimized by the action network is defined as
\[ E_a = \frac{1}{2} e_a^T e_a. \] (26)
The weight update law for the action network weight is a gradient descent algorithm, which is given by
\[ \dot{\hat{W}}_a = -a_a \varphi_a \left( \hat{W}^T_a \varphi_a + \frac{1}{2} R^{-1} G^T \nabla \varphi_c^T \hat{W}_c \right)^T \] (27)
where \( a_a \) is the learning rate of the action network. Define the weight estimation error of the action network as
\[ \tilde{W}_a = W_a - \hat{W}_a. \] (28)
Thus, (29) can be rewritten as
\[ \dot{\hat{W}}_a = a_a \varphi_a \left( \hat{W}^T_a \varphi_a - \frac{1}{2} R^{-1} G^T \nabla \varphi_c^T \hat{W}_c \right)^T + W^T_a \varphi_a - \frac{1}{2} R^{-1} G^T \nabla \varphi_c^T \hat{W}_c. \] (29)
As \( v = -(1/2)R^{-1}G^T J_\eta \), according to (14) and (23), we have
\[ W^T_a \varphi_a + e_a = \frac{1}{2} R^{-1} G^T \nabla \varphi_c^T \hat{W}_c - 1/2 R^{-1} G^T \nabla e_c. \] (30)
Thus, (29) can be rewritten as
\[ \dot{\tilde{W}}_a = -a_a \varphi_a \left( \hat{W}^T_a \varphi_a + \frac{1}{2} R^{-1} G^T \nabla \varphi_c^T \hat{W}_c - \epsilon_{12} \right)^T \] (31)
where \( \epsilon_{12} = -e_a - 1/2 R^{-1} G^T \nabla e_c. \)

C. Design of the Compensation Controller

In this subsection, a compensation controller is designed to overcome the approximation errors of the critic and action networks. Before the detailed design method, the following assumption is necessary.

Assumption 3: The approximation errors of the critic and action networks, that is, \( \tilde{c}_c \) and \( \tilde{e}_a \) satisfy \( ||\tilde{c}_c|| \leq \epsilon_{cM} \) and \( ||\tilde{e}_a|| \leq \epsilon_{aM} \). The residual error is upper bounded, that is, \( ||\tilde{e}_M|| \leq \epsilon_{M} \), \( \epsilon_{cM}, \epsilon_{aM}, \) and \( \epsilon_M \) are positive numbers. The vectors of the activation functions of the action network satisfy \( ||\varphi_a|| \leq \varphi_{aM} \), where \( \varphi_{aM} \) is a positive number.

Define the compensation controller as
\[ v_c = -\frac{K_c G^T \eta}{\eta^T \eta + b} \] (32)
where \( K_c \geq ||G||^2 \epsilon_{cM}^2 (\eta^T \eta + b)/2 \eta^T G G^T \eta \), and \( b > 0 \) is a constant. Then, the optimal control law can be expressed as
\[ v_{all} = \tilde{v} + v_c. \] (33)
where \( \tilde{v} \) is the compensation controller, and \( \tilde{v} \) is the output of the action network. Substituting (33) into (4), we can get
\[ \tilde{v} = F + G \hat{W}_a \varphi_a + G v_c, \] (34)
As \( \hat{W}_a \varphi_a = W_a \varphi_a - \hat{W}_a \varphi_a = v - \epsilon_a - \hat{W}_a \varphi_a \) we can obtain
\[ \tilde{v} = F + G v - G e_a - G \hat{W}_a \varphi_a + G v_c. \] (35)
In the next subsection, the stability analysis will be given.

D. Stability Analysis

For continuous-time ADP methods, the signals need to be persistently exciting in the learning process [40], that is, the persistence of excitation assumption.

Assumption 4: Let the signal \( \xi_2 \) be persistently exciting over the interval \([t, t + T]\), that is, there exist constants \( \beta_1 > 0, \beta_2 > 0 \) and \( T > 0 \), such that, for all \( t \)
\[ \beta_1 I \leq \int_t^{t + T} \xi_2(t) \xi_2^T(t) d\tau \leq \beta_2 I \] (36)
holds.

Remark 3: This assumption makes system (4) be persistently excited sufficiently for tuning critic and action networks. The persistent excitation assumption ensures \( \xi_{2m} \leq ||\xi_2|| \), where \( \xi_{2m} \) is a positive number [40].
Before giving the main result, the following preparation works are presented.

Lemma 1: For \( \forall x \in \mathbb{R}^n \), we have
\[ ||x|| \leq \sqrt{m} \] (37)
Proof: Let \( x = (x_1, x_2, \ldots, x_n)^T \). As \( ||x||_2^2 = \sum_{i=1}^n |x_i|^2 \leq \left( \sum_{i=1}^n |x_i|^2 \right) = ||x||_1 \), we can get \( ||x||_2 \leq ||x||_1 \). As \( ||y||_2^2 = \sum_{i=1}^n |y_i|^2 \leq n \sum_{i=1}^n |x_i|^2 = n ||x||_2^2 \), we can obtain \( ||x||_1 \leq \sqrt{m} ||x||_2 \).

Theorem 1: For system (1), if \( f(\cdot) \) satisfies Assumptions 1 and 2, then we have
\[ ||F(\eta) - F(\eta')||_2 \leq k ||(\eta - \eta')||_2 \] (38)
where \( \eta = \begin{bmatrix} x \\ y \end{bmatrix} \) and \( \eta' = \begin{bmatrix} x' \\ y' \end{bmatrix} \). Let \( k^R_j = \max[\lambda_j^R, \lambda_j^L] \) and \( k^L_j = \max[\lambda_j^R, \lambda_j^L] \), \( j = 1, 2, \ldots, n \). Let \( k' = \sum_{j=1}^n k^R_j + \sum_{j=1}^n k^L_j \) and \( k = k' \sqrt{m} \).

Proof: According to Assumption 2 and the mean value theorem for multivariable functions, we have
\[ ||f^R_j(x, y) - f^R_j(x', y')||_1 \leq \lambda_j^R ||x - x'||_1 + \lambda_j^R ||y - y'||_1. \] (39)
According to the definition of one-norm, we have \( ||\eta - \eta'||_1 = ||x - x'||_1 + ||y - y'||_1 \), and
\[ ||f^R_j(x, y) - f^R_j(x', y')||_1 \leq k^R_j ||\eta - \eta'||_1. \] (40)
According to (40), we can get
\[
|| f^R(\textbf{x}, \textbf{y}) - f^R(\textbf{x}', \textbf{y}') ||_1 \leq \sum_{j=1}^{n} \frac{\eta_j}{\sqrt{\lambda_{max}(R)}} || \textbf{y} - \textbf{y}' ||_1.
\] (41)

According to the idea from (40) and (41), we can also obtain
\[
|| f^I(\textbf{x}, \textbf{y}) - f^I(\textbf{x}', \textbf{y}') ||_1 \leq \sum_{j=1}^{n} \frac{\eta_j}{\sqrt{\lambda_{min}(Q)}} || \textbf{y} - \textbf{y}' ||_1.
\] (42)

Therefore, we can get
\[
|| F(\eta) - F(\eta') ||_1 = || f^R(\textbf{x}, \textbf{y}) - f^R(\textbf{x}', \textbf{y}') ||_1 + || f^I(\textbf{x}, \textbf{y}) - f^I(\textbf{x}', \textbf{y}') ||_1 \\
\leq k || \eta - \eta' ||_1.
\] (43)

According to Lemma 1, we have
\[
|| F(\eta) - F(\eta') ||_2 \leq || F(\eta) - F(\eta') ||_1
\] (44)

and
\[
|| \eta - \eta' ||_1 \leq \sqrt{2n} || \eta - \eta' ||_2.
\] (45)

From (43)–(45), we can obtain
\[
|| F(\eta) - F(\eta') ||_2 \leq k || \eta - \eta' ||_2.
\]

The proof is completed.

Next theorems show that the estimation error of critic and action networks are UUB.

Theorem 2: Let the weights of the critic network \( \hat{W}_c \) be updated by (21). If the inequality \( \| \tilde{c}_1^I / \tilde{c}_3 \| \hat{W}_c > \epsilon_{\text{HM}} \) holds, then the estimation error \( \hat{W}_c \) converges to the set \( \hat{W}_c \leq \beta_1^{-1} \sqrt{\beta_2 T} (1 + 2 \rho \beta_2 \alpha_c) \epsilon_{\text{HM}} \) exponentially, where \( \rho > 0 \).

Proof: Let \( \hat{W}_c \) be defined as in (18). Define the following Lyapunov function candidate:
\[
L = \frac{1}{2 \alpha_c} \hat{W}_c^T \hat{W}_c.
\] (46)

The derivative of (46) is given by
\[
\dot{L} = \hat{W}_c^T \left( - \tilde{c}_1^I / \tilde{c}_3 \hat{W}_c + \tilde{c}_1^I \epsilon_H \right) \\
\geq - \left| \frac{\tilde{c}_1^I}{\tilde{c}_3} \hat{W}_c \right| \left( \left| \frac{\tilde{c}_1^I}{\tilde{c}_3} \hat{W}_c \right| - \left| \frac{\epsilon_H}{\tilde{c}_3} \right| \right).
\] (47)

As \( \tilde{c}_3 \geq 1 \), we have \( \| \epsilon_H / \tilde{c}_3^I \| < \epsilon_{\text{HM}} \). If \( \| \tilde{c}_1^I / \tilde{c}_3 \| \hat{W}_c > \epsilon_{\text{HM}} \), then we can get \( \dot{L} \leq 0 \). That means \( L \) decreases and \( \| \tilde{c}_3^I \hat{W}_c \| \) is bounded. In light of [42] and technical Lemma 2 in [40], \( \hat{W}_c \leq \beta_1^{-1} \sqrt{\beta_2 T} (1 + 2 \rho \beta_2 \alpha_c) \epsilon_{\text{HM}} \).

Theorem 3: Let the optimal control law be expressed as in (33). The weight update laws of the critic and action networks are given as in (21) and (27), respectively. If there exists parameters \( l_2 \) and \( l_3 \), that satisfy
\[
l_2 > || G ||
\] (48)

and
\[
l_3 > \max \left\{ || G ||^2 / \lambda_{\text{min}}(R), \frac{2k + 3}{\lambda_{\text{min}}(Q)} \right\}
\] (49)

respectively, then the system state \( \eta \) in (4) is UUB and the weights of the critic and action networks, that is, \( \hat{W}_c \) and \( \hat{W}_a \) converge to finite neighborhoods of the optimal ones.

Proof: The proof can be seen in Appendix A.

IV. SIMULATION STUDY

Example 1: Our first example is chosen in ([3], Example 3) with modifications. Consider the following nonlinear complex-valued harmonic oscillator system:
\[
\dot{z} = i z - \frac{2z(z^2 - \frac{5}{2})}{2z^2 - 1} + (1 + i)u
\] (52)

where \( z \in \mathbb{C} \), \( z = z_r + iz^I \), and \( u = u_r + iu^I \). The utility function is defined as \( \tilde{r} (z, u, \tilde{z}) = z_r Q_r z_r + u_r H R_r u_r \). Let \( Q_r = E \) and \( R_r = E \), where \( E \) is the identity matrix with a suitable dimension. Let \( \eta = [z_r, z^I]^T \) and \( \nu = [u_r, u^I]^T \). Let the critic and action networks be expressed as \( \hat{J}(\eta) = \hat{W}_c(\eta) \) and \( \hat{v}(\eta) = \hat{W}_a(\eta) \), where \( Y_c \) and \( Y_a \) are constant matrices with suitable dimensions. The activation functions of the critic and action networks are hyperbolic tangent functions [20]. The structures of the critic and action networks are 2-8-1 and 2-8-2, respectively. The initial weights of \( \hat{W}_c \) and \( \hat{W}_a \) are selected arbitrarily from \((-0.1, 0.1)\), respectively. The learning rates of the critic and action networks are selected as \( \alpha_c = \alpha_a = 0.01 \). Let \( z_0 = -1 + i \). Let \( K_r = 3 \) and \( b = 1.02 \). Implementing the developed ADP method for 40 time steps, the trajectories of the control and state are shown in Figs. 1 and 2, respectively. The weights of the critic and action networks converge to \( \hat{W}_c = [-0.0904; 0.0989; -0.0586; 0.0214; -0.0304; 0.0435; -0.0943; -0.0866] \) and \( \hat{W}_a = [-0.0269; 0.0117; 0.0197; -0.0548; 0.0336; -0.0790; 0.0789; -0.0980; -0.0825; -0.0881; 0.0078; -0.0354; \)
The developed ADP method will be justified by a complex-valued system, where we can see that the system state is UUB, which verifies the effectiveness of the developed optimal control scheme.

Finally, the simulation examples are given to show the effectiveness of the developed control method for complex-valued systems. First, the performance index function is defined based on complex-valued state and control. Then, system transformations are used to overcome Cauchy–Riemann conditions. With the transformed system and the corresponding performance index function, a new ADP-based optimal control method is established. A compensation controller is presented to compensate the approximation errors of neural networks. Finally, the simulation examples are given to show the effectiveness of the developed optimal control scheme.

APPENDIX A

PROOF OF THEOREM 3

Proof: Choose the Lyapunov function candidate as
\[ V = V_1 + V_2 + V_3 \]
where \( V_1 = 1/2\alpha_i \tilde{W}_c^T \tilde{W}_c \), \( V_2 = l_2/2\alpha_i \tr(\tilde{W}_a^T \tilde{W}_a) \), and \( V_3 = \eta^T \eta + l_3 J(\eta) \) with \( l_2 > 0, l_3 > 0 \).

Taking the derivative of the Lyapunov function candidate (54), we can get
\[ \dot{V} = \dot{V}_1 + \dot{V}_2 + \dot{V}_3. \]
According to (22), we have
\[ \dot{V}_1 = -((\tilde{W}_c^T \xi_2)^T \tilde{W}_c \xi_2 + (\tilde{W}_c^T \xi_2)^T \tilde{W}_c \xi_2). \]

The derivative of \( V_3 \) can be expressed as
\[ \dot{V}_3 = (2\eta^T F - 2\eta^T G \tilde{W}_a^T \phi_a + 2\eta^T G v - 2\eta^T G \tilde{W}_a^T \phi_a + 2\eta^T G v - 2\eta^T G \tilde{W}_a^T \phi_a). \]

From Theorem 1, we have \( 2\eta^T F \leq 2k \| \eta \|^2 \). In addition, we can obtain
\[ -2\eta^T G \tilde{W}_a^T \phi_a \leq \| \eta \|^2 + \| G \|^2 (\tilde{W}_a^T \phi_a)^T \tilde{W}_a^T \phi_a \]
\[ 2\eta^T G v \leq \| \eta \|^2 + \| G \|^2 \| v \|^2 \]
\[ -2\eta^T G \tilde{W}_a^T \phi_a \leq \| \eta \|^2 + \| G \|^2 \| v \|^2 \]
From (32), we can get
\[ 2\eta^T G v = 2\eta^T G \left( \frac{-\tilde{K}_c \tilde{G}_a}{\eta^T \eta + b} \right) \leq -\| G \|^2 \tilde{v}_{am}. \]

Then, (57) can be rewritten as
\[ \dot{V}_3 \leq (2k + 3 - l_3 \lambda_{min}(Q))\| \eta \|^2 + (\| G \|^2 - l_3 \lambda_{min}(R))\| v \|^2 \]
Let \( Z = [\eta^T, v^T, (\tilde{W}_c^T \xi_2)^T, (\tilde{W}_c^T \xi_2)^T]^T \), and \( N_V = [0, 0, (\epsilon_H/\xi_3), M_V]^T \), where \( M_V = -(l_2/2)R^{-1}G^T \)}
\[ \nabla_\mathcal{G}^2 \hat{W}_c + I_{2 \times 12}, \quad \text{and} \quad M_V = \text{diag} \left( (l_{3 \lambda_{\min}}(Q) - 2k - 3), \quad (l_{3 \lambda_{\min}}(R) - \|G\|_F^2), \quad 1, \quad l_2 - \|G\|_F^2 \right). \]
Thus, we have
\[ \hat{V} \leq -||\mathbf{Z}||^2 l_{3 \lambda_{\min}}(M_V) + ||\mathbf{Z}|| ||N_V||. \]

According to (48) and (49), we can see that if \( ||\mathbf{Z}|| \geq ||N_V|| / l_{3 \lambda_{\min}}(M_V) \equiv \mathbf{Z}_B \), then the Lyapunov candidate \( \hat{V} \leq 0 \).

As \( M_{V4} \) and \( \varepsilon_H \lambda_3 \) are both upper bounded, we have \( ||N_V|| \) is upper bounded. Therefore, the state \( \eta \), the weight errors \( \hat{W}_c \) and \( \hat{W}_d \) are UUB [44]. The proof is completed.

**REFERENCES**


