IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 13, 2021, accepted May 2, 2021, date of publication May 6, 2021, date of current version May 17, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3078122

DNN-Based H_∞ Control Scheme of Nonlinear Time-Varying Dynamic Systems With External Disturbance and Its Application to UAV Tracking Design

BOR-SEN CHEN[1,2], (Life Fellow, IEEE), MIN-YEN LEE[1], AND TZU-HAN LIN[1]
[1]Department of Electrical Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan
[2]Department of Electrical Engineering, Yuan Ze University, Taoyuan City 32003, Taiwan

Corresponding author: Bor-Sen Chen (bschen@ee.nthu.edu.tw)

This work was supported by the Ministry of Science and Technology of Taiwan under Grant MOST 108-2221-E-007-099-MY3.

ABSTRACT The main difficulty in the traditional nonlinear H_∞ control design lies in how to solve the nonlinear partial differential Hamilton-Jacobi-Isaacs equation (HJIE), especially for nonlinear time-varying systems. In this study, a novel HJIE-embedded DNN H_∞ control scheme is proposed to be efficiently trained for nonlinear H_∞ stabilization and tracking control designs of nonlinear dynamic systems with the external disturbance. The proposed DNN-based H_∞ control approach not only capitalizes on the availability of theoretical partial differential HJIE but also reduces the amount of empirical data and the complexity to train HJIE-embedded DNN. We have shown that the proposed DNN-based H_∞ control scheme can approach the theoretical result of H_∞ robust control when the training error approaches zero and the asymptotic stability is also guaranteed if the nonlinear time-varying system is free of external disturbance. The proposed method could be easily extended to DNN-based H_∞ reference tracking control of nonlinear systems for more practical applications. Finally, two examples, including (*i*) an H_∞ stabilization of nonlinear time-varying system and (*ii*) an H_∞ unmanned aerial vehicle (UAV) reference tracking control system, are proposed to illustrate the design procedure and to demonstrate the effectiveness of our DNN-based H_∞ method.

INDEX TERMS Nonlinear time-varying dynamic system, nonlinear H_∞ stabilization and tracking control, Hamilton Jacobi Issac equation (HJIE), DNN-based H_∞ control design, unmanned aerial vehicle (UAV) tracking control.

I. INTRODUCTION

Deep neural network (DNN) is an information processing model inspired by biological neural systems and enables us to perform tasks by learning from big data for a large variety of applications. In the last few years, powerful big data-driven methods based on DNN have been exceptionally applied to speech recognition [1], [2], image classification [3], [4], translation of languages [5], etc. These kinds of works are usually simple to perform by human beings but are still very difficult to execute by machines. In order to fit input/output data pairs of the above applications, we need to train the best architecture of an DNN as possible [6], [7]. The current training methodology of deep learning methods is to employ a big data-driven approach. Once DNN has been trained, it can respond to never-observed input data with the optimal output according to past trained knowledge [8].

Recently, deep learning methods have made great progress in several application domains successfully due to the development of hardware. In general, the traditional deep learning methodology employs big data-driven approach, i.e. it acquires a very large amount of empirical data about system behavior for training to achieve the specific performance optimization [9]. However, opposite to image classification and speech recognition, system dynamic models and theoretical optimal results have been well developed over several decades of intense research and very often available in the system control field [10], [11]. Consequently, the application of deep learning to system control designs and optimization problems offers more possibilities than a pure big data-driven

The associate editor coordinating the review of this manuscript and approving it for publication was Nasim Ullah.

approach in the traditional deep learning methodology. It is believed that these can be regarded as priori expert knowledge that should not be missed and ignored in the deep learning approaches. Therefore, this study puts forth a new DNN-based H_∞ control design to utilize the availability of system models and theoretical minimax Nash game results of H_∞ stabilization design problem. This will reduce the amount of training data and the complexity of training DNN for the robust H_∞ control, especially for external disturbance data which are always unavailable and unpredictable for training DNN in real system applications.

In general, due to the system uncertainties in modeling, there exists time-varying parameters in the physical system [12], [13]. For example, the value of resistance in motor system, which depends on environmental temperature, will varies within a range during the control process [14]. Moreover, since the physical systems are always nonlinear with external disturbance, nonlinear H_∞ control of nonlinear time-varying systems has been a major research in the robust control field for decades [15], [16]. In recent decades, researchers have come up with ideas that boast more constructive methodologies on how to recover and design robust H_∞ controller, which are particularly stimulated by a variety of applications, especially the flight and attitude control in outerspace [17], [18]. Generally, nonlinear H_∞ control is proposed to minimize the worst-case effect of external disturbance on system performance. Based on the nonlinear dynamic system, the optimal nonlinear H_∞ control design needs to solve a nonlinear partial differential Hamilton-Jacobi-Isaacs equation (HJIE) for nonlinear H_∞ control law [19]. In general, there exists no efficient method to solve the nonlinear partial differential HJIE analytically or numerically, especially in the case of H_∞ tracking control design for nonlinear time-varying systems, and may not have global analytic solutions even in simple cases. To deal with this problem, in previous studies, the nonlinear time-invariant system is interpolated by a set of local linearized systems through fuzzy interpolation method [20], [21], global linearization method [22], [23], gain scheduling method [24], [25], etc. Then the nonlinear partial differential HJIE can be replaced by a set of Riccati-like inequalities while the Lyapunov function in the HJIE is selected as a specific quadratic function, i.e., $V(x(t)) = x^T(t)Px(t)$ for some positive symmetric matrix P. By utilizing the technique of Schur complement, the set of Riccati-like inequalities can be transformed to a set of linear matrix inequalities (LMIs), which can be easily solved with the help of LMIs TOOLBOX in MATLAB [26].

Recently, in order to avoid solving HJIEin finite-time nonliner H_∞ robust control, a finite-time $\mathcal{L}_2$ gain synchronous control for continuous-time positive hidden Markov jump system was introduced via T-S Fuzzy model approach [35]. A finite-time stabilization was introduced for positive Markovian jumping neural network in [36]. A finite-time positiveness and distributed control was introduced for Lipschitz nonlinear multi-agent system in [37]. Further, radial basis function (RBF) neural network has been utilized for nonlinear system control design. For example, a RBF neural network-based adaptive robust control was proposed for nonlinear bilateral teleoperation manipulators with uncertainties and time-varying dealy in [38]. A radial basis function neural network (RBFNN) -based adaptive sliding mode control design was introduced for delayed nonlinear multilayer robotic system with cooperative manipulation in [39]. Fuzzy output feedback controls in [40]–[42] were also employed to treat robust control of nonlinear dynamic system. The related works about the application of robust RBFNN can be refered to [43], [44].

However, even the above approximation methods can efficiently solve HJIE for H_∞ control of nonlinear time-varying systems, there is some approximation error between the nonlinear systems and a set of interpolated local linear systems. Furthermore, the conventional nonlinear Lyapunov function $V(x(t))$ is always limited to a quadratic function $V(x(t)) = x^T(t)Px(t)$, which may lead to a conservative result of nonlinear H_∞ control of nonlinear time-invariant systems [20]–[27]. Moreover, in order to decrease the T-S fuzzy approximation error, we need to increase the number of T-S fuzzy rules, which will significantly increase the number of Riccati-like equations to be solved [24]–[27]. This will lead to the inconvenience and complexity in the large scale systems or interconnected systems nowadays. For example, in the fuzzy H_∞ reference tracking control of quadrotor UAV system of example 2 in the sequel, due to highly nonlinear dynamic system of quadrotor, there are a large number of Riccati-like equations to be solved for fuzzy control gains in the fuzzy control systems, and it needs to compute fuzzy control at every time instant [27]. It is a very complicate process and more efforts are still needed in the practical H_∞ fuzzy tracking control application to quadrotor UAV.

Even H_∞ robust control designs of nonlinear time-invariant system have been widely studied in the past decades. At present, few H_∞ robust control designs of nonlinear time-varying system have been discussed because we need to solve a more difficult partial differential equation of time-varying HJIE than the nonlinear time-invariant system [42], which cannot be easily solve analytically or numerically. More precisely, the difficulties to the research of DNN H_∞ control scheme for time varying nonlinear dynamic system are (i) how to solve $\frac{\partial V(x(t),t)}{\partial x(t)}$ and $\frac{\partial V(x(t),t)}{\partial t}$ from HJIE in (4) and (ii) how to prove the HJIE-embedded H_∞ control scheme in Fig. 1 to approach the theoretical H_∞ control strategy in the time-varying nonlinear dynamic system as training error $\epsilon(t)$ approaches to zero.

In this study, based on nonlinear dynamic model, an HJIE-embedded DNN is constructed to be trained for solving $\frac{\partial V(x(t),t)}{\partial t}$ and $\frac{\partial V(x(t),t)}{\partial x(t)}$, i.e. $\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)} = [(\frac{\partial V(x(t),t)}{\partial x(t)})^T \ (\frac{\partial V(x(t),t)}{\partial t})^T]^T$, of the nonlinear partial differential HJIE for nonlinear H_∞ controller of nonlinear

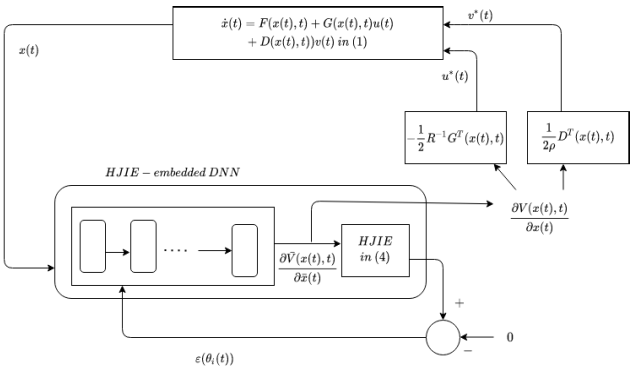

FIGURE 1. Flow chart of HJIE-embedded DNN-based H_∞ control scheme of nonlinear time-varying system in (1). In the training phase, $x(t)$ is generated by nonlinear dynamic model with H_∞ control $u^*(t)$ and the worst-case external disturbance $v^*(t)$ through the output of the HJIE-embedded DNN. In the operation phase, $x(t)$ is the real state of nonlinear system with H_∞ control input $u^*(t)$ and real external disturbance $v(t)$.

time-varying systems with external disturbance. In the training process as shown in Fig. 1, we input system state $x(t)$ into HJIE-embedded DNN to obtain output $\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)}$, which must satisfy with the HJIE. The error computed by the embedded HJIE is fed back to train the weightings of DNN to approach $\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)}$ of HJIE, which is the functional of $\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)}$. We can prove that when the error of embedded HJIE approaches to 0, the HJIE-embedded H_∞ control scheme in Fig. 1 will approach the H_∞ robust stabilization control of nonlinear time-varying system with external disturbance. Further, if external disturbance disappears, the asymptotical stability of nonlinear time-varying system is also guaranteed.

The HJIE can be considered as a priori expert knowledge of H_∞ control of nonlinear time-varying system to train DNN. The proposed HJIE-embedded DNN H_∞ robust control design can significantly reduce the amount of training data and training time when compared with the traditional deep learning approaches based on big data. The reason is that we can train the DNN based on the state $x(t)$ generated by the nonlinear dynamic system model through the H_∞ control input $u^*(t)$ and the worst-case external disturbance $v^*(t)$, which can be generated via the DNN output to replace the unavailable $v(t)$ from the H_∞ disturbance rejection perspective in the training process as shown in Fig. 1.

The main contributions of this article are described as follows:

(I) An HJIE-embedded DNN H_∞ control design is proposed to integrate the available theoretical H_∞ control results with DNN learning schemes to efficiently solve robust H_∞ control design problems for nonlinear systems, especially nonlinear time-varying systems. The training process can be accomplished by Adam learning algorithm [29] to adjust DNN parameters to satisfy HJIE to achieve the robust H_∞ control performance of nonlinear time-varying system.

(II) Unlike the conventional methods to solve $V(x(t), t)$ from HJIE directly, an HJIE-embedded DNN is constructed to approach $\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)}$ of HJIE of the H_∞ robust control strategy of nonlinear time-varying dynamic system with external disturbance. The DNN-based control scheme can be shown to approach the H_∞ robust control of nonlinear time-varying system by Adam learning algorithm and can be easily extended to the design of H_∞ reference tracking control on nonlinear dynamic systems with external disturbance.

(III) By the proposed HJIE-embedded DNN approach, the priori expert knowledge of nonlinear dynamic models and theoretical robust H_∞ control results, which have been acquired over decades, can complement the traditional pure big data-driven deep learning approaches to facilitate the use of deep learning schemes for application to robust H_∞ control system designs of complex nonlinear time-varying systems, which cannot be easily solved by conventional methods. Further, since the H_∞ control $u^*(t)$ and the worst-case external disturbance $v^*(t)$ are used to generate $x(t)$ by the nonlinear system model to train HJIE-embedded DNN in the training phase, we can save a large amount of training data and training time than the conventional data-driven DNN control methods.

The remainder of this study is organized as follows. In Section II, we discuss the concept of H_∞ control. In Section III, a novel model of DNN-based H_∞ control design is introduced to deal with time-varying nonlinear systems. In Section IV, a numerical example and a UAV example are provided to demonstrate the performance of the proposed method. The conclusion is made in Section V.

Notations: A^T: the transpose of matrix A; $A = A^T \geq 0$: symmetric positive semi-definite matrix A; I_n: n-by-n identity matrix; $0_{n\times m}$: n-by-m zero matrix; $\mathbb{R}^n$: the set of n-tuple real vectors; $\mathbb{R}^{n\times m}$: the set of all real $n \times m$ matrices; $L_2[0, \infty)$: a set of real n-tuple functions with finite energy, i.e., for $\forall w(t) \in L_2[0, \infty)$, $\|w(t)\|_2 = (\int_0^\infty w^T(t)w(t)dt)^{\frac{1}{2}} < \infty$.

II. PROBLEM FORMULATION

Robust H_∞ control design has been a major research topic for decades and it encompasses a broad spectrum of areas and impacts. In general, external disturbances are always unavoidable in real control systems, such as the loadings and environmental interference. In the past decades, robust H_∞ control strategies have been developed to efficiently attenuate the effect of external disturbance on the nonlinear quadratic stabilization performance from the worst-case perspective. In this section, we review the concept of robust H_∞ control of nonlinear dynamic systems with external disturbance.

Consider the following nonlinear time-varying system with external disturbance:

$$\begin{aligned} \dot{x}(t) &= F(x(t), t) + G(x(t), t)\, u(t) + D(x(t), t)\, \upsilon(t) \\ x(0) &= x_0 \end{aligned} \tag{1}$$

where $x(t) \in \mathbb{R}^n$ denotes the state vector, $x(0) \in \mathbb{R}^n$ denotes the initial condition, $u(t) \in \mathbb{R}^m$ denotes the input vector, $F(x(t), t) \in \mathbb{R}^n$ and $G(x(t), t) \in \mathbb{R}^{n\times m}$ denote system functions. $\upsilon(t) \in \mathbb{R}^k$ denotes external disturbance and

$D(x(t), t) \in \mathbb{R}^{n\times k}$ denotes the coupling matrix from external disturbance to the nonlinear system.

The following robust min max H_∞ control strategy is employed to efficiently attenuate the external disturbance on the quadratic stabilization [11]–[16]:

$$\min_{u(t)} \max_{\upsilon(t)\in L_2[0,\infty)} \frac{\int_0^\infty (x^T(t)Qx(t)+u^T(t)Ru(t))dt - V(x(0),0)}{\int_0^\infty \upsilon^T(t)\upsilon(t)dt} \le \rho \quad (2)$$

where $Q = Q^T \ge 0$ and $R = R^T > 0$ are the weighting matrices of state variables and control input, respectively, and $\rho > 0$ is the prescribed disturbance attenuation level. $V(x(t), t) > 0$ is the Lyapunov function of nonlinear time-varying system in (1). $V(x(0), 0)$ in the numerator of (2) is to extract the effect of initial condition $x(0)$. The physical meaning of robust H_∞ control strategy in (2) is that the worst-case effect of external disturbance $v(t)$ on the quadratic control performance must be minimized by control $u(t)$ and should be also less than or equal to a prescribed disturbance attenuation level $\rho > 0$ from the energy perspective.

Remark 1: The choice of weighting parameter Q and R of control strategy in (2) is a trade-off between the stabilization performance and control effort. If the designer wants $x(t)$ to approach to 0 quickly without the consideration of control effort, then Q is selected large than R. If the designer wants to save control effort and does not care about the convergence rate of $x(t)$, then R is selected larger than Q. if $x_i(t)$ (i.e., ith state of $x(t)$) is more significant than $x_j(t)$ (i.e., jth state of $x(t)$), then the diagonal term in Q_{ii} is larger than Q_{jj}. Similarly, if $u_i(t)$ (i.e., ith component of $u(t)$) is with a more cost than $u_j(t)$ (i.e., jth component of $u(t)$), R_{ii} is larger than R_{jj}.

Theorem 1: (a) The min max H_∞ *control strategy in (2) for the nonlinear time-varying system in (1) can be solved by the following H_∞ control $u^*(t)$ and the worst-case external disturbance $v^*(t)$:*

$$u^*(t) = -\frac{1}{2}R^{-1}G^T(x,t)\left(\frac{\partial V(x(t),t)}{\partial x(t)}\right)$$
$$\upsilon^*(t) = \frac{1}{2\rho}D^T(x,t)\left(\frac{\partial V(x(t),t)}{\partial x(t)}\right) \quad (3)$$

where $V(x(t), t) > 0$, $V(0, t) = 0$ is the solution of the following HJIE:

$$\begin{aligned} &HJIE \\ &= \left(\frac{\partial V(x(t),t)}{\partial t}\right) + \left(\frac{\partial V(x(t),t)}{\partial x}\right)^T F(x(t),t) + x^T(t)Qx(t) \\ &\quad - \frac{1}{4}\left(\frac{\partial V(x(t),t)}{\partial x(t)}\right)^T G(x(t),t)R^{-1}G^T(x(t),t)\left(\frac{\partial V(x(t),t)}{\partial x(t)}\right) \\ &\quad + \frac{1}{4\rho}\left(\frac{\partial V(x(t),t)}{\partial x(t)}\right)^T D(x(t),t)D^T(x(t),t)\left(\frac{\partial V(x(t),t)}{\partial x(t)}\right) \\ &= 0 \end{aligned} \quad (4)$$

(b) If the nonlinear time-varying system in (1) is free of external disturbance $v(t)$, i.e., $v(t) = 0$, then the H_∞ control strategy in (3) will achieve asymptotic stability, i.e., $x(t) \to 0$ as $t \to \infty$.

Proof: See Appendix A. □

HJIE in (4) is a partial time-varying nonlinear differential equation. In general, it is still very difficult to solve HJIE in (4) for H_∞ control of (1). The conventional methods (e.g., [42]) always solve $V(x(t), t)$ from HJIE in (4) directly. If we employ DNN to solve $V(x(t), t)$ directly in Fig. 1, we still need to calculate $\frac{\partial V^T(x(t),t)}{\partial x(t)}$ for $u^*(t)$ in (4), which is not easy to perform in the DNN-based H_∞ control in Fig. 1 because $V(x(t), t)$ and $x(t)$ are all numerical data. In this study, unlike the conventional methods to solve $V(x(t), t)$ of HJIE in (4), we want to solve $(\frac{\partial V(x(t),t)}{\partial x(t)})$ and $(\frac{\partial V(x(t),t)}{\partial t})$, i.e., $\frac{\partial V(\tilde{x}(t),t)}{\partial \tilde{x}(t)} = [(\frac{\partial V^T(x(t),t)}{\partial x(t)})\ (\frac{\partial V^T(x(t),t)}{\partial t})]^T$, from (4) by the proposed HJIE-embedded DNN scheme in Fig.1 for $u^*(t)$ and $v^*(t)$ in (3) in the off-line training phase. $u^*(t)$ and $v^*(t)$ are feedback to nonlinear model in (1) to replace $v(t)$ and $u(t)$ to generate $x(t)$ to train DNN by Adam learning algorithm to approach $\frac{\partial V(\tilde{x}(t),t)}{\partial \tilde{x}(t)}$ to solve HJIE $= 0$ in (4) to achieve H_∞ robust control. More details will be discussed in the sequel.

In some specific systems, the coefficients of system functions $F(x(t), t)$ and $G(x(t), t)$ are time-invariant as the following nonlinear time-invariant system:

$$\begin{aligned} \dot{x}(t) &= F(x(t)) + G(x(t))u(t) + D(x(t))\upsilon(t), \\ x(0) &= x_0 \end{aligned} \quad (5)$$

Then the above theorem is modified as follows.

Corollary 1 ([11], [19]): The robust H_∞ control strategy in (2) for the nonlinear time-invariant system in (5) can be solved by the following H_∞ control $u^*(t)$ and worst-case external disturbance $v^*(t)$:

$$u^*(t) = -\frac{1}{2}R^{-1}G^T(x(t))\left(\frac{\partial V(x(t))}{\partial x(t)}\right)$$
$$\upsilon^*(t) = \frac{1}{2\rho}D^T(x(t))\left(\frac{\partial V(x(t))}{\partial x(t)}\right) \quad (6)$$

where $V(x(t)) > 0$ and $V(0) = 0$ is the solution of the following time-invariant HJIE:

$$\begin{aligned} &x^T(t)Qx(t) + \left(\frac{\partial V(x(t))}{\partial x(t)}\right)^T F(x(t)) \\ &\quad - \frac{1}{4}\left(\frac{\partial V(x(t))}{\partial x(t)}\right)^T G(x(t))R^{-1}G^T(x(t))\left(\frac{\partial V(x(t))}{\partial x(t)}\right) \\ &\quad + \frac{1}{4\rho}\left(\frac{\partial V(x(t))}{\partial x(t)}\right)^T D(x)D^T(x)\left(\frac{\partial V(x(t))}{\partial x(t)}\right) = 0 \end{aligned} \quad (7)$$

Proof : Similar to Appendix A.

The physical meaning of Corollary 1 is that in the nonlinear time-invariant system of (5) the H_∞ control $u^*(t)$ in (6) will achieve the minmax H_∞ control performance with a prescribed attenuation level ρ in (2) even under the worst-case effect of $v^*(t)$ in (6) among all possible $v(t) \in L_2([0, \infty])$. Since the nonlinear system in (5) is time-invariant, HJIE in (7) is free of $\frac{\partial V(x(t),t)}{\partial t}$ as the HJIE in (4) due to the time-varying characteristic in (1). Since finite-time H_∞ control is an interesting topic in practical application [35]–[37], if we impose an additional flexibility on the terminal time in the H_∞ quadratic performance index in (2), the finite-time

min max H_∞ control strategy is given as follows:

$$\min_{u(t)} \max_{\upsilon(t)\in L_2[0,t_f)} \frac{V(x(t_f),t_f)+\int_0^{t_f}\left(x^T(t)Qx(t)+u^T(t)Ru(t)\right)dt-V(x(0),0)}{\int_0^{t_f}\upsilon^T(t)\upsilon(t)dt} \leq \rho \tag{8}$$

where $V(x(t), t_f)$ is the penalty on the terminal state $x(t_f)$. And then we get the following result.

Remark 2: At present, some published works have focused on solving the H_∞ robust control of nonlinear time-invariant systems in (5). Since the HJII in (7) is still very difficult to solve directly, many methods have been developed. Some published works do not solve HJIE directly. They used the global linearization method [22], [23], gain scheduling method [24], [24] or fuzzy interpolation method [20], [21] by combining several local linearized systems to approximate nonlinear time-invariant system in (5). Then, the HJIE will become a set of Riccati-like equations, which could be transformed to a set of LMIs and it can be easily solved by LMI TOOLBOX in MATLAB. However, in some highly nonlinear systems like quadrotor UAV, a large number of local linear systems are needed to approach nonlinear time-invariant system. Thus, much effort is needed in the design procedure and it takes more computation time to calculate control signal $u(t)$.

Corollary 2 ([11], [19]): The min max *H_∞ control strategy in (8) for the time-varying nonlinear system with external disturbance in (1) can be solved by:*

$$u^*(t) = -\frac{1}{2}R^{-1}G^T(x,t)\left(\frac{\partial V(x,t)}{\partial x(t)}\right)$$
$$\upsilon^*(t) = \frac{1}{2\rho}D^T(x,t)\left(\frac{\partial V(x,t)}{\partial x(t)}\right) \tag{9}$$

where $V(x(t), t) > 0$ with $V(0, t) = 0$ is the solution of the following time-varying HJIE:

$$\left(\frac{\partial V(x(t),t)}{\partial t}\right)+\left(\frac{\partial V(x(t),t)}{\partial x(t)}\right)^T F(x(t),t)+x^T(t)Qx(t) - \frac{1}{4}\left(\frac{\partial V(x(t),t)}{\partial x(t)}\right)^T G(x(t),t)R^{-1}G^T(x(t),t) \times \left(\frac{\partial V(x(t),t)}{\partial x(t)}\right)+\frac{1}{4\rho}\left(\frac{\partial V(x(t),t)}{\partial x(t)}\right)^T D(x(t),t) \times D^T(x(t),t)\left(\frac{\partial V(x(t),t)}{\partial x(t)}\right)=0 \tag{10}$$

Proof : Similar to Appendix A.

Corollary 3 ([11], [19]): The finite-time min max *H_∞ control strategy in (8) for the nonlinear time-invariant system in (5) is given by*

$$u^*(t) = -\frac{1}{2}R^{-1}G^T(x(t))\left(\frac{\partial V(x(t))}{\partial x(t)}\right)$$
$$\upsilon^*(t) = \frac{1}{2\rho}D^T(x(t))\left(\frac{\partial V(x(t))}{\partial x(t)}\right) \tag{11}$$

where $V(x(t)) \geq 0$ with $V(0) = 0$ is the solution of the following HJIE:

$$\left(\frac{\partial V(x(t))}{\partial x(t)}\right)^T F(x(t))+x^T(t)Qx(t) - \frac{1}{4}\left(\frac{\partial V(x(t))}{\partial x(t)}\right)^T G(x(t))R^{-1}G^T(x(t))\left(\frac{\partial V(x(t))}{\partial x(t)}\right) + \frac{1}{4\rho}\left(\frac{\partial V(x(t))}{\partial x(t)}\right)^T D(x(t))D^T(x(t),t)\left(\frac{\partial V(x(t))}{\partial x(t)}\right)=0 \tag{12}$$

Proof : Similar to Appendix A.

For the finite-time minmax H_∞ control strategy in (11) of nonlinear time-invariant system in (5), if we could solve $\frac{\partial V(x(t))}{\partial x(t)}$ from HJIE in (12), then we could obtain the H_∞ robust control $u^*(t) = -\frac{1}{2}R^{-1}G^T(x(t),t)(\frac{\partial V(x(t),t)}{\partial x(t)})$ and the worst-case external disturbance $v^*(t) = \frac{1}{2\rho}D^T(x(t),t)(\frac{\partial V(x(t),t)}{\partial x(t)})$ to achieve the finite-time minmax H_∞ control strategy.

Suppose we want to design H_∞ reference tracking control for nonlinear system in (5) to track a desired reference $r(t)$. Then the H_∞ reference tracking performance is given as (13), shown at the bottom of the page. Let us denote the tracking error as follows.

$$e(t) = x(t) - r(t) \tag{14}$$

Then we have

$$\begin{aligned}\dot{e}(t) &= \dot{x}(t) - \dot{r}(t)\\ &= F(x(t)) + G(x(t))u(t) + D(x(t))v(t) - \dot{r}(t)\\ &= F(e(t)+r(t)) - \dot{r}(t) + G(e(t)+r(t))u(t)\\ &\quad + D(e(t)+r(t))v(t)\end{aligned} \tag{15}$$

which can be considered as nonlinear time-varying dynamic system due to explicit time function $r(t)$ and $\dot{r}(t)$ in the above tracking error system. Let us denote

$$F_e(e(t),t) = F(e(t)+r(t)) - \dot{r}(t)$$
$$G_e(e(t),t) = G(e(t)+r(t))$$
$$D_e(e(t),t) = D(e(t)+r(t))$$

Then the tracking error dynamic system in (15) can be arranged as the following nonlinear time-varying system.

$$\dot{e}(t) = F_e(e(t),t) + G_e(e(t),t)u(t) + D_e(e(t),t)v(t) \tag{16}$$

$$\min_{u(t)} \max_{\upsilon(t)\in L_2[0,\infty)} \frac{\int_0^\infty [(x(t)-r(t))^T Q(x(t)-r(t))+u^T(t)Ru(t)]dt - V(x(0)-r(0),0)}{\int_0^\infty v^T(t)v(t)dt} \leq \rho \tag{13}$$

and the H_∞ reference tracking control problem in (13) is modified as the following H_∞ stabilization of the tracking error system in (16):

$$\min_{u(t)} \max_{v(t)\in L_2[0,\infty)} \frac{\int_0^\infty [e^T(t)Qe(t)+u^T(t)Ru(t)]dt - V(e(0),0)}{\int_0^\infty v^T(t)v(t)dt} \le \rho \tag{17}$$

In this situation, the H_∞ tracking control in (13) of nonlinear time-invariant system in (5) can be transformed to the H_∞ stabilization control problem in (17) of nonlinear error dynamic system in (16). Then we get the following result.

Corollary 4: (a) The minimax H_∞ tracking strategy in (13) or (17) for the nonlinear time-varying tracking error system in (16) is solved by:

$$u^*(t) = -\frac{1}{2}R^{-1}G_e^T(e(t),t)\left(\frac{\partial V(e(t),t)}{\partial e(t)}\right)$$
$$v^*(t) = \frac{1}{2\rho}D_e^T(e(t),t)\left(\frac{\partial V(e(t),t)}{\partial e(t)}\right) \tag{18}$$

where $V(e(t),t) \ge 0$ and $V(0,t) = 0$ is the solution of the following HJIE:

$$\begin{aligned} &HJIE \\ &= \frac{\partial V(e(t),t)}{\partial t} + \left(\frac{\partial V(e(t),t)}{\partial e(t)}\right)^T F_e(e(t),t) + e^T(t)Qe(t) \\ &\quad - \frac{1}{4}\left(\frac{\partial V(e(t),t)}{\partial e(t)}\right)^T G_e(e(t),t)R^{-1}G_e^T(e(t),t) \\ &\quad \times \left(\frac{\partial V(e(t),t)}{\partial e(t)}\right) + \frac{1}{4\rho}\left(\frac{\partial V(e(t),t)}{\partial e(t)}\right)^T \\ &\quad \times D_e(e(t),t)D_e^T(e(t),t)\left(\frac{\partial V(e(t),t)}{\partial e(t)}\right) = 0 \end{aligned} \tag{19}$$

(b) If the nonlinear system in (5) is free of external disturbance or the tracking error dynamic system in (16) is free of $v(t)$, i.e., $v(t) = 0$, than the H_∞ tracking control $u^*(t)$ in (18) will achieve the asymptotic tracking ability, i.e., $e(t) \to 0$ as $t \to \infty$.

Proof: Similar to Appendix A. □

The robust H_∞ reference tracking problem in Corollary 4 is equivalent to solving the robust H_∞ stabilization problem in (2) of the nonlinear time-varying system in (1) in Theorem 1 only with $e(t)$ replacing $x(t)$. Therefore, the robust H_∞ stabilization design method in Theorem 1 can be employed to treat this equivalent H_∞ stabilization problem of the tracking error system in (16) and (17) in Corollary 4 only with $e(t)$ replacing $x(t)$. In general, it is very difficult to solve the H_∞ stabilization design problem in (3) and (4) and H_∞ tracking design problem in (18) and (19) of nonlinear tracking error system (16) by the convenient methods. Following the HJIE-embedded DNN-based tracking control scheme in Fig.1 for the H_∞ stabilization control of nonlinear time-varying system in (1), based on Corollary 4, an HJIE-embedded DNN-based control scheme for the H_∞ robust tracking control strategy in (13) and (17) is also given in Fig. 2. In the off-line training phase, the state vector $x(t)$ is generated by nonlinear system model in (5) with the worst-case disturbance $v^*(t)$ and DNN-based control $u^*(t)$ as input. $x(t)$ is extracted by the reference signal $r(t)$ to obtain $e(t)$ as input into DNN to output $\frac{\partial V(\bar{x}(t),t)}{\partial \bar{x}(t)} = [(\frac{\partial V^T(x(t),t)}{\partial x(t)})\ (\frac{\partial V^T(x(t),t)}{\partial t})]^T$, which is used to generate $v^*(t)$ and $u^*(t)$ as well as is substituted into HJIE in (19) to train DNN to solve HJIE in (19) for H_∞ roubst tracking control in (18). In this study, a reference tracking of quadrotor UAV will be given as a design example by the proposed HJIE-embedded DNN H_∞ tracking control method in Corollary 4 in the sequel.

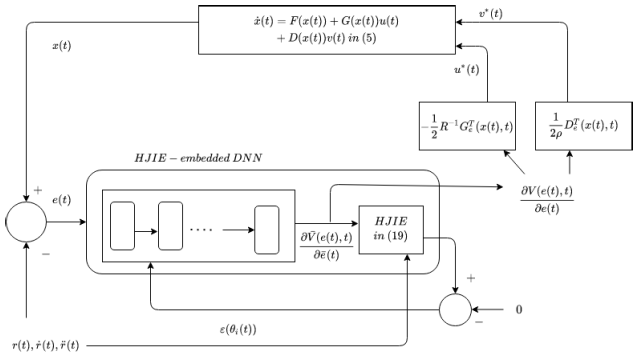

FIGURE 2. The flow chart of HJIE DNN-based reference tracking control scheme based on Corollary 4 and Fig. 1. In the off-line training phase, $u^*(t)$ and $v^*(t)$ is fedback to the nonlinear system in (5) to generate $x(t)$ and then minus $r(t)$ to obtain $e(t)$, which is accessed to DNN to output $\frac{\partial \bar{V}(\bar{e}(t))}{\bar{e}(t)}$ to solve HJIE in (19). The error of HJIE in (19) will be fed back to train DNN by Adam learning algorithm. In the on-line operation phase, $v^*(t)$ is not needed to be sent to the error dynamics and is replaced by real $v(t)$.

In general, there exists no efficient analytic or numerical method to solve the above partial differential HJIEs for robust H_∞ control design of nonlinear dynamic system, except some special nonlinear systems. In the last decades, several application methods such as T-S fuzzy method [20], [21], global linearization technique [22], [23], gain schedule scheme [24], [25], etc, have been employed to interpolate several local linearized systems to approximate the time-invariant nonlinear system in (5) so that the HJIE in (7) can be transformed to a set of local Riccati-like equations with the assumption that the Lyapunov function $V(x(t))$ in (7) is of the form $V(x(t)) = x^T(t)Px(t)$ with positive symmetric matrix P, which will limit the solution of HJIE [26]. Moreover, in some practical cases, there are a large number of Riccati-like equations to be solved for a large number of local control gains k_i. For example, in the quadrotor UAV tracking control of example 2 in the sequel, there are about 125 local fuzzy systems to approximate the nonlinear quadrotor dynamic system in [27]. Furthermore, we need to compute fuzzy control signal $u(t) = \sum_{i=1}^{125} h_i(x)k_ix(t)$ at every time instant, where $h_i(x)$ denotes the ith fuzzy interpolation basis. Moreover, there still exist some difficulties to treat the nonlinear time-varying system in (1) by the conventional T-S fuzzy method and global linearization technique which have always been employed to interpolate some local linear time-invariant systems to approximate nonlinear

time-invariant systems. More efforts are still needed for approximating nonlinear time-varying systems by these methods [20]–[23]. For example, internal type-2 fuzzy model is needed for the approximation of time-varying nonlinear systems, which will increase the complexity of control design procedures.

Recently, deep learning schemes have been employed to learn system behavior by data-driven approach through a large amount of empirical data and have been successfully applied to image classification, communication, translation of language, speech recognition, etc [1]–[3]. Even some of these deep learning schemes have very good results, they need a very large amount of data for training. Further, they cannot guarantee some specific performance like H_2 optimal control performance or H_∞ robust control performance. Nevertheless, few deep neural network learning methods have been applied to control system designs and optimization methods. In the system control field, system dynamic models and theoretical control design methods have been well developed and available for several decades. They can be considered as priori expert knowledge in deep learning approaches to solve some difficult problems in the control system designs such as the above min max H_∞ reference tracking control system design problem in (17)–(19) of nonlinear time-varying tracking error system in (16) under external disturbance. Consequently, system model-based and HJIE-embedded DNN H_∞ control design can save much training data and training time than the conventional big data-driven DNN control method.

In the above min max H_∞ robust control design problem, the difficulty lies in how to solve $\frac{\partial V(x(t),t)}{\partial t}$ and $\frac{\partial V(x(t),t)}{\partial x(t)}$ of the nonlinear partial differential HJIE in (4), (7), (10) and (12) for H_∞ robust stabilization control or solve $\frac{\partial V(e(t),t)}{\partial t}$ and $\frac{\partial V(e(t),t)}{\partial e(t)}$ of HJIE in (19) for H_∞ robust reference tracking control. In this study, an DNN-based H_∞ control design is introduced to efficiently solve these HJIEs for H_∞ robust control design problems of nonlinear time-varying systems.

III. DNN-BASED H_∞ CONTROL DESIGN OF NONLINEAR TIME-VARYING SYSTEMS

For the nonlinear time-varying system with external disturbance in (1), the robust min max H_∞ robust control strategy $u^*(t)$ in (3) can efficiently attenuate the effect of worst-case external disturbance $v^*(t)$ on the quadratic stabilization control performance to a desired level ρ in (2). However, from Theorem 1, the min max H_∞ robust control in (3) needs to solve the HJIE in (4). In general, it is not easy to solve HJIE analytically or numerically for robust H_∞ control design. Moreover, if $V(x(t), t)$ in (4) is solved by training schemes like conventional methods, we still need to find $\frac{\partial V(x(t),t)}{\partial x(t)}$ for the H_∞ control law $u^*(t)$ and the worst-case disturbance $v^*(t)$ in (3). It is a difficult work for DNN-based control design. Therefore, we suggest solving $\frac{\partial V(x(t),t)}{\partial x(t)}$ and $\frac{\partial V(x(t),t)}{\partial t}$ instead of $V(x(t), t)$ from HJIE in (4) by training DNN directly.

In the off-line training phase, the training process of the proposed HJIE-embedded DNN for $\frac{\partial V(x(t),t)}{\partial x(t)}$ and $\frac{\partial V(x(t),t)}{\partial t}$ to solve HJIE in (4) for the H_∞ control $u^*(t)$ and the worst-case disturbance $\upsilon^*(t)$ in (3) of the nonlinear time-varying system in (1) is shown in Fig. 1. In Fig. 1, the system state $x(t)$ is first generated by nonlinear dynamic model in (1), then we input state vector $x(t)$ into DNN. After several hidden layers of DNN, we expect it can produce $\frac{\partial V(x(t),t)}{\partial x(t)}$ and $\frac{\partial V(x(t),t)}{\partial t}$, i.e. $\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)} = [(\frac{\partial V(x(t),t)}{\partial t})^T \ (\frac{\partial V(x(t),t)}{\partial x(t)})^T]^T$ at the output of DNN, which will be sent to the block of HJIE to calculate the value of HJIE in the following. As a consequence, the output of HJIE-embedded DNN is expected to approach $\frac{\partial V(x(t),t)}{\partial x(t)}$ and $\frac{\partial V(x(t),t)}{\partial t}$ of HJIE $= 0$ after training. The error $\varepsilon(\theta_i(t))$ of HJIE in the training step i at time t is sent back to the DNN to update weighting parameters of DNN, i.e. if the outputs $(\frac{\partial V(x(t),t)}{\partial x(t)})_\varepsilon$ and $(\frac{\partial V(x(t),t)}{\partial t})_\varepsilon$ of DNN are substituted into HJIE to calculate its value as follows.

$$
\begin{aligned}
&HJIE_\varepsilon \\
&= \left(\frac{\partial V\,(x(t),t)}{\partial t}\right)_\varepsilon + \left(\frac{\partial V\,(x(t),t)}{\partial x(t)}\right)_\varepsilon^T F\,(x(t),t) \\
&\quad + x^T(t)Qx(t) - \frac{1}{4}(\frac{\partial V(x(t),t)}{\partial x(t)})_\varepsilon^T G(x(t),t)R^{-1}G^T(x(t),t) \\
&\quad \times (\frac{\partial V(x(t),t)}{\partial x(t)})_\varepsilon + \frac{1}{4\rho}(\frac{\partial V(x(t),t)}{\partial x(t)})_\varepsilon^T D\,(x(t),t) \\
&\quad \times D^T(x(t),t)(\frac{\partial V(x(t),t)}{\partial x(t)})_\varepsilon = \varepsilon(\theta_i(t))
\end{aligned} \tag{20}
$$

then $\varepsilon(\theta_i(t))$ will be sent back to train weighting and bias parameters of neurons in the $i+1$ training step of DNN until the output of DNN equals to $\frac{\partial V(x(t),t)}{\partial x(t)}$ and $\frac{\partial V(x(t),t)}{\partial t}$. Further, the output $\frac{\partial V(x(t),t)}{\partial x(t)}$ of DNN is multiplied by $-\frac{1}{2}R^{-1}G^T(x(t), t)$ to produce H_∞ control $u^*(t)$ in (3), which is the control input to the nonlinear system in (1), and multiplied by $\frac{1}{2\rho}D^T(x(t), t)$ to produce the worst-case external disturbance $v^*(t)$ in (3), which is also sent back to the nonlinear system in (1) as external disturbance in the training phase.

In real systems, external disturbance $\upsilon(t)$ is always unavailable. Moreover, for the H_∞ robust control, it is very difficult to generate $x(t)$ of nonlinear time-varying system in (1) for all possible $v(t) \in \mathcal{L}_2([0, \infty])$ to train DNN in the off-line training phase from the big data perspective. In this situation, the worst-case disturbance $v^*(t)$ is used to replace all possible external disturbance $v(t) \in \mathcal{L}_2([0, \infty])$ from the min max H_∞ disturbance rejection point of view in (2). Then the system state $x(t)$ generated by $u^*(t)$ and $v^*(t)$ is sent to DNN and HJIE as input signals from nonlinear time-varying system in (1) to start another cycle of training. Therefore, in the off-line training phase, the training signal vector $x(t)$ can be generated by nonlinear dynamic model in (1) with $u^*(t)$ and $v^*(t)$ instead of $u(t)$ and $\upsilon(t)$, to save the training time and amount of empirical data and reduce the complexity of conventional training schemes of DNN by big data.

On the other hand, in the operation phase, the state vector $x(t)$ can be obtained from the real nonlinear system with real external disturbance $\upsilon(t)$ and H_∞ control $u^*(t)$. Therefore, the proposed HJIE-embedded DNN can bridge the gap between deep neural network learning and traditional robust

H_∞ control theory to efficiently solve the complex nonlinear robust H_∞ control design problem and speed up the training process of HJIE-embedded DNN for nonlinear robust H_∞ control design of nonlinear time-varying system with uncertain external disturbance.

Remark 3: For the nonlinear H_∞ reference tracking control design problem in (18) and (19) in Corollary 4, the input information of HJIE-embedded DNN includes $e(t)$, $r(t)$ and $\dot{r}(t)$ as shown in (15). The flowchart of HJIE-embedded DNN reference tracking control scheme based on Corollary 4 and Fig. 1 is shown in Fig. 2.

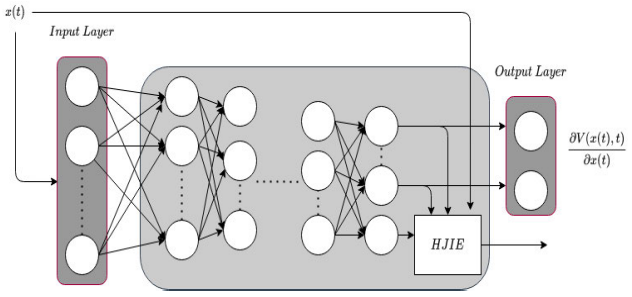

FIGURE 3. The HJIE-embedded DNN architecture of H_∞ robust control in Fig. 1. The HJIE-embedded DNN architecture of H_∞ robust reference tracking control in Fig. 2 is similar but with $x(t)$ being replaced by $e(t)$.

The architecture of HJIE-embedded DNN consists of input layer, multiple hidden layers, HJIE layer and output layer as shown in Fig. 3. The input layer, output layer and HJIE layer do not have activation function. The neurons in hidden layers use *LeakyReLU*, which is not equal to zero but has a constant gradient when the input x is negative and is the same as *ReLU* when the input x is positive, as the activation function [28]. In this way, we can keep the advantage of *ReLU* and avoid the problem of facing "dead *ReLU*" (*ReLU* is not activated when the input is negative). The definition of *LeakyReLU* is given as follows.

$$f(x) = \begin{cases} \lambda_1 x & if\ x > 0 \\ \lambda_2 x & if\ x \le 0 \end{cases}$$

where λ_1 and λ_2 are some constant and $\lambda_1, \lambda_2 \in (0, 1)$. The error $\varepsilon(\theta_i(t))$ of HJIE in (20) is fed back to train the weighting and bias parameters of DNN by Adam learning algorithm [29] to minimize the objective function $\varepsilon^2(\theta_i(t))$ as follows.

$$\theta_i(t) = \theta_{i-1}(t) - \frac{\eta}{\sqrt{\hat{v}_i(t) + \tau}} \hat{m}_i(t), \quad i = 1, \ldots, I \quad (21)$$

where $\theta_i(t)$ denotes the weighting parameter vector to be trained for DNN to output $(\frac{\partial V(x(t),t)}{\partial x(t)})_\varepsilon$ and $(\frac{\partial V(x(t),t)}{\partial t})_\varepsilon$ at time t, η denotes the learning rate or stepsize and I denotes the number of training timesteps. The bias-corrected estimators $\hat{m}(t)$ and $\hat{v}(t)$ are defined as follows.

$$\hat{m}_i(t) = \frac{m_i(t)}{1 - \beta_1^i}$$
$$\hat{v}_i(t) = \frac{v_i(t)}{1 - \beta_2^i}$$

$$m_i(t) = \beta_1 m_{i-1}(t) + (1 - \beta_1) g_i(t)$$
$$v_i(t) = \beta_2 v_{i-1}(t) + (1 - \beta_2) g_i^2(t)$$

where $g_i(t) = \frac{\partial}{\partial \theta_i(t)} \sqrt{\frac{1}{N} \sum_{i=1}^{N} \varepsilon^2(\theta_i(t))}$ is the gradient, i.e. the vector of partial derivative of objective function with respect to $\theta_i(t)$ at time step i at time t. N denotes batch size. $\beta_1, \beta_2 \in [0, 1]$ are the degree of previous impact influence on the current direction and can be specified by the designer. With β_1 and β_2, we can utilize the concept of momentum. Using the concept of momentum, we can avoid being trapped in local minimum and speed up the learning process [29]. If the direction of current gradient is the same as the accumulated gradient, then the gradient will be strengthened, otherwise, the gradient will be weakened. β_1^i and β_2^i are the ith power of β_1 and β_2, respectively. τ is a small number to prevent the denominator from being zero. $m_i(t)$, $v_i(t)$ are the moving average of gradient and squared gradient at time t. With $\hat{v}_i(t)$, we can take the advantage of the idea of adaptive learning rate, which is large at the beginning and small near the minimum. The Adam learning algorithm combines the advantages of momentum and RMSProp [30] and is an efficient parameter-specific adaptive learning method. Since Adam learning algorithm is easy to implement and has great performance, it is one of the most popular optimizer being used recently.

Remark 4: From [29], the convergence of weighting parameter vector has been proven. If the number of hidden nodes and timesteps are large enough, the neural network updating weighting parameter $\theta_i(t)$ by Adam learning algorithm can converge to a globally optimal $\theta_i^*(t)$ at a linear convergence rate as $i \to \infty$ in (21).

In the case of robust H_∞ control strategy in (2), we need to solve the time-varying partial differential HJIE in (4) for the H_∞ control of $u^*(t)$ and the worst-case external disturbance $v^*(t)$ in (3). In this situation, we denote

$$\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)} = \begin{bmatrix} \frac{\partial V(x(t),t)}{\partial x(t)} \\ \frac{\partial V(x(t),t)}{\partial t} \end{bmatrix}, \quad \bar{F}(\bar{x}(t), t) = \begin{bmatrix} F(x(t), t) \\ 1 \end{bmatrix}$$
$$\bar{R}^{-1} = \begin{bmatrix} R^{-1} & 0 \\ 0 & 0 \end{bmatrix}, \quad \bar{G}(\bar{x}(t), t) = \begin{bmatrix} G(x(t), t) & 0 \\ 0 & 0 \end{bmatrix},$$
$$\bar{D}(\bar{x}(t), t) = \begin{bmatrix} D(x(t), t) \\ 0 \end{bmatrix}$$

Then the time-varying HJIE of H_∞ control strategy in (4) can be represented by:

$$\begin{aligned} HJIE = & \left(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)} \right)^T \bar{F}(\bar{x}(t), t) + x^T(t) Q x(t) \\ & - \frac{1}{4} \left(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)} \right)^T \bar{G}(\bar{x}(t), t) \bar{R}^{-1} \bar{G}^T(\bar{x}(t), t) \\ & \times \left(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)} \right) + \frac{1}{4\rho} \left(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)} \right)^T \bar{D}(\bar{x}(t), t) \\ & \times \bar{D}^T(\bar{x}(t), t) \left(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)} \right) = 0 \end{aligned} \quad (22)$$

In the off-line training process, the output $(\frac{\partial \bar{V}(\bar{x})}{\partial \bar{x}})_\varepsilon$ of DNN is sent to HJIE to calculate error $\varepsilon(\theta_i(t))$ in the following as shown in Fig. 1.

$$\begin{aligned} HJIE_\varepsilon = & \left(\frac{\partial \bar{V}\left(\bar{x}(t)\right)}{\partial \bar{x}(t)}\right)_\varepsilon^T \bar{F}\left(\bar{x}(t)\right) + x^T(t)\, Q x(t) \\ & - \frac{1}{4}\left(\frac{\partial \bar{V}\left(\bar{x}(t)\right)}{\partial \bar{x}(t)}\right)_\varepsilon^T \bar{G}\left(\bar{x}(t)\right) \bar{R}^{-1} \bar{G}^T\left(\bar{x}(t)\right) \\ & \times \left(\frac{\partial \bar{V}\left(\bar{x}(t)\right)}{\partial \bar{x}(t)}\right)_\varepsilon + \frac{1}{4\rho}\left(\frac{\partial \bar{V}\left(\bar{x}(t)\right)}{\partial \bar{x}(t)}\right)_\varepsilon^T \bar{D}\left(\bar{x}(t)\right) \\ & \times \bar{D}^T\left(\bar{x}(t)\right)\left(\frac{\partial \bar{V}\left(\bar{x}(t)\right)}{\partial \bar{x}(t)}\right)_\varepsilon = \varepsilon(\theta_i(t)) \end{aligned} \tag{23}$$

The error $\varepsilon(\theta_i(t))$ of HJIE will be sent back to train DNN to output the precise $\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)}$ for robust H_∞ control signal $u^*(t) = -\frac{1}{2}R^{-1}G^T(x(t),t)\left(\frac{\partial V(x(t),t)}{\partial x(t)}\right)$ and the worst-case external disturbance $v^*(t) = \frac{1}{2\rho}D^T(x(t),t)(\frac{\partial V(x(t),t)}{\partial x(t)})$ in the training process.

The following theorem is to prove that when the error $\varepsilon(\theta_i(t))$ of embedded-HJIE DNN approaches 0 in Adam learning process, the output $(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)})_\varepsilon$ of DNN can approach $(\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)})$.

Theorem 2: If $\varepsilon(\theta_i(t)) \to 0$ in (23) by Adam learning algorithm, then $(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)})_\varepsilon \to (\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)})$ in (22), i.e., HJIE-embedded DNN-based control $u^*(t)$ in Fig. 1 will approach the theoretical H_∞ robust control $u^*(t)$ in (3) of nonlinear time-varying system in (1).

Proof: See Appendix B. □

Remark 5: In Theorem 2, it is seen that the proposed HJIE-embedded DNN-based control scheme $u^*(t)$ in Fig. 1 will approach the minmax H_∞ control $u^*(t)$ in (3) as output $\varepsilon(\theta_i(t))$ of HJIE in (23) approaches zero after the training process of Adam learning scheme in (21). By Theorem 1, as $v(t) = 0$, the minmax H_∞ control $u^*(t)$ in (3) could achieve the asymptotic stability of nonlinear time-varying system in (1) when $v(t) = 0$. Therefore, as $\varepsilon(\theta_i(t)) \to 0$, the proposed HJIE-embedded DNN-based control will achieve the asymptotic stability of nonlinear time-varying in (1) with $v(t) = 0$.

Remark 6: Recently, the disturbance observer-based control have been widely used for nonlinear system to eliminate the external disturbance [45]. However, the conventional disturbance observer-based control is always based on the singular descriptor model. In general, the disturbance observer-based control needs a singular descriptor model to estimate disturbance which is very difficult to design for nonlinear time-varying system. In this study, we do not need to estimate external disturbance but it effect can be effectively attenuated by the DNN-based H_∞ control strategy in (2) to a desired attenuation level ρ of external disturbance by minimizing their worst-case effect in (2). The proposed DNN-based control scheme could be applied to solve $\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)}$ of HJIE in (22) for the optimal control $u^*(t)$ to directly attenuate the effect of external disturbance on the quadratic performance to a desired level ρ

In the off-line training phase, we input state vector $x(t)$, which is generated by nonlinear system model in (1) with the H_∞ control $u^*(t)$ and worst-case external disturbance $v^*(t)$, into DNN as shown in Fig. 1. In this way, we can train HJIE-embedded DNN to output $\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)}$ if $\varepsilon(\theta_i(t))$ calculated by HJIE approaches 0. Nevertheless, in the practical applications, we always stop the training phase and transfer into the operation phase if $\varepsilon^2(\theta_i(t))$ is smaller than a prescribed value $\delta > 0$. In this study, we set $\delta = 0.01$ for the design examples in the sequel. In the operation phase, based on the trained weighting parameters $\theta_i(t)$ of DNN in the off-line training phase, we can input $x(t)$ of real nonlinear dynamic systems into DNN. However, the weighting parameters $\theta_i(t)$ of DNN can be still updated by on-line training via Adam learning algorithm in (21) if $|\varepsilon(\theta_i(t))| > \delta$ during the control process in the operation phase. In fact, we could train HJIE-embedded DNN in the operation phase through Adam learning algorithm without the influence on the DNN-based control if $\varepsilon^2(\theta_i(t)) \geq \delta$ for some specific δ to improve the H_∞ robust control performance in the operation phase. In this study, unlike the conventional big data-driven DNN, since training data $x(t)$ is generated by system model, we can significantly save a much amount of training data and training time of the HJIE-embedded DNN and avoid the instability of the control system at the beginning in the operation phase. Therefore, the proposed DNN-based H_∞ robust control design is suitable for the robust H_∞ control of nonlinear time-varying systems.

Similarly, in the off-line training process of the HJIE-embedded DNN-based reference tracking control scheme in Fig. 2, the error $\varepsilon_e(\theta_i(t))$ of HJIE will be sent back to train the parameters of neurons in DNN based on Adam learning algorithm in (21). Let us denote the $\varepsilon_e(\theta_i(t))$ as follows:

$$\begin{aligned} & HJIE_\varepsilon \\ & = \left(\frac{\partial \bar{V}(\bar{e}(t),t)}{\partial \bar{e}(t)}\right)_\epsilon^T \bar{F}_e(\bar{e}(t),t) + e^T(t) Q e(t) \\ & \quad - \frac{1}{4}\left(\frac{\partial \bar{V}(\bar{e}(t),t)}{\partial \bar{e}(t)}\right)_\epsilon^T \bar{G}_e(\bar{e}(t),t) \bar{R}^{-1} \bar{G}_e^T(\bar{e}(t),t) \\ & \quad \times \left(\frac{\partial \bar{V}(\bar{e}(t),t)}{\partial \bar{e}(t)}\right)_\epsilon + \frac{1}{4\rho}\left(\frac{\partial \bar{V}(\bar{e}(t),t)}{\partial \bar{e}(t)}\right)_\epsilon^T \bar{D}_e(\bar{e}(t),t) \\ & \quad \times \bar{D}_e^T(\bar{e}(t),t)\left(\frac{\partial \bar{V}(\bar{e}(t),t)}{\partial \bar{e}(t)}\right)_\epsilon = \varepsilon_e(\theta_i(t)) \end{aligned} \tag{24}$$

where

$$\frac{\partial \bar{V}(\bar{e}(t),t)}{\partial \bar{e}(t)} = \begin{bmatrix} \frac{\partial V(e(t),t)}{\partial e(t)} \\ \frac{\partial V(e(t),t)}{\partial t} \end{bmatrix}, \quad \bar{F}_e(\bar{e}(t),t) = \begin{bmatrix} F_e(e(t),t) \\ 1 \end{bmatrix}$$

$$\bar{R} = diag\{R, 0\}, \quad \bar{G}_e(\bar{e}(t),t) = diag\{G_e(e(t),t), 1\}$$

$$\bar{D}_e(\bar{e}(t),t) = \begin{bmatrix} D_e(e(t),t) \\ 0 \end{bmatrix}$$

Then we get the following result of the HJIE-embedded DNN-based reference tracking control scheme in Fig. 2 for the minmax H_∞ reference tracking control strategy in Corollary 4.

Corollary 5: If $\varepsilon_e(\theta_i(t))$ approaches to 0 in (24) by Adam learning algorithm, than $\left(\frac{\partial \bar{V}(\bar{e}(t),t)}{\partial \bar{e}(t)}\right)_\epsilon$ in (24) approaches to $\frac{\partial \bar{V}(\bar{e}(t),t)}{\partial \bar{e}(t)}$ in (19), i.e., HJIE-embedded DNN reference control $u^*(t)$ in Fig. 2 will approach the H_∞ robust reference tracking control $u^*(t)$ in (18) of the nonlinear system in (5).

Proof: Similar to Appendix B. □

Obviously, from Corollary 5, if $\varepsilon_e(\theta_i(t))$ approaches to zero in the HJIE-embedded DNN reference tracking control scheme in Fig. 2, the DNN will output $\frac{\partial \bar{V}(\bar{e}(t),t)}{\partial \bar{e}(t)}$ to produce H_∞ robust reference tracking control $u^*(t)$ in (18) to achieve the minmax H_∞ tracking strategy in (13) or (17).

However, in practical applications, we will stop the training process if $|\varepsilon_e(\theta_i(t))| < \eta$ for a prescribed threshold $\eta > 0$. In the sequel, an example of reference tracking control of nonlinear quadrotor UAV based on HJIE-embedded DNN H_∞ tracking scheme in Fig. 2 will be given to illustrate the design procedure of HJIE-embedded DNN-based H_∞ reference tracking control scheme and validate it's H_∞ reference tracking performance.

Remark 7: The minimax H_∞ control design problem in Theorem 1 will be reduced to the optimal quadratic control design problem when $v(t) = 0$ in (1) and $\rho = \infty$, i.e., $u^*(t) = -\frac{1}{2}R^{-1}G^T(x,t)(\frac{\partial V(x(t),t)}{\partial x(t)})$ in (3) is the optimal control solution of $min_{u(t)} \int_0^\infty [x^T(t)Qx(t) + u^T(t)Ru(t)]dt$ for nonlinear time-varying system in (1) without external disturbance $v(t)$ where $V(x(t),t) > 0$ is the solution of HJE $= \frac{\partial V(x(t),t)}{\partial t} + (\frac{\partial V(x(t),t)}{\partial x(t)})^T F(x(t),t) + x^T(t)Qx(t) - \frac{1}{4}(\frac{\partial V(x(t),t)}{\partial x(t)})^T G(x(t),t) G^T(x(t),t)(\frac{\partial V(x(t),t)}{\partial x(t)}) = 0$ which is modification of HJIE in (4) with $\rho = \infty$. Therefore, only with some modification, the proposed DNN-based control scheme in Fig. 1 or Fig. 2 could be also employed to treat the optimal H_2 quadratic control of nonlinear time-varying system in (1) without consideration of external disturbance, i.e., $v(t) = 0$.

IV. SIMULATION

After the introduction of HJIE-embedded DNN H_∞ robust control design of nonlinear time-varying systems under external disturbance, two examples are given to illustrate the design procedure and to confirm the performance of the proposed design method. An example of HJIE-embedded DNN H_∞ robust stabilization control scheme is provided at first for a nonlinear time-varying system with high frequency external disturbance. The second example is to design an HJIE-embedded DNN H_∞ reference tracking control of practical quadrotor unmanned aerial vehicle (UAV).

Example 1: Robust DNN-based finite-time H_∞ control design of nonlinear time-varying system with high frequency external disturbance.

Suppose we want to design an HJIE-embedded DNN H_∞ control with $\rho = 1$ and terminal time $t_f = 80s$ in (8) for the following nonlinear time-varying system with high frequency external disturbance.

$$\begin{aligned}\dot{x_1}(t) &= \frac{x_1 \sin 5t}{1 + x_1(t)\, x_2(t)} + (1 + x_2(t))\, u_1(t) \\ &\quad + x_1(t) \sin 30t \\ \dot{x_2}(t) &= \frac{x_2 \cos t}{1 + 2x_1(t)\, x_2(t)} + (0.5 + x_1(t))\, u_2(t) \\ &\quad + x_2(t) \cos 30t * N(0, 0.1)\end{aligned} \tag{25}$$

where $N(0, 0.1)$ denotes the normal distribution with zero mean and 0.1 variance. Let us denote

$$X(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}, \quad F(X(t), t) = \begin{bmatrix} \frac{x_1(t) \sin 5t}{1+x_1(t)x_2(t)} \\ \frac{x_2(t) \cos t}{1+2x_1(t)x_2(t)} \end{bmatrix},$$
$$G(X, t) = \begin{bmatrix} 1 + x_2(t) & 0 \\ 0 & 0.5 + x_1(t) \end{bmatrix}, \quad u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix},$$
$$D(X, t) = \begin{bmatrix} x_1(t) & 0 \\ 0 & x_2(t) \end{bmatrix}, \quad \upsilon(t) = \begin{bmatrix} \sin 30t \\ \cos 30t * N(0, 0.1) \end{bmatrix}$$

According to Corollary 2, we need to solve $\frac{\partial V(x(t),t)}{\partial x(t)}$ and $\frac{\partial V(x(t),t)}{\partial t}$ of the following time-varying HJIE by DNN learning for the robust H_∞ control design.

$$\begin{aligned}&(\frac{\partial V(x(t), t)}{\partial t}) + (\frac{\partial V(x(t), t)}{\partial x(t)})^T \begin{bmatrix} \frac{x_1 \sin 5t}{1+x_1(t)x_2(t)} \\ \frac{x_2 \cos t}{1+2x_1(t)x_2(t)} \end{bmatrix} \\ &- \frac{1}{4}(\frac{\partial V(x, t)}{\partial x})^T \begin{bmatrix} 1 + x_2(t) & 0 \\ 0 & 0.5 + x_1(t) \end{bmatrix} R^{-1} \\ &\times \begin{bmatrix} 1 + x_2(t) & 0 \\ 0 & 0.5 + x_1(t) \end{bmatrix}^T (\frac{\partial V(x(t), t)}{\partial x(t)}) \\ &+ x^T(t)Qx(t) + \frac{1}{4\rho}(\frac{\partial V(x(t), t)}{\partial x(t)})^T \begin{bmatrix} x_1(t) & 0 \\ 0 & x_2(t) \end{bmatrix} \\ &\times \begin{bmatrix} x_1(t) & 0 \\ 0 & x_2(t) \end{bmatrix}^T \left(\frac{\partial V(x(t), t)}{\partial x(t)}\right) = 0\end{aligned} \tag{26}$$

In general, it is almost impossible to solve $\frac{\partial V(x(t),t)}{\partial x(t)}$ and $\frac{\partial V(x(t),t)}{\partial t}$ from the above HJIE analytically or numerically for the robust H_∞ control law $u^*(t) = -\frac{1}{2}R^{-1}G^T(x(t), t)(\frac{\partial V(x(t),t)}{\partial x(t)})$ and worst-case external disturbance $\upsilon^*(t) = \frac{1}{2\rho}D^T(x(t), t)(\frac{\partial V(x,t)}{\partial x})$ in (9). Therefore, the proposed HJIE-embedded DNN H_∞ control scheme is employed to treat the H_∞ robust stabilization of the nonlinear time-varying system in (25). Let us denote

$$\frac{\partial \bar{V}(\bar{x})}{\partial \bar{x}(t)} = \begin{bmatrix} \frac{\partial V(x(t),t)}{\partial x(t)} \\ \frac{\partial V(x(t),t)}{\partial t} \end{bmatrix}, \quad \bar{F}(\bar{x}(t)) = \begin{bmatrix} \frac{x_1(t) \sin 5t}{1+x_1(t)x_2(t)} \\ \frac{x_2(t) \cos 5t}{1+2x_1(t)x_2(t)} \\ 1 \end{bmatrix},$$
$$\bar{R}^{-1} = \begin{bmatrix} R^{-1} & 0 \\ 0 & 0 \end{bmatrix}, \quad \bar{D}(\bar{x}(t)) = \begin{bmatrix} x_1(t) & 0 \\ 0 & x_2(t) \\ 0 & 0 \end{bmatrix},$$
$$\bar{G}(\bar{x}(t)) = \begin{bmatrix} 1 + x_2(t) & 0 & 0 \\ 0 & 0.5 + x_1(t) & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

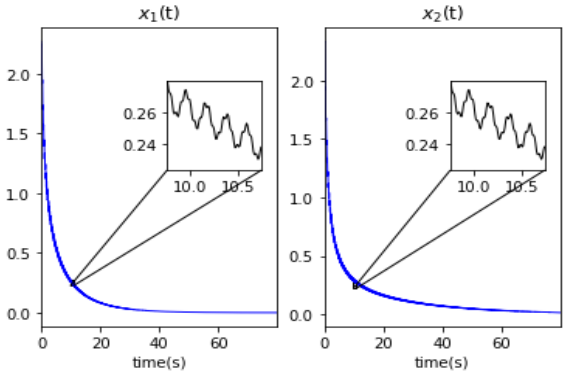

FIGURE 4. State trajectories of example 1 under DNN-based H_∞ control. The proposed HJIE-embedded DNN H_∞ control could achieve the robust stabilization of nonlinear time-varying system with high frequency external disturbance in (1). The zoom-in fluctuations in $x_1(t)$ and $x_2(t)$ show the effect of high frequency external disturbance on the state trajectories, which has been significantly attenuated by the DNN-based H_∞ control scheme.

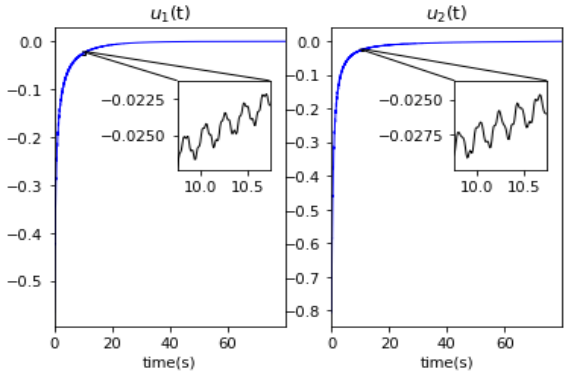

FIGURE 5. Control signals of example 1 under DNN-based H_∞ control. The zoom-in control signals fluctuate reversely with the zoom-in trajectories in Fig. 4 show the proposed DNN-based H_∞ control scheme could override the effect of hight frequency external disturbance.

Then the HJIE in (22) needs to be solved for H_∞ control $u^*(t)$ and the worst-case disturbance $v^*(t)$. From the flow chart in Fig. 1, HJIE-embedded DNN is trained through Adam learning algorithm in (21) by $HJIE_\varepsilon = \varepsilon(\theta_i(t))$ in (23) to solve the time-varying HJIE in (26) for robust H_∞ control $u^*(t)$ and worst-case disturbance $v^*(t)$ in (9) with 20000 initial conditions which are chosen randomly around the origin in the off-line training phase. After $|\varepsilon(\theta_i(t))| \leq 0.01$, we stop the offline training phase and begin the operation phase. The architecture of DNN in Fig. 3 used in this example contains input layers, four hidden layers, an HJIE layer and an output layer. The hidden layers consist of 50, 30, 10 and 3 hidden units, sequentially. We set the parameters in Adam algorithm as $\beta_1 = 0.9, \beta_2 = 0.999, \tau = 10^{-7}$ and $N = 30$. The weighting matrices are chosen as: $Q = I_2, R = I_2$. The sampling time used in this example is 0.01. After HJIE-embedded DNN H_∞ controller being trained by Adam learning algorithm, the simulation results of state trajectory $x_1(t)$ and $x_2(t)$ of the nonlinear time-varying system in (25) controlled by the trained HJIE-embedded DNN H_∞ controller in operation phase with the initial condition $x(0) = [2.27\ 2.33]^T$ are shown in Fig. 4. The state trajectories have become stable under the HJIE-embedded DNN H_∞ control after a short transient response in the beginning under the impact of external disturbance. The corresponding control signals generated by the trained HJIE-embedded DNN H_∞ controller are found in Fig. 5. The real H_∞ control performance of the proposed HJIE-embedded DNN H_∞ control scheme in Fig. 1 for nonlinear time-varying system in (25) is calculated as follows:

$$\frac{\int_0^{80}[x^T(t)Qx(t) + u^T(t)Ru(t)]dt - V(x(0),0)}{\int_0^{\infty} \upsilon^T(t)\upsilon(t)dt} \tilde{=} 0.15 \quad (27)$$

The zoom-in fluctuation in Fig. 4 is the effect of high frequency external disturbance on the state trajectories of nonlinear time-varying system in (1), which can be significantly attenuated by the proposed HJIE-embedded DNN-based H_∞ control scheme. From the zoom-in control signals in Fig. 5, it is seen that the DNN-based H_∞ control $u^*(t)$ in zoom is found to be reversely fluctuated with zoom-in state trajectory to significantly override the effect of the high frequency external disturbance on the state trajectory of nonlinear time-varying system in (25).

Example 2: DNN-based H_∞ reference tracking control scheme of quadrotor UAV.

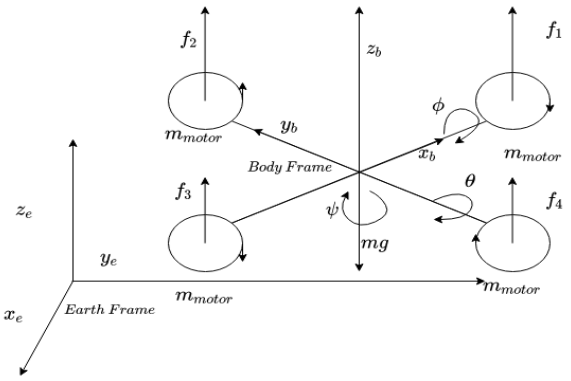

FIGURE 6. Quadrotor UAV system in example 2.

The H_∞ reference tracking design example of quadrotor unmanned aerial vehicle (UAV) is given to demonstrate the tracking performance of the proposed DNN-based H_∞ control scheme. In [31], [32], two reference frames including the inertial earth-fixed frame (x_e, y_e, z_e) and the body-fixed frame (x_b, y_b, z_b) are needed to model the dynamic of quadrotor UAV. The schematic dynamic system of a quadrotor UAV is presented in Fig. 6. The position of the center of the gravity of the quadrotor is denoted by $[x\ y\ z]^T$. The position vector is to describe the linear position related to the inertial frame $(\mathcal{E})$. Moreover, the orientation of the quadrotor depends on three Euler angles $[\theta\ \phi\ \psi]^T$, which are roll, pitch and yaw, respectively. The orientation vector is associated with the body frame $(\mathcal{B})$. The linear velocity and acceleration of the quadrotor in the earth-frame are given as $[\dot{x}\ \dot{y}\ \dot{z}]^T$ and $[\ddot{x}\ \ddot{y}\ \ddot{z}]^T$, respectively. The angular velocity and angular acceleration

of the quadrotor in the body-frame are denoted by $[\dot{\theta}\ \dot{\phi}\ \dot{\psi}]^T$ and $[\ddot{\theta}\ \ddot{\phi}\ \ddot{\psi}]^T$, respectively. To apply the proposed method to the quadrotor UAV, we transform the H_∞ tracking problem in (13) of the quadrotor UAV into the H_∞ stabilization problem (17) of time-varying tracking error system in (16). We denote the tracking error of distance and angular by $e_1(t) = x(t) - x_d(t)$, $e_3(t) = y(t) - y_d(t)$, and $e_5(t) = z(t) - z_d(t)$, $e_7(t) = \theta(t) - \theta_d(t)$, $e_9(t) = \phi(t) - \phi_d(t)$, $e_{11}(t) = \psi(t) - \psi_d(t)$, respectively, where $x_d(t)$, $y_d(t)$, $z_d(t)$ are desired reference position and $\theta_d(t)$, $\phi_d(t)$, $\psi_d(t)$ are the desired reference roll, pitch and yaw angle, respectively. The velocity error and angular velocity error are given as $e_2(t) = \dot{x}(t) - \dot{x}_d(t)$, $e_4(t) = \dot{y}(t) - \dot{y}_d(t)$, and $e_6(t) = \dot{z}(t) - \dot{z}_d(t)$, $e_8(t) = \dot{\theta}(t) - \dot{\theta}_d(t)$, $e_{10}(t) = \dot{\phi}(t) - \dot{\phi}_d(t)$, $e_{12}(t) = \dot{\psi}(t) - \dot{\psi}_d(t)$, respectively. Therefore, the tracking error dynamics of the quadrotor can be expressed using the Newton-Euler equations as (28), shown at the bottom of the page, where $u_t(t)$, $u_\theta(t)$, $u_\phi(t)$, $u_\psi(t)$ are the total thrust and the rotational force produced by four rotors, respectively. $\upsilon_x(t)$, $\upsilon_y(t)$, $\upsilon_z(t)$ are external disturbances in the three translation dynamics of quadrotor UAV, respectively. $\upsilon_\theta(t)$, $\upsilon_\phi(t)$, $\upsilon_\psi(t)$ are external disturbances caused by the unexpected rotation forces in raw, pitch, yaw dynamic quadrotor UAV, respectively. m denotes the mass of quadrotor UAV, J_θ, J_ϕ, J_ψ are the moments of inertia, $K_x, K_y, K_z, K_\theta, K_\phi, K_\psi$ are the aerodynamic damping coefficients, g is the acceleration of gravity, l represents the distance from the center of each quadrotor UAV to the center of gravity, and c represents constant of force-to-moment factor.

By the conventional H_∞ fuzzy tracking control of quadrotor UAV in [27], 125 fuzzy local linear systems are used to interpolate to approximate the nonlinear tracking error system in [27] due to its highly nonlinearity of quadrotor UAV. In this situation, we need to solve 125 Riccati-like inequalities for H_∞ fuzzy controller $u(t) = \sum_{i=1}^{125} h_i(x)k_i x(t)$, which needs to be computed in every time instant, too. Hence, more efforts are needed to employ H_∞ fuzzy controller for practical applications. Therefore, the proposed DNN-based H_∞ control design in Corollary 4 is employed to simplify the robust H_∞ tracking control design of quadrotor UAV. Thus, the nonlinear robust H_∞ reference tracking control scheme in (17) of quadrotor UAV can be designed by the robust H_∞ reference tracking control in (18) with $\rho = 2$ for the following nonlinear time-varying tracking error system in (16):

$$\dot{e}(t) = F_e(e(t), t) + G_e(e(t), t)U(t) + D_e(e(t), t)V(t) \quad (29)$$

where

$$\begin{aligned} e(t) = [&x(t) - x_d(t), \dot{x}(t) - \dot{x}_d(t), y(t) - y_d(t), \\ &\dot{y}(t) - \dot{y}_d(t), z(t) - z_d(t), \dot{z}(t) - \dot{z}_d(t), \\ &\theta(t) - \theta_d(t), \dot{\theta}(t) - \dot{\theta}_d(t), \phi(t) - \phi_d(t), \\ &\dot{\phi}(t) - \dot{\phi}_d(t), \psi(t) - \psi_d(t), \dot{\psi}(t) - \dot{\psi}_d(t)]^T, \end{aligned}$$

$$F_e(e(t), t) = \begin{bmatrix} e_2(t) \\ -\frac{K_x}{m}e_2(t) - \frac{K_x}{m}\dot{x}_d(t) - \ddot{x}_d(t) \\ e_4(t) \\ -\frac{K_y}{m}e_4(t) - \frac{K_y}{m}\dot{y}_d(t) - \ddot{y}_d(t) \\ e_6(t) \\ -\frac{K_x}{m}e_6(t) - \frac{K_z}{m}\dot{z}_d(t) - \ddot{z}_d(t) - g \\ e_8(t) \\ -\frac{k_\theta l}{J_\theta}e_8(t) - \frac{k_\theta l}{J_\theta}\dot{\theta}_d(t) - \ddot{\theta}_d(t) \\ e_{10}(t) \\ -\frac{k_\phi l}{J_\phi}e_{10}(t) - \frac{k_\phi l}{J_\phi}\dot{\phi}_d(t) - \ddot{\phi}_d(t) \\ e_{12}(t) \\ -\frac{k_\psi}{J_\psi}e_{12}(t) - \frac{k_\psi}{J_\psi}\dot{\psi}_d(t) - \ddot{\psi}_d(t) \end{bmatrix},$$

$$U(t) = \begin{bmatrix} u_t(t)\ u_\theta(t)\ u_\phi(t)\ u_\psi(t) \end{bmatrix}^T,$$

$$V(t) = [\upsilon_x(t)\ \upsilon_y(t)\ \upsilon_z(t)\ \upsilon_\theta(t)\ \upsilon_\phi(t)\ \upsilon_\psi(t)]^T$$

$$\begin{bmatrix} \dot{e}_1(t) \\ \dot{e}_2(t) \\ \dot{e}_3(t) \\ \dot{e}_4(t) \\ \dot{e}_5(t) \\ \dot{e}_6(t) \\ \dot{e}_7(t) \\ \dot{e}_8(t) \\ \dot{e}_9(t) \\ \dot{e}_{10}(t) \\ \dot{e}_{11}(t) \\ \dot{e}_{12}(t) \end{bmatrix} = \begin{bmatrix} e_2(t) \\ -\frac{K_x}{m}e_2(t) - \frac{K_x}{m}\dot{x}_d(t) - \ddot{x}_d(t) \\ e_4(t) \\ -\frac{K_y}{m}e_4(t) - \frac{K_y}{m}\dot{y}_d(t) - \ddot{y}_d(t) \\ e_6(t) \\ -\frac{K_x}{m}e_6(t) - \frac{K_z}{m}\dot{z}_d(t) - \ddot{z}_d(t) - g \\ e_8(t) \\ -\frac{k_\theta l}{J_\theta}e_8(t) - \frac{k_\theta l}{J_\theta}\dot{\theta}_d(t) - \ddot{\theta}_d(t) \\ e_{10}(t) \\ -\frac{k_\phi l}{J_\phi}e_{10}(t) - \frac{k_\phi l}{J_\phi}\dot{\phi}_d(t) - \ddot{\phi}_d(t) \\ e_{12}(t) \\ -\frac{k_\psi}{J_\psi}e_{12}(t) - \frac{k_\psi}{J_\psi}\dot{\psi}_d(t) - \ddot{\psi}_d(t) \end{bmatrix} + \begin{bmatrix} 0 \\ (\cos(e_9(t) + \phi_d(t))\sin(e_7(t) + \theta_d(t)) \\ \cos(e_{11}(t) + \psi_d(t)) + \sin(e_9(t) + \phi_d(t)) \\ \sin(e_{11}(t) + \psi_d(t)))\ \frac{u_t(t)}{m} \\ 0 \\ (\cos(e_9(t) + \phi_d(t))\sin(e_7(t) + \theta_d(t)) \\ \cos(e_{11}(t) + \psi_d(t)) - \sin(e_9(t) + \phi_d(t)) \\ \sin(e_{11}(t) + \psi_d(t)))\ \frac{u_t(t)}{m} \\ 0 \\ (\cos(e_9(t) + \phi_d(t))\cos(e_7(t) + \theta_d(t)))\ \frac{u_t(t)}{m} \\ 0 \\ \frac{l}{J_\theta}u_\theta(t) \\ 0 \\ \frac{l}{J_\phi}u_\phi(t) \\ 0 \\ \frac{c}{J_\psi}u_\psi(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \upsilon_x(t) \\ 0 \\ \upsilon_y(t) \\ 0 \\ \upsilon_z(t) \\ 0 \\ \upsilon_\theta(t) \\ 0 \\ \upsilon_\phi(t) \\ 0 \\ \upsilon_\psi(t) \end{bmatrix} \quad (28)$$

$$D_e(e(t),t)=\begin{bmatrix}0&1&0&0&0&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0&0&0&0\\0&0&0&0&0&1&0&0&0&0&0&0\\0&0&0&0&0&0&0&1&0&0&0&0\\0&0&0&0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&0&0&0&0&1\end{bmatrix}^T$$

It is hard to control all six outputs independently due to the underactuated property of the quadrotor [27]. In this case, we only select the desired position $x_d(t)$, $y_d(t)$, $z_d(t)$ and desired yaw $\psi_d(t)$ to construct the reference guidance system. The yaw trajectory is specified by the designer and we set $\psi_d(t)=0$ in this example. On the other hand, the desired roll reference $\phi_d(t)$ and the desired pitch reference $\theta_d(t)$ are computed accordingly in the following.

$$\begin{aligned}\phi_d(t)&=\sin^{-1}\left(\frac{m}{T(t)}(u_x(t)\sin\psi_d(t)-u_y(t)\cos\psi_d(t))\right)\\\theta_d(t)&=\tan^{-1}\left(\frac{1}{u_z(t)+g}(u_x(t)\cos\psi_d+u_y(t)\cos\psi_d)\right)\end{aligned}\tag{30}$$

where $T(t)$ denotes the total thrust.

From [33], the virtual input u_x, u_y, u_z and total thrust $T(t)$ can be defined as follows:

$$\begin{aligned}T(t)&=m\sqrt{u_x^2(t)+u_y^2(t)+u_z^2(t)}\\u_x(t)&=\frac{T(t)}{m}(\cos\phi_d(t)\sin\theta_d(t)\cos\psi_d(t)\\&\quad+\sin\phi_d(t)\sin\psi_d(t))\\u_y(t)&=\frac{T(t)}{m}(\cos\phi_d(t)\sin\theta_d(t)\sin\psi_d(t)\\&\quad-\sin\phi_d(t)\cos\psi_d(t))\\u_z(t)&=\frac{T(t)}{m}(\cos\phi_d(t)\cos\theta_d(t))-g\end{aligned}\tag{31}$$

The quadrotor's system parameters of this example are listed as: $K_x=K_y=K_z=0.01Ns^2/rad$, $J_\theta=J_\phi=J_\psi=0.1Ns^2/rad$, $m=2kg$, $l=1.2m$, $K_\theta=K_\phi=K_{z\psi}=0.01Ns^2/rad$, $c=1$, $g=9.8m^2/sec$. We set the radius of the trajectory as $10m$, the sampling time as $0.01s$, terminal time $t_f=30s$. The weighting matrices are chosen as: $Q=10^{-2}I_{12}, R=I_4$. The desired reference position trajectory and yaw angle trajectory are selected as: $x_d(t)=10\sin(0.5t)$, $y_d(t)=10\cos(0.5t)$, $z_d(t)=t$. The external disturbances are given as: $v_x=0.1N(0,1)$ $v_y=0.1N(0,1)$, $v_z=0.1N(0,1)$, $v_\theta=0.1N(0,1)$, $v_\phi=0.1N(0,1)$, $v_\psi=0.1N(0,1)$ where $N(0,1)$ denotes normal distribution with mean 0 and unit variance.

Based on Corollary 4, the minmax H_∞ tracking trajectory strategy in (13) and (17) for the nonlinear time-varying tracking error system of quadrotor in (29) needs to solve the following HJIE:

$$\begin{aligned}&\frac{\partial V(e(t),t)}{\partial t}+\left(\frac{\partial V(e(t),t)}{\partial e(t)}\right)^T F_e(e(t),t)+e^T(t)Qe(t)\\&-\frac{1}{4}\left(\frac{\partial V(e(t),t)}{\partial e(t)}\right)^T G_e(e(t),t)R^{-1}G_e^T(e(t),t)\\&\times\left(\frac{\partial V(e(t),t)}{\partial e(t)}\right)+\frac{1}{4\rho}\left(\frac{\partial V(e(t),t)}{\partial e(t)}\right)^T D_e(e(t),t)\\&\times D_e^T(e(t),t)\left(\frac{\partial V(e(t),t)}{\partial e(t)}\right)=0\end{aligned}\tag{32}$$

where $F_e(e(t),t)$, $G_e(e(t),t)$ and $D_e(e(t),t)$ are defined in (29) and $\rho=2$.

It is very difficult to solve the HJIE in (32) analytically or numerically for H_∞ reference tracking control $u^*(t)=-\frac{1}{2}R^{-1}G_e^T(e(t),t)\frac{\partial V(e(t),t)}{\partial e(t)}$ in Corollary 4 for the quadrotor system in (28). Therefore the proposed HJIE-embedded DNN control scheme in Fig. 2 is employed for the H_∞ reference tracking control design of quadrotor UAV. The architecture of DNN used in this example contains an input layer, five hidden layers, an HJIE layer and an output layer. There are 100, 150, 120, 80 and 13 hidden units in each hidden layer, sequentially. In the off-line training phase, we randomly select 200000 initial state vectors around

$$G_e(e(t),t)=\begin{bmatrix}0&0&0&0\\(\cos(e_9(t)+\phi_d(t))\sin(e_7(t)+\theta_d(t))\times\cos(e_{11}(t)+\psi_d(t))+\sin(e_9(t)+\phi_d(t))\times\sin(e_{11}(t)+\psi_d(t))\frac{u_t(t)}{m}&0&0&0\\0&0&0&0\\(\cos(e_9(t)+\phi_d(t))\sin(e_7(t)+\theta_d(t))\times\cos(e_{11}(t)+\psi_d(t))+\sin(e_9(t)+\phi_d(t))\times\sin(e_{11}(t)+\psi_d(t))\frac{u_t(t)}{m}&0&0&0\\0&0&0&0\\\cos(e_9(t)+\phi_d(t))\cos(e_7(t)+\theta_d(t))\frac{u_t(t)}{m}&0&0&0\\0&0&0&0\\0&\frac{l}{J_\theta}u_\theta(t)&0&0\\0&0&0&0\\0&0&\frac{l}{J_\phi}u_\phi(t)&0\\0&0&0&0\\0&0&0&\frac{c}{J_\psi}u_\psi(t)\end{bmatrix}$$

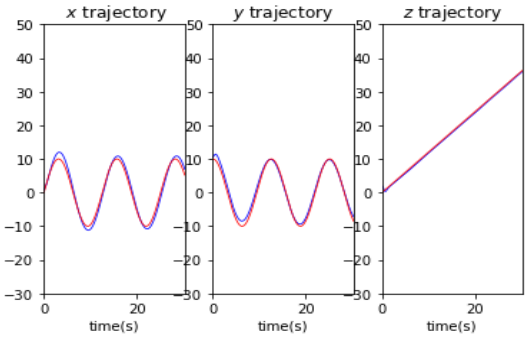

FIGURE 7. The tracking performance of three position trajectories of the UAV under HJIE-embedded DNN H_∞ tracking control.

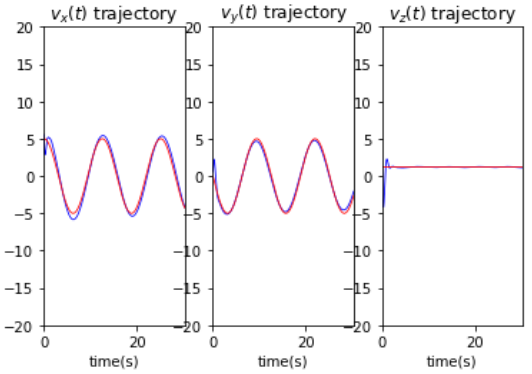

FIGURE 8. The tracking performance of three linear velocity trajectories of the UAV under the proposed HJIE-embedded DNN H_∞ tracking control.

the origin to generate $x(t)$ to obtain $e(t)$ from the training error dynamic equation in (28) and then input them into the HJIE-embedded DNN in Fig. 2 to train DNN-based H_∞ tracking control of quadrotor UAV by Adam learning algorithm in (21). We set the parameters in Adam algorithm as $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\eta = 0.001$, $\tau = 10^{-7}$ and $N = 30$. After the absolute error $|\varepsilon(\theta_i(t))|$ of HJIE is less than 0.01, we can begin the operation phase. In order to validate the tracking performance of the trained DNN-based H_∞ tracking controller of quadrotor UAV, the simulation results of operation phase with an initial state vector $x(0) = [0.54, 5.55, 10.57, 0.57, 2, 1.7, 0.5, 0, 0.52, 0.52, 0.55, 0.52]^T$ are shown in Figs. 7-10. In Figs. 7-10, the UAV state trajectories of position, angle, velocity and angular velocity can achieve the robust H_∞ tracking performance under influence of external disturbance in the operation phase. The control signals generated by the proposed HJIE-embedded DNN H_∞ controller are shown in Fig. 11. From the 3-D trajectory of quadrotor UAV as shown in Fig. 12, the quadrotor UAV can track the desired trajectory quickly after a short transient response.

The real trajectory tracking performance of the proposed HJIE-embedded DNN-based H_∞ robust reference tracking control scheme in Fig. 2 is calculated as follows:

$$\frac{\int_0^{30}[e^T(t)Qe(t) + u^T(t)Ru(t)]dt - V(e(0),0)}{\int_0^{30} v^T(t)v(t)dt} = 1.61 \quad (33)$$

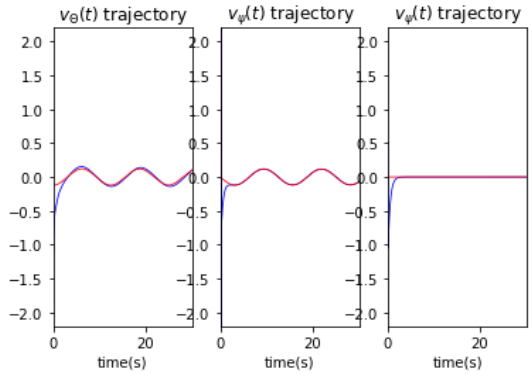

FIGURE 9. The tracking performance of three angular velocity trajectories of the UAV under the proposed HJIE-embedded DNN H_∞ tracking control.

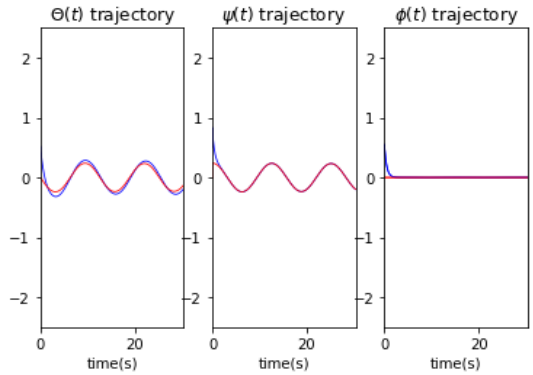

FIGURE 10. The tracking performance of three Euler angle trajectories of the UAV under the proposed HJIE-embedded DNN H_∞ tracking control.

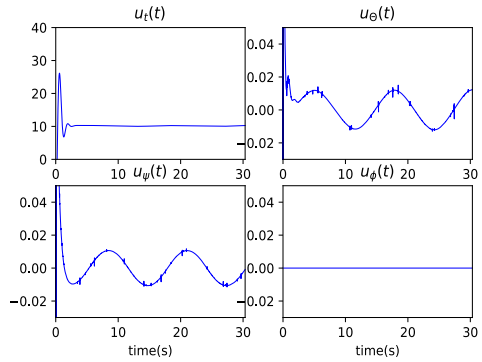

FIGURE 11. The control signals of the UAV under the proposed HJIE-embedded DNN H_∞ tracking controller.

Obviously, the proposed HJIE-embedded DNN-based H_∞ robust reference tracking control scheme in Fig. 2 could perform quiet well. For comparison, the H_∞ T-S fuzzy model reference tracking control design in [27] is used to compare with the proposed DNN-based H_∞ control scheme of quadrotor UAV. By the T-S fuzzy controller $u(t) = \sum_{i=1}^{125} h_i(x(t))K_i e(t)$ with five fuzzy rules on $\theta(t)$, $\phi(t)$ and $\psi(t)$, respectively, in [27], where $h_i(t)$ is the ith interpolation function and K_i is the ith controller gain, we need to solve

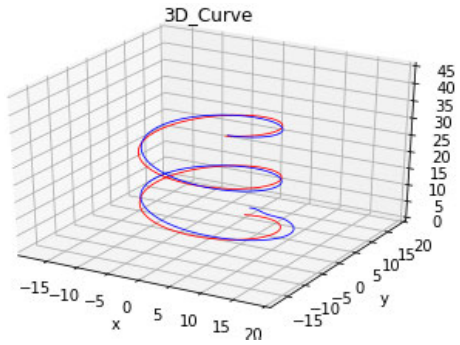

FIGURE 12. 3-D trajectory tracking performance of the UAV under HJIE-embedded based DNN H_∞ tracking control.

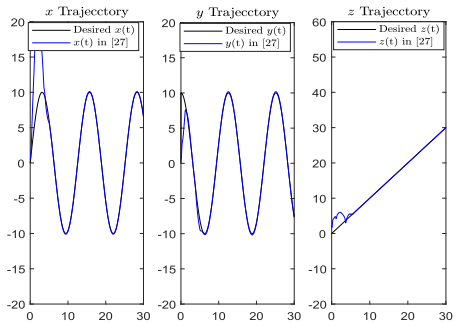

FIGURE 13. The tracking performance of three position trajectories of UAV by T-S fuzzy tracking control strategy in [27].

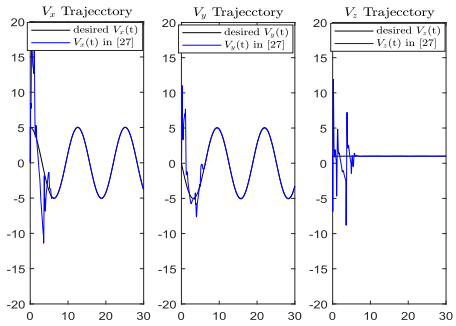

FIGURE 14. The tracking performance of three velocity trajectories of UAV by T-S fuzzy tracking control strategy in [27].

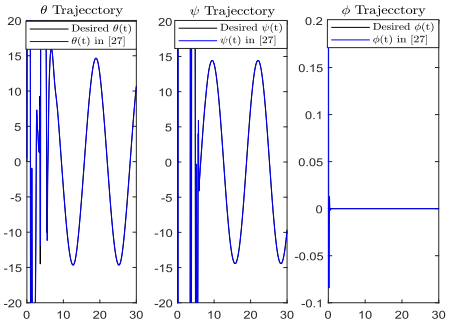

FIGURE 15. The tracking performance of three angle trajectories of UAV by T-S fuzzy tracking control strategy in [27].

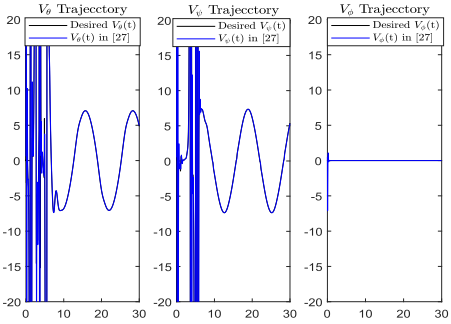

FIGURE 16. The tracking performance of three angular velocity trajectories of UAV by T-S fuzzy tracking control strategy in [27].

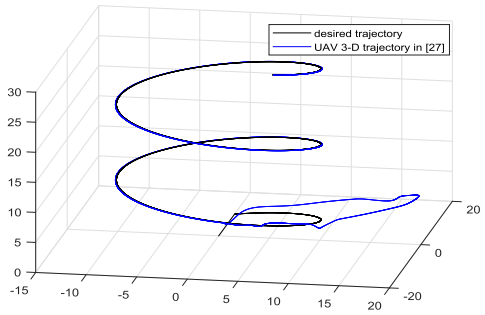

FIGURE 17. 3D trajectory of $x(t)$, $y(t)$ and $z(t)$ in UAV by T-S fuzzy tracking control strategy in [27]. The fluctuations of $V_x(t)$, $V_y(t)$, $V_z(t)$ in Fig. 14 as well as $\theta(t)$, $\phi(t)$, $\psi(t)$ in Fig. 15 and $V_\theta(t)$, $V_\phi(t)$, $V_\psi(t)$ in Fig. 16 cannot be seen in this figure.

125 linear matrix inequalities (LMIs), which are used to approximate HJIE in (32) by fuzzy interpolation finctions $\{h_i(e(t))\}_{i=1}^{125}$ under the assumption $V(e(t)) = e^T(t)Pe(t)$ for fuzzy controller $\{K_i\}_{i=1}^{125}$. The reference tracking performance of H_∞ model reference tracking control in [27] of quadrotor UAV with the same initial condition and external disturbance is shown in Figs. 13–17. The 3-D tracking performance of H_∞ T-S fuzzy tracking control is given in Fig. 17. The real tracking tracking performance of H_∞ robust T-S fuzzy model reference tracking is calculated as follows:

$$\frac{\int_0^{30}[e^T(t)Qe(t) + u^T(t)Ru(t)]dt - V(e(0), 0)}{\int_0^{30} v^T(t)v(t)dt} = 6.51 \quad (34)$$

From Figs. 13–16, there exist some fluctuation and even instability phenomenon in the initial H_∞ tracking control process. However, the fluctuations of $V_x(t)$, $V_y(t)$, $V_z(t)$ in Fig. 14 as well as $\theta(t)$, $\phi(t)$, $\psi(t)$ in Fig. 15 and $V_\theta(t)$, $V_\phi(t)$ and $V_\psi(t)$ in Fig. 16 cannot be seen in 3-D trajectories in Fig. 17, where only the trajectory $x(t)$, $y(t)$ and $z(t)$ is shown. In fact, due to the limitation of the number of fuzzy IF-THEN rules, the universe of discourse of H_∞ T-S fuzzy tracking control in [27] is limited to a small region, i.e., the T-S fuzzy tracking control strategy for UAV system in [27] can only be applied to track a small class of reference trajectory. For example, the suitable desired trajectories with $x_d(t) = 2\cos 0.5t$, $y_d(t) = 2\sin 0.5t$ and $z_d(t) = t$ can improve the tracking performance of H_∞ fuzzy tracking control with

125 fuzzy IF-THEN rules in [27]. If we want to improve the H_∞ tracking performance, we need to increase the fuzzy rules on $\theta(t)$, $\phi(t)$ and $\psi(t)$ or decrease the scale of desired trajectory. However, the increment of the fuzzy rules will increase the difficulty of design and the computational complexity of fuzzy controller $u(t)$. On the other hand, the decrease of the scale of the desired trajectory $r(t)$ will limit the application region. Instead of solving 125 LMIs, the proposed HJIE-based H_∞ reference tracking control scheme of quadrotor UAV can solve HJIE in (32) directly and globally. Therefore, the tracking performance of the proposed DNN-based H_∞ control in Fig. 2 is much better than the tracking performance of H_∞ fuzzy tracking control in [27].

V. CONCLUSION

In this study, in order to overcome the design difficulty and complexity of H_∞ robust control design of nonlinear time-varying system with external disturbance, a DNN-based control design is proposed to simplify the robust H_∞ control design as shown in Fig. 1. With the nonlinear time-varying dynamic model as well as H_∞ control $u^*(t)$ and the worst-case disturbance $v^*(t)$, we generate trajectories by nonlinear time-varying system model as input to DNN and train weighting and bias parameters of neurons in the hidden units of DNN via Adam learning algorithm by the HJIE error $\varepsilon(\theta_i(t))$ so that DNN is not only a big-data driven scheme for solving the conventional classification and recognition problem, but also a dynamic model-based scheme to be with much potential for solving system control design in nonlinear systems and filter design in nonlinear signal processing systems. With the proposed method, we can save a large amount of training data and much training time of DNN for robust DNN-based H_∞ tracking control design of nonlinear time-varying systems. In Theorem 2, we have proven that the proposed HJIE-embedded DNN-based control scheme could approach the nonlinear H_∞ robust control design by Adam learning algorithm. Therefore, the proposed DNN-based control scheme could achieve the H_∞ robust control design strategy of nonlinear time-varying system in (1) from solving HJIE in (4) directly, which is very difficult to solve analytically or numerically by conventional methods at present. Further, the DNN-based H_∞ reference tracking control of nonlinear dynamic system can be transformed to the DNN-based H_∞ stabilization problem of nonlinear time-varying tracking error system for more practical applications. In Corollary 5, we have also proven that when HJIE error $\epsilon(\theta_i(t))$ approaches to zero, the HJIE-embedded DNN reference tracking control scheme in Fig. 2 can achieve the H_∞ reference tracking performance in (13) or (17). Finally, two simulation examples of a DNN-based H_∞ stabilization design of nonlinear time-varying system and a DNN-based H_∞ reference tracking control design of complex quadrotor UAV are also provided to illustrate the design procedure and to confirm the H_∞ reference tracking performance of the proposed DNN-based H_∞ control scheme in nonlinear time-varying system with external disturbance. In future researches, we will focus on the observer-based DNN-based H_∞ control of nonlinear time-varying systems while state variables cannot be measured directly and a robust H_∞ DNN-based state estimator is needed simultaneously. Especially, since the HJIE of H_∞ observer includes the terms $F(x(t), t)$ and $D(x(t), t)$, which are function of unavailable state $x(t)$, more efforts are needed for DNN-based H_∞ observer-based control design.

APPENDIX A
PROOF OF THEOREM 1

The proof is divided into part (a) and part (b) as follows:

(a) In general, because $\upsilon(t)$ is independent of $u(t)$ in (1), the min max H_∞ control design problem in (2) is equivalent to the following min max quadratic control design problem [34].

$$\min_{u(t)} \max_{\upsilon(t)} \int_0^\infty (x^T(t)Qx(t) + u^T(t)Ru(t) - \rho\upsilon^T(t)\upsilon(t))dt \le V(x(0), 0) \tag{35}$$

By chain rule, we can get

$$\frac{dV(x(t), t)}{dt} = (\frac{\partial V(x(t), t)}{\partial x(t)})^T (F(x(t), t) + G(x(t), t)u(t) + D(x(t), t)\upsilon(t)) + (\frac{\partial V(x(t), t)}{\partial t}) \tag{36}$$

By integrating the above equation from $t = 0$ to $t = \infty$, we can obtain:

$$\begin{aligned} &V(x(\infty), \infty) - V(x(0), 0) \\ &= \int_0^\infty (\frac{\partial V(x(t), t)}{\partial x(t)})^T (F(x(t), t) + G(x(t), t)u(t) \\ &\quad + D(x(t), t)\upsilon(t)) + (\frac{\partial V(x(t), t)}{\partial t})dt \end{aligned} \tag{37}$$

By adding the right hand side of (37), and then extracting the left hand side of (37), we get

$$\begin{aligned} &\int_0^\infty [x^T(t)Qx(t) + u^T(t)Ru(t) - \rho\upsilon^T(t)\upsilon(t)]dt \\ &= \int_0^\infty (\frac{\partial V(x(t), t)}{\partial x(t)})^T [F(x(t), t) + G(x(t), t)u(t) \\ &\quad + D(x(t), t)\upsilon(t)] + (\frac{\partial V(x(t), t)}{\partial t}) \\ &\quad + [x^T(t)Qx(t) + u^T(t)Ru(t) - \rho\upsilon^T(t)\upsilon(t)]dt \\ &\quad - V(x(\infty), \infty) + V(x(0), 0) \end{aligned} \tag{38}$$

Completing the square for $u(t)$ and $\upsilon(t)$ on the right hand side of (38), we have

$$\begin{aligned} &\int_0^\infty [x^T(t)Qx(t) + u^T(t)Ru(t) - \rho\upsilon^T(t)\upsilon(t)]dt \\ &= \int_0^\infty [H(x(t), t) + (u(t) + \frac{1}{2}R^{-1}G^T(x(t), t) \\ &\quad \times (\frac{\partial V(x(t), t)}{\partial x(t)}))^T R(u(t) + \frac{1}{2}R^{-1}G^T(x(t), t)(\frac{\partial V(x(t), t)}{\partial x(t)})) \\ &\quad - \rho(\upsilon(t) - \frac{1}{2\rho}D^T(x(t), t)(\frac{\partial V(x(t), t)}{\partial x(t)}))^T \\ &\quad \times (\upsilon(t) - \frac{1}{2\rho}D^T(x(t), t)(\frac{\partial V(x(t), t)}{\partial x(t)}))]dt \\ &\quad - V(x(\infty), \infty) + V(x(0), 0) \end{aligned} \tag{39}$$

where $H(x(t), t)$ is denoted as

$$\begin{aligned} H(x(t),t) &= (\frac{\partial V(x(t),t)}{t}) + \left(\frac{\partial V(x(t),t)}{\partial x(t)}\right)^T F(x(t),t) \\ &\quad - \frac{1}{4}\left(\frac{\partial V(x(t),t)}{\partial x(t)}\right)^T G(x(t),t)R^{-1}G^T(x(t),t) \\ &\quad \times \left(\frac{\partial V(x(t),t)}{\partial x(t)}\right) + x^T(t)Qx(t) \\ &\quad + \frac{1}{4\rho}\left(\frac{\partial V(x(t),t)}{\partial x(t)}\right)^T D(x(t),t) \\ &\quad \times D^T(x(t),t) \times (\frac{\partial V(x(t),t)}{\partial x(t)}) \end{aligned}$$

which is the HJIE in (4). Then, if the HJIE in (4) holds, we perform min max in (35) as the following

$$\begin{aligned} &\min_{u(t)}\max_{v(t)} \int_0^\infty [x^T(t)Qx(t) + u^T(t)Ru(t) - \rho v^T(t)v(t)]dt \\ &= \min_{u(t)}\max_{v(t)} \int_0^\infty [(u(t) + \frac{1}{2}R^{-1}G^T(x,t)(\frac{\partial V(x(t),t)}{\partial x(t)}))^T R \\ &\quad \times (u(t) + \frac{1}{2}R^{-1}G^T(x,t)(\frac{\partial V(x(t),t)}{\partial x(t)})) - \rho(v(t) \\ &\quad - \frac{1}{2\rho}D^T(x,t)\frac{\partial V(x(t),t)}{\partial x(t)}))^T(v(t) \\ &\quad - \frac{1}{2\rho}D^T(x,t)\frac{\partial V(x(t),t)}{\partial x(t)}))]dt \\ &\quad - V(x(\infty),\infty) + V(x(0),0) \\ &= -V(x(\infty),\infty) + V(x(0),0) \end{aligned} \tag{40}$$

with the min max strategies $u^*(t)$ and $v^*(t)$ should be obtained as (3). From (40), we can obtain

$$\begin{aligned} &\min_{u(t)}\max_{v(t)} \int_0^\infty [x^T(t)Qx(t) + u^T(t)Ru(t) - \rho v^T(t)v(t)]dt \\ &\quad = -V(x(\infty),\infty) + V(x(0),0) \le V(x(0),0) \end{aligned} \tag{41}$$

by the fact $V(x(\infty),\infty) \ge 0$, which is (35).

(b) If $v(t) = 0$, then (41) becomes the following H_2 quadratic optimal

$$\min_{u(t)} \int_0^\infty [x^T(t)Qx(t) + u^T(t)Ru(t)]dt \le V(x(0),0) \tag{42}$$

i.e., the minmax H_∞ control strategy will become the H_2 quadratic optimal control strategy as $v(t) = 0$. Since $V(x(0),0)$ is finite in (42), it is clear that as t approach to infinite, both $x(t)$ and $u(t)$ will approach to zero, i.e., if the nonlinear time-varying system in (1) is free of disturbance, then the H_∞ control $u^*(t)$ in (3) will achieve the asymptotic stability. Q.E.D

APPENDIX B
PROOF OF THEOREM 2

Suppose

$$(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)})_\varepsilon = (\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)}) + h(\bar{x}(t)) \tag{43}$$

where $h(\bar{x}(t))$ is the error function. Subtracting (23) with (22), $\varepsilon(\theta_i(t))$ can be rewritten as:

$$\begin{aligned} &\varepsilon(\theta_i(t)) \\ &= HJIE_\varepsilon - HJIE \\ &= ((\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)})_\varepsilon^T - (\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)})^T)\bar{F}(\bar{x}(t)) \\ &\quad - \frac{1}{4}(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)})_\varepsilon^T \bar{G}(\bar{x}(t))\bar{R}^{-1}\bar{G}^T(\bar{x}(t))(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)})_\varepsilon \\ &\quad + \frac{1}{4\rho}(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)})_\varepsilon^T \bar{D}(\bar{x}(t))\bar{D}^T(\bar{x}(t))(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)})_\varepsilon \\ &\quad + \frac{1}{4}(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)})^T \bar{G}(\bar{x}(t))\bar{R}^{-1}\bar{G}^T(\bar{x}(t))(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)}) \\ &\quad - \frac{1}{4\rho}(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)})^T \bar{D}(\bar{x}(t))\bar{D}^T(\bar{x}(t))(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)}) \\ &= h^T(\bar{x}(t))\bar{F}(\bar{x}(t)) \\ &\quad - \frac{1}{4}(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)})^T \bar{G}(\bar{x}(t))\bar{R}^{-1}\bar{G}^T(\bar{x}(t))h(\bar{x}(t)) \\ &\quad - \frac{1}{4}h^T(\bar{x}(t))\bar{G}(\bar{x}(t))\bar{R}^{-1}\bar{G}^T(\bar{x}(t))(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)}) \\ &\quad - \frac{1}{4}h^T(\bar{x}(t))\bar{G}(\bar{x}(t))\bar{R}^{-1}\bar{G}^T(\bar{x}(t))h(\bar{x}(t)) \\ &\quad + \frac{1}{4\rho}(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)})^T \bar{D}(\bar{x}(t))\bar{D}^T(\bar{x}(t))h(\bar{x}(t)) \\ &\quad + \frac{1}{4\rho}h^T(\bar{x}(t))\bar{D}(\bar{x}(t))\bar{D}^T(\bar{x}(t))(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)}) \\ &\quad + \frac{1}{4\rho}h^T(\bar{x}(t))\bar{D}(\bar{x}(t))D^T(\bar{x}(t))h(\bar{x}(t)) \end{aligned} \tag{44}$$

By the symmetric property, the following equations hold:

$$\begin{aligned} &(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)})^T \bar{D}(\bar{x}(t))\bar{D}^T(\bar{x}(t))h(\bar{x}(t)) \\ &\quad = h^T(\bar{x}(t))\bar{D}(\bar{x}(t))\bar{D}^T(\bar{x}(t))(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)}) \\ &(\frac{\partial \bar{V}(\bar{x})}{\partial \bar{x}(t)})^T \bar{G}(\bar{x}(t))\bar{R}^{-1}\bar{G}^T(\bar{x}(t))h(\bar{x}(t)) \\ &\quad = h^T(\bar{x}(t))\bar{G}(\bar{x}(t))\bar{R}^{-1}\bar{G}^T(\bar{x}(t))(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)}) \end{aligned} \tag{45}$$

Then, by the above equations, (44) can be written as:

$$\begin{aligned} &\varepsilon(\theta_i(t)) \\ &= h^T(\bar{x}(t))\bar{F}(\bar{x}(t)) \\ &\quad - \frac{1}{2}h^T(\bar{x}(t))\bar{G}(\bar{x}(t))\bar{R}^{-1}\bar{G}^T(\bar{x}(t))(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)}) \\ &\quad - \frac{1}{4}h^T(\bar{x}(t))\bar{G}(\bar{x}(t))\bar{R}^{-1}\bar{G}^T(\bar{x}(t))h(\bar{x}(t)) \\ &\quad + \frac{1}{2\rho}h^T(\bar{x}(t))\bar{D}(\bar{x}(t))\bar{D}^T(\bar{x}(t))(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)}) \\ &\quad + \frac{1}{4\rho}h^T(\bar{x}(t))\bar{D}(\bar{x}(t))\bar{D}^T(\bar{x}(t))h(\bar{x}(t)) \\ &= h^T(\bar{x})[\bar{F}(\bar{x}) - \frac{1}{2}\bar{G}(\bar{x})\bar{R}^{-1}\bar{G}^T(\bar{x})(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)}) \\ &\quad - \frac{1}{4}\bar{G}(\bar{x})\bar{R}^{-1}\bar{G}^T(\bar{x})h(\bar{x}) + \frac{1}{2\rho}\bar{D}(\bar{x})\bar{D}^T(\bar{x})(\frac{\partial \bar{V}(\bar{x})}{\partial \bar{x}}) \\ &\quad + \frac{1}{4\rho}\bar{D}(\bar{x})\bar{D}^T(\bar{x})h(\bar{x})] \end{aligned} \tag{46}$$

If $\varepsilon(\theta_i(t)) \to 0$ in (46), then we have

$$h^T(\bar{x})[\bar{F}(\bar{x}(t)) - \frac{1}{2}\bar{G}(\bar{x}(t))\bar{R}^{-1}\bar{G}^T(\bar{x}(t))(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)}) - \frac{1}{4}\bar{G}(\bar{x}(t))\bar{R}^{-1}\bar{G}^T(\bar{x}(t))h(\bar{x}(t)) + \frac{1}{2\rho}\bar{D}(\bar{x}(t))\bar{D}^T(\bar{x}(t))(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)}) + \frac{1}{4\rho}\bar{D}(\bar{x}(t))\bar{D}^T(\bar{x}(t))h(\bar{x}(t))] \to 0 \quad (47)$$

Clearly, the term

$$[\bar{F}(\bar{x}(t)) - \frac{1}{2}\bar{G}(\bar{x}(t))\bar{R}^{-1}\bar{G}^T(\bar{x}(t))(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)}) - \frac{1}{4}\bar{G}(\bar{x}(t))\bar{R}^{-1}\bar{G}^T(\bar{x}(t))h(\bar{x}(t)) + \frac{1}{2\rho}\bar{D}(\bar{x}(t))\bar{D}^T(\bar{x}(t))(\frac{\partial \bar{V}(\bar{x})}{\partial \bar{x}}) + \frac{1}{4\rho}\bar{D}(\bar{x}(t))\bar{D}^T(\bar{x}(t))h(\bar{x}(t))] \quad (48)$$

in (47) is different from HJIE $= 0$ in (22) and it will not be zero for all $\bar{x}(t)$. As a result,, we can conclude that $h(\bar{x}(t)) \to 0$ and it implies $(\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)})_\varepsilon \to (\frac{\partial \bar{V}(\bar{x}(t))}{\partial \bar{x}(t)})$, i.e., $HJIE_\epsilon \to HJIE = 0$ as $\varepsilon(\theta_i(t)) \to 0$. According to Theorem 1, the HJIE-embedded DNN-based H_∞ control $u^*(t)$ in Fig. 1 will approach H_∞ control strategy $u^*(t)$ in (3) in Theorem 1. Q.E.D.

REFERENCES

[1] A. L. Maas, P. Qi, Z. Xie, A. Y. Hannun, C. T. Lengerich, D. Jurafsky, and A. Y. Ng, "Building DNN acoustic models for large vocabulary speech recognition," *Comput. Speech Lang.*, vol. 41, pp. 195–213, Jan. 2017.
[2] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 1, pp. 30–42, Jan. 2012.
[3] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1285–1298, May 2016.
[4] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," *IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, Rhode Island, 2012, pp. 3642–3649.
[5] S. P. Singh, A. Kumar, H. Darbari, L. Singh, A. Rastogi, and S. Jain, "Machine translation using deep learning: An overview," in *Proc. Int. Conf. Comput., Commun. Electron. (Comptelix)*, Jaipur, India, Jul. 2017, pp. 162–167.
[6] A. E. Eshratifar, A. Esmaili, and M. Pedram, "BottleNet: A deep learning architecture for intelligent mobile cloud computing services," in *Proc. IEEE/ACM Int. Symp. Low Power Electron. Design (ISLPED)*, Lausanne, Switzerlan, Jul. 2019, pp. 1–6.
[7] C. Deng, S. Liao, Y. Xie, K. K. Parhi, X. Qian, and B. Yuan, "PermDNN: Efficient compressed DNN architecture with permuted diagonal matrices," in *Proc. 51st Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Fukuok, Japan, Oct. 2018, pp. 189–202.
[8] M. Zahangir Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C Van Esesn, A. A S. Awwal, and V. K. Asari, "The history began from AlexNet: A comprehensive survey on deep learning approaches," Mar. 2018, *arXiv:1803.01164*. [Online]. Available: http://arxiv.org/abs/1803.01164
[9] J. Yin, S. Su, J. Xun, T. Tang, and R. Liu, "Data-driven approaches for modeling train control models: Comparison and case studies," *ISA Trans.*, vol. 98, pp. 349–363, Mar. 2020.
[10] B.-S. Chen, C.-S. Wu, and H.-J. Uang, "A minimax tracking design for wheeled vehicles with trailer based on adaptive fuzzy elimination scheme," *IEEE Trans. Control Syst. Technol.*, vol. 8, no. 3, pp. 418–434, May 2000.
[11] A. J. van de Schaft, "L_2 gain analysis of nonlinear systems and nonlinear state feedback H_∞ control," *IEEE Trans. Automatica*, vol. 37, no. 6, pp. 770–784, Jun. 1992.
[12] K. S. Narendra and J. H. Taylor, "Lyapunov functions for nonlinear time-varying systems," *Inf. Control*, vol. 12, no. 5, pp. 378–393, May 1968.
[13] S. Guo and L. Han, *Stability and Control of Nonlinear Time-Varying Systems*. Singapore: Springer, 2018.
[14] G. Kenné, T. Ahmed-Ali, F. Lamnabhi-Lagarrigue, and A. Arzandé, "Nonlinear systems time-varying parameter estimation: Application to induction motors," *Electr. Power Syst. Res.*, vol. 78, no. 11, pp. 1881–1888, Nov. 2008.
[15] J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis, "State-space solutions to standard H_2 and H_∞ control problems," *IEEE Trans. Autom. Control*, vol. 34, no. 8, pp. 831–847, Aug. 1989.
[16] P. P. Khargonekar and M. A. Rotea, "Mixed H_2/H_∞ control: A convex optimization approach," *IEEE Trans. Autom. Control*, vol. 36, no. 7, pp. 824–837, Jul. 1991.
[17] M. Herrera, W. Chamorro, A. P. Gomez, and O. Camacho, "Sliding mode control: An approach to control a quadrotor," in *Proc. Asia–Pacific Conf. Comput. Aided Syst. Eng.*, Quito, Ecuador, 2015, Jul. 2015, pp. 314–319.
[18] H. Lee and Y. Kim, "Fault-tolerant control scheme for satellite attitude control system," *IET Control Theory Appl.*, vol. 4, no. 8, pp. 1436–1450, Aug. 2010.
[19] N. Sakamoto and A. J. van der Schaft, "Analytical approximation methods for the stabilizing solution of the Hamilton–Jacobi equation," *IEEE Trans. Autom. Control*, vol. 53, no. 10, pp. 2335–2350, Nov. 2008.
[20] G. Feng, "A survey on analysis and design of model-based fuzzy control systems," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 5, pp. 676–697, Oct. 2006.
[21] W.-H. Chen and B.-S. Chen, "Robust stabilization design for nonlinear stochastic system with Poisson noise via fuzzy interpolation method," *Fuzzy Sets Syst.*, vol. 217, pp. 41–61, Apr. 2013.
[22] B. Chen, W. Chen, and H. Wu, "Robust H_2/H_∞ global linearization filter design for nonlinear stochastic systems," *IEEE Trans. Circuits Syst. I, Reg. Papers: Reg. Papers*, vol. 56, no. 7, pp. 1441–1454, Jul. 2009.
[23] K. A. Hoo and J. C. Kantor, "Global linearization and control of a mixed-culture bioreactor with competition and external inhibition," *Math. Biosci.*, vol. 82, no. 1, pp. 43–62, Nov. 1986.
[24] K. Jouili and H. Jerbi, "A computationally Lyapunov nonlinear gain scheduling control of nonlinear systems with stability guarantees," in *Proc. 11th Int. Conf. Comput. Modeling Simulation*, vol. 12, Oct. 2009, pp. 1131–1137.
[25] R. A. Nichols, R. T. Reichert, and W. J. Rugh, "Gain scheduling for H-infinity controllers: A flight control example," *IEEE Trans. Control Syst. Technol.*, vol. 1, no. 2, pp. 69–79, Jun. 1993.
[26] B.-S. Chen, C.-S. Tseng, and H.-J. Uang, "Robustness design of nonlinear dynamic systems via fuzzy linear control," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 5, pp. 571–585, Oct. 1999.
[27] B.-S. Chen, C.-P. Wang, and M.-Y. Lee, "Stochastic robust team tracking control of multi-UAV networked system under Wiener and Poisson random fluctuations," *IEEE Trans. Cybern.*, early access, Jan. 10, 2020, doi: 10.1109/TCYB.2019.2960104.
[28] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," Nov. 2015, *arXiv:1505.00853*. [Online]. Available: http://arxiv.org/abs/1505.00853
[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, San Diego, CA, USA, 2014, pp. 1–15. [Online]. Available: http://arxiv.org/abs/1412.6980
[30] Y. N. Dauphin, H. De Vries, and Y. Bengio, "Equilibrated adaptive learning rates for non-convex optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, no. 3, 2015, pp. 1504–1512.
[31] B. Zhao, B. Xian, Y. Zhang, and X. Zhang, "Nonlinear robust adaptive tracking control of a quadrotor UAV via immersion and invariance methodology," *IEEE Trans. Ind. Electron.*, vol. 62, no. 5, pp. 2891–2902, May 2015.
[32] J. O. Pedro and P. J. Kala, "Nonlinear control of quadrotor UAV using Takagi-Sugeno fuzzy logic technique," in *Proc. 10th Asian Control Conf. (ASCC)*, May 2015, pp. 1–6.
[33] F. Kendoul, Z. Yu, and K. Nonami, "Guidance and nonlinear control system for autonomous flight of minirotorcraft unmanned aerial vehicles," *J. Field Robot.*, vol. 27, no. 3, pp. 311–334, May 2010.

[34] B.-S. Chen, W.-Y. Chen, C.-T. Yang, and Z. Yan, "Noncooperative game strategy in cyber-financial systems with Wiener and Poisson random fluctuations: LMIs-constrained MOEA approach," *IEEE Trans. Cybern.*, vol. 48, no. 12, pp. 3323–3336, Dec. 2018.
[35] C. Ren, S. He, X. Luan, F. Liu, and H. R. Karimi, "Finite-time L_2-gain asynchronous control for continuous-time positive hidden Markov jump systems via T-S fuzzy model approach," *IEEE Trans. Cybern.*, vol. 51, no. 1, pp. 77–87, Jan. 2021.
[36] C. Ren and S. He, "Finite-time stabilization for positive Markovian jumping neural networks," *Appl. Math. Comput.*, vol. 365, Jan. 2020, Art. no. 124631.
[37] C. Ren, R. Nie, and S. He, "Finite-time positiveness and distributed control of Lipschitz nonlinear multi-agent systems," *J. Franklin Inst.*, vol. 356, no. 15, pp. 8080–8092, Oct. 2019.
[38] W. Deng and J. Yao, "Extended-state-observer-based adaptive control of electrohydraulic servomechanisms without velocity measurement," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 3, pp. 1151–1161, Jun. 2020.
[39] W. Deng, J. Yao, and D. Ma, "Time-varying input delay compensation for nonlinear systems with additive disturbance: An output feedback approach," *Int. J. Robust Nonlinear Control*, vol. 28, no. 1, pp. 31–52, May 2017.
[40] X. H. Chang, C. Yang, and J. Xiong, "Quantized fuzzy output feedback H_∞ control for nonlinear systems with adjustment of dynamic parameters," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 10, pp. 2005–2015, Sep. 2018.
[41] X. H. Chang, L. Zhang, and J. H. Park, "Robust static output feedback H_∞ control for uncertain fuzzy systems," *Fuzzy Sets Syst.*, vol. 273, pp. 87–104, Aug. 2015.
[42] W. M. Haddad and V. Kapila, "Robust stabilization for continuous-time systems with slowly time-varying uncertain real parameters," *IEEE Trans. Autom. Control*, vol. 43, no. 7, pp. 987–992, Jul. 1998.
[43] Z. Chen, F. Huang, W. Sun, J. Gu, and B. Yao, "RBF-neural-network-based adaptive robust control for nonlinear bilateral teleoperation manipulators with uncertainty and time delay," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 2, pp. 906–918, Apr. 2020.
[44] Z. Chen, F. H. Huang, W. J. Chen, J. H. Zhang, W. C. Sun, J. W. Chen, J. Gu, and S. Q. Zhu, "RBFNN-based adaptive sliding mode control design for delayed nonlinear multilateral telerobotic system with cooperative manipulation," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 906–918, Feb. 2020.
[45] V.-P. Vu and T. D. Do, "Fault/state estimation observer synthesis for uncertain T-S fuzzy systems," *IEEE Access*, vol. 7, pp. 358–369, 2019.

BOR-SEN CHEN (Life Fellow, IEEE) received the B.S. degree in electrical engineering from the Tatung Institute of Technology, Taipei, Taiwan, in 1970, the M.S. degree in geophysics from National Central University, Taiwan, in 1973, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 1982. He has been a Lecturer, an Associate Professor, and a Professor with the Tatung Institute of Technology, from 1973 to 1987. He is currently the Tsing Hua Distinguished Chair Professor of Electrical Engineering and Computer Science with National Tsing Hua University, Hsinchu, Taiwan. His current research interests include control engineering, signal processing, and systems biology. He received the Distinguished Research Award from the National Science Council of Taiwan four times. He is also a National Chair Professor of the Ministry of Education of Taiwan.

MIN-YEN LEE received the B.S. degree in bioindustrial mechatronics engineering from National Chung Hsing University, Taichung City, Taiwan, in 2014. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan. His current research interests include robust control, fuzzy control, and nonlinear stochastic systems.

TZU-HAN LIN received the B.S. degree in electrical engineering from National Chiao Tung University, Hsinchu, Taiwan, in 2018. She is currently pursuing the master's degree in electrical engineering with National Tsing Hua University, Hsinchu.