Received November 19, 2021, accepted December 4, 2021, date of publication December 8, 2021,
date of current version December 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3133899

Robust H_∞ Deep Neural Network-Based Filter Design of Nonlinear Stochastic Signal Systems

BOR-SEN CHEN[1,2], (Life Fellow, IEEE), PO-HSUN WU[1], AND MIN-YEN LEE[1]
[1]Department of Electrical Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan
[2]Department of Electrical Engineering, Yuan Ze University, Chung-Li, 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 Recently, deep neural network (DNN) schemes based on big data-driven methods have been successfully applied to image classification, communication, translation of language, speech recognition, etc. However, more efforts are still needed to apply them to complex robust nonlinear filter design in signal processing, especially for the robust nonlinear H_∞ filter design for robust state estimation of nonlinear stochastic signal system under uncertain external disturbance and output measurement noise. In general, the design problem of robust nonlinear H_∞ filter needs to solve a complex Hamilton-Jacobi-Isaacs equation (HJIE), which is not easily solved analytically or numerically. Further, the robust nonlinear H_∞ filter is not easily designed by training DNN directly via conventional big data schemes. In this paper, a novel robust H_∞ HJIE-embedded DNN-based filter design is proposed as a co-design of H_∞ filtering algorithm and DNN learning algorithm for the robust state estimation of nonlinear stochastic signal systems with external disturbance and output measurement noise. In the proposed robust H_∞ DNN-based filter design, we have proven that when the approximation error of HJIE by the trained DNN through Adam learning algorithm approaches to 0, the HJIE-embedded DNN-based filter will approach the robust nonlinear H_∞ filter of nonlinear stochastic signal system with uncertain external disturbance and output measurement noise. Finally, a trajectory estimation problem of 3-D geometry incoming nonlinear stochastic missile system by the proposed robust H_∞ HJIE-embedded DNN-based filter scheme through the measurement by the sensor of radar system with external disturbance and measurement noise is given to illustrate the design procedure and validate its robust H_∞ filtering performance when compared with the extended Kalman filter and particle filter.

INDEX TERMS Deep neural network (DNN), robust H_∞ filter, nonlinear stochastic signal system, extended Kalman filter, particle filter, Hamilton-Jacobi Isaacs equation (HJIE), co-design of H_∞ filtering, and deep neural network learning.

I. INTRODUCTION

In the last decades, the robust H_∞ filter has been widely applied to the field of signal processing when the statistical information of external disturbance and measurement noise are unavailable and their worst-case effects on the filtering error need to be efficiently attenuated. Therefore the robust H_∞ filter is always used to estimate the state or detect the signal in the signal transmission systems corrupted with uncertain external disturbance and random measurement noise [1], [2]. In the linear stochastic signal systems, the H_∞ filter design needs to solve a Riccati-like equation or a corresponding linear matrix inequality (LMI) for filter gain [3]. In many practical signal processings, stochastic signal transmission systems always have nonlinear dynamic behavior with uncertain external disturbance and output random measurement noise [4]–[6]. However, nonlinear H_∞ filter design problem for the state estimation of nonlinear stochastic signal system needs to solve a highly nonlinear filter-system state-coupled partial differential Hamilton-Jacobi-Isaacs equation (HJIE), which can not be efficiently solved at present [3], [7].

The conventional extended Kalman filter [8] and uncented Kalman filter [9], which are based on the linearization at the predicted state of previous stage, are not easily employed to treat this nonlinear optimal filter design problem of nonlinear stochastic signal system with uncertain external disturbance and output measurement noise [10]. The reason is that their

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

filtering performance will be degraded by the error propagation due to iterative linearization and unknown statistics of external disturbance and output measurement noise. The particle filter (PF) [11], [12] is suitable for the filtering design of nonlinear stochastic signal systems with given noise statistics but unsuitable for filtering design problem under uncertain external disturbances and measurement noise with unknown or uncertain statistics. In order to avoid solving a complex nonlinear partial differential HJIE for the minmax robust H_∞ filter design of nonlinear stochastic signal systems, several interpolation methods such as fuzzy interpolation method [13]–[15], global linearization method [16] and gain scheduling method [17] have been employed to interpolate several local linearized systems to approximate the nonlinear stochastic signal system with external disturbance and measurement noise so that HJIE of robust nonlinear H_∞ filter design could be interpolated by a set of local Riccati-like equations or a set of equivalent local linear matrix inequalities (LMIs).

For example, the T-S fuzzy interpolation method is used to interpolate several local linearized systems by some fuzzy bases to approximate nonlinear stochastic signal system [13]–[15], [18]–[20]. Then a T-S fuzzy filter is employed to interpolate these local linear filters by some fuzzy bases to achieve robust nonlinear H_∞ filtering performance by solving the corresponding LMIs. Besides, the global linearization scheme employs some smooth interpolation functions to interpolate several local linearized systems at the vertices of the convex hull of polytope Ω of all linearized systems of a nonlinear stochastic signal systems to approach the nonlinear stochastic signal system [16], [21]. Then, a global linearization filter can be designed by interpolating local linear filters by some smooth functions through solving a set of LMIs to achieve the robust H_∞ filtering scheme [16], [20]–[21]. However, these interpolation schemes need to solve a large number of LMIs. For example, if n local linearized systems are interpolated to approximate a nonlinear stochastic signal system, it needs to solve n^2 LMIs for nonlinear robust H_∞ filtering design [13]–[15], [18]–[19]. Therefore, for a highly nonlinear stochastic signal system, it will increase the design complexity of nonlinear robust H_∞ filter in the design procedure and the computational complexity of robust H_∞ interpolation filter at every time step. Furthermore, these interpolation filter design methods of local linearized systems are based on the fact that the solution of HJIE is limited to the form of the quadratic Lyapunov function $V(\tilde{x}(t)) = \tilde{x}^T(t)P\tilde{x}(t)$ for some $P^T = P > 0$, where $\tilde{x}(t)$ denotes the filtering (estimating) error. This will lead to a conservative solution of nonlinear HJIE. Since the state $x(t)$ of signal system is unavailable and is to be estimated in the filter design problem, the interpolation functions need to base on the estimated state $\hat{x}(t)$, it will increase the design difficulty and filtering accuracy, especially at the begining of estimation when a large initial estimation error.

In this study, as shown in Fig. 1, a robust H_∞ deep neural network (DNN)-based filter scheme is proposed as a co-design of H_∞ filter scheme and DNN learning algorithm for the robust state estimation of nonlinear stochastic signal system by solving $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ via the output of DNN for the H_∞ filter gain $k^*(\hat{x}(t))$ from HJIE directly. The design procedure of H_∞ DNN-based filter is divided into two phases, i.e., (i) off-line training phase and (ii) on-line operation phase. In the off-line training phase, the proposed HJIE-embedded DNN in Fig. 1 is to be trained by Adam learning algorithm to output $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ at time t via solving HJIE of the nonlinear robust H_∞ filtering design problem, and $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ can be used to produce H_∞ filter gain $k^*(\hat{x}(t))$, the worst-case external disturbance $v^*(t)$ and output measurement noise $n^*(t)$ simultaneously for the next step off-line training as shown in Fig. 1. However, the most difficulty to solve HJIE by training DNN to output $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ is that HJIE is the function of $\hat{x}(t), \tilde{x}(t)$ and $x(t)$, where $x(t)$ is unavailable and to be estimated. Therefore, the H_∞ filter gain $k^*(\hat{x}(t))$, the worst-case external disturbance $v^*(t)$ and output measurement noise $n^*(t)$ will be sent to nonlinear stochastic signal system, Luenberger-type filter and estimation error system to replace unavailable $v(t)$ and $n(t)$ to generate $y(t)$, $\hat{x}(t)$, $\tilde{x}(t)$ and $x(t) = \hat{x}(t) + \tilde{x}(t)$, respectively, for the next training step at time t in the off-line training phase. The replacement of unavailable external disturbance $v(t)$ and measurement noise $n(t)$ by the worst-case $v^*(t)$ and $n^*(t)$ in the H_∞ filtering scheme does not influence on the H_∞ filtering performance because the robust H_∞ filtering scheme is designed based on the worst-case $v^*(t)$ and $n^*(t)$ instead of $v(t)$ and $n(t)$. After off-line training phase of robust H_∞ DNN-based filtering scheme, it will be shifted to on-line operation phase of H_∞ DNN-based filter scheme. In the on-line operation phase, $y(t)$ is generated by real stochastic signal system with real $v(t)$ and $n(t)$. We have proven that as the error $\epsilon(\theta(t))$ of solving HJIE approaches to zero by training DNN through Adam learning algorithm, the output of HJIE-embedded DNN will approach to $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ of HJIE of the robust H_∞ filtering strategy and the proposed HJIE-embedded DNN-based filter will approach to the real robust H_∞ filter of nonlinear stochastic signal system. However, in practical applications, we will stop the Adam learning algorithm in the training phase when the absolute HJIE error $|\epsilon(\theta(t))| \leq \delta$ for some precribed small positive value δ.

The contributions of this paper are described as follows:

1) A novel HJIE-embedded DNN-based filter scheme is proposed for the co-design of H_∞ filtering scheme with DNN learning algorithm for the robust state estimation of nonlinear stochastic signal system with uncertain external disturbance and output measurement noise. The worst-case external disturbance $v^*(t)$ and output measurement noise $n^*(t)$ are employed in the off-line training phase to replace $v(t)$ and $n(t)$ to generate output $y(t)$ by nonlinear signal model, $\hat{x}(t)$ by the filter model, and $\tilde{x}(t)$ by filter error model and $x(t) = \hat{x}(t) + \tilde{x}(t)$ so

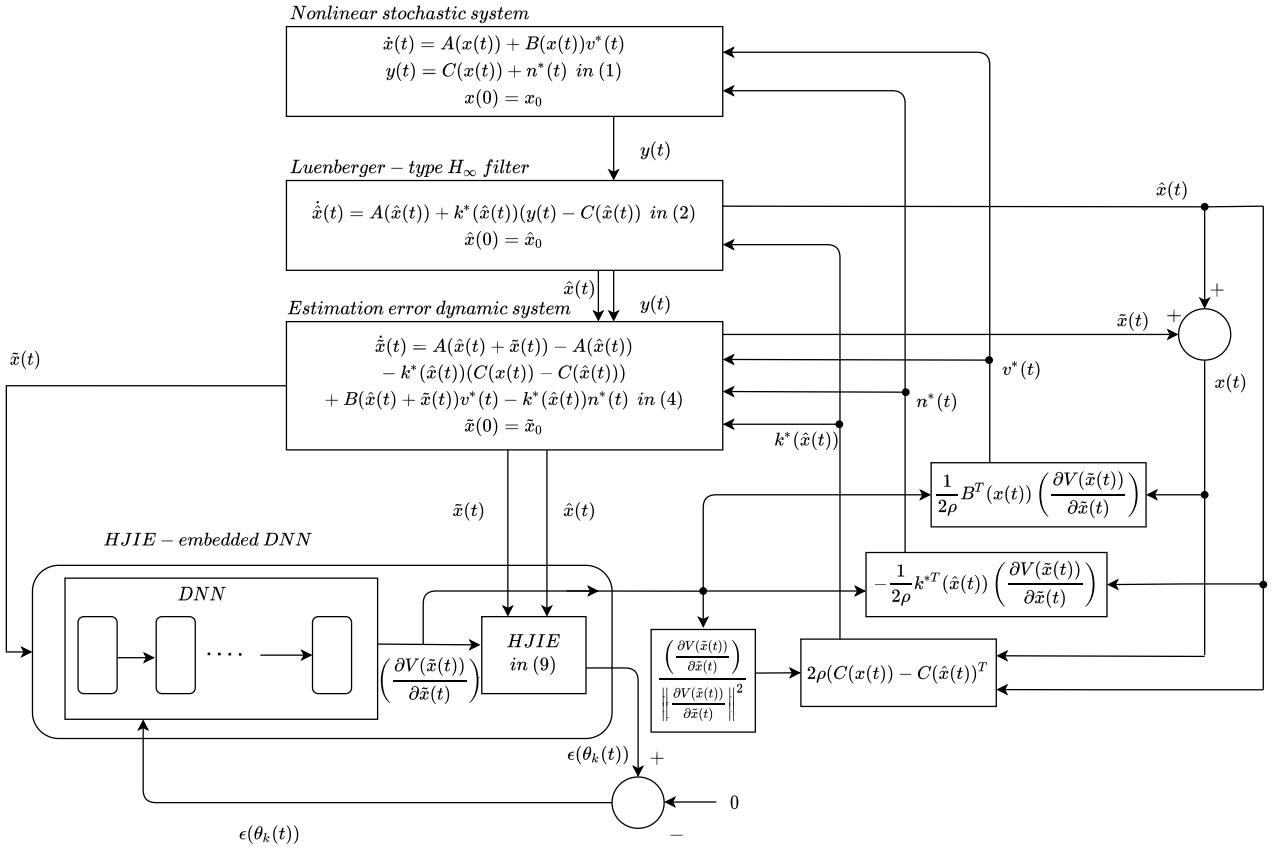

FIGURE 1. The flow chart of robust H_∞ HJIE-embedded DNN-based filter scheme of nonlinear stochastic signal system in off-line training phase. It is a co-design of the conventional H_∞ Luenberger-type filtering scheme and an HJIE-embedded DNN learning scheme to be trained via Adam learning algorithm for the robust state estimation of nonlinear stochastic signal system in (1) with uncertain external disturbance and output measurement noise. Since $x(t)$ is unavailable in the off-line training phase, we need to embed an estimation error system in (4) to generate $\tilde{x}(t)$ so that we could obtain $x(t) = \tilde{x}(t) + \hat{x}(t)$ for calculating $A(x(t))$, $B(x(t))$ and $C(x(t))$ in HJIE. Further $\tilde{x}(t)$ is inputed into HJIE-embedded DNN to output $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ to generate the H_∞ observer gain $k^*(t)$, the worst-case external disturbance $v^*(t)$ and measurement noise $n^*(t)$ as well as to solve $HJIE = 0$. If $HJIE = \epsilon(\theta_k(t)) \neq 0$, $\epsilon(\theta_k(t))$ is fedback to train DNN by Adam learning algorithm. In the off-line operation phase, if $\epsilon(\theta_k(t)) \to 0$, we show that the output of DNN approaches to $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ and we will shift to on-line operation phase. In the on-line operation phase, $y(t)$ is generated by real $v(t)$ and $n(t)$. In general, we do not need to train DNN in on-line operation phase. However, if $|\epsilon(\theta_k(t))| > \delta$ for some prescribed small value $\delta > 0$. We could feedback $\epsilon(\theta_k(t))$ to train DNN in the on-line operation phase without influence on the H_∞ filtering performance.

that the $x(t)$-$\hat{x}(t)$-$\tilde{x}(t)$-coupled nonlinear HJIE could be solved by HJIE-embedded DNN for H_∞ DNN-based filter design. Adam learning algorithm is employed to efficiently train the weighting parameters of DNN to output $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ to solve nonlinear partial differential HJIE directly for the robust H_∞ filtering design of nonlinear stochastic signal system. We also prove that when the error of solving $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ from HJIE approaches to 0 in the DNN training process, the proposed HJIE-embedded DNN-based filter scheme can approach to the robust H_∞ filter design of nonlinear stochastic signal systems.

2) Instead of solving $V(\tilde{x}(t))$ from HJIE of the robust H_∞ filter design of nonlinear stochastic signal system in the conventional design methods, the proposed HJIE-embedded DNN directly solves $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ from HJIE to obtain the robust H_∞ filter gain $k^*(\hat{x}(t))$ directly, which can avoid the very difficult calculation work of $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ from the numerical values of $V(\tilde{x}(t))$ and $\tilde{x}(t)$ in the conventional design method.
3) The proposed robust H_∞ DNN-based filtering scheme can provide a co-design of H_∞ filtering scheme and deep learning algorithm to bridge the gap between the traditional theoretical nonlinear robust H_∞ filter theory and recent pure big data-driven deep learning schemes to facilitate the application of DNN schemes to treat the complex nonlinear H_∞ filter design problem in nonlinear stochastic signal systems to save much training data and computation time of the conventional big data-driven deep learning filter design method.

The remainder of this study is organized as follows: In Section II, the problem description is given. The theoretical robust H_∞ filter gain and the worst-case external disturbance and measurement noise are solved based on $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ of HJIE. The robust H_∞ HJIE-embedded DNN-based filter scheme is

proposed through the Adam learning algorithm in Section III. It is also shown that the proposed robust H_∞ DNN-based filter scheme will approach to theoretical robust H_∞ filter if the approximation error of HJIE approximated by DNN via the off-line training through Adam learning algorithm converges to 0. A simulation example of the state estimation of incoming ballistic missile by the sensor of radar is given to illustrate the design procedure and validate the performance of proposed H_∞ DNN-based filter is given in Section IV. Finally, a conclusion is given in Section V.

Notation: $\mathbb{R}^n$: The set of n-tuple real vectors. $\mathbb{R}^{n\times m}$: The set of real $n \times m$ matrices. $x(t) \in \mathbb{R}^n$, then $\|x(t)\| = \left(\sum_{i=1}^n x_i^2(t)\right)^{1/2} = \left(x^T(t)x(t)\right)^{1/2}$, where $x^T(t)$ denotes the transpose of $x(t)$. $\mathcal{L}_2[0, t_f]$: The set of n-tuple functions with finite energy within $[0, t_f]$, i.e., for $x(t) \in \mathcal{L}_2[0, t_f]$, $\int_0^{t_f} \|x(t)\|\, dt = \int_0^{t_f} \left(x^T(t)x(t)\right)^{1/2} dt < \infty$, where t_f denotes the terminal time. The Jacobian vector and Hessian matrix of $V(x(t))$ are denoted as $\left(\frac{\partial V(x(t))}{\partial x(t)}\right) \triangleq \left(\frac{\partial V(x(t))}{\partial x_1(t)}, \ldots, \frac{\partial V(x(t))}{\partial x_i(t)}, \ldots, \frac{\partial V(x(t))}{\partial x_n(t)}\right)^T$ and $\left(\frac{\partial^2 V(x(t))}{\partial^2 x(t)}\right) \triangleq \begin{bmatrix} \frac{\partial^2 V(x(t))}{\partial^2 x_1(t)} & \cdots & \frac{\partial^2 V(x(t))}{\partial x_1(t)\partial x_n(t)} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 V(x(t))}{\partial x_n(t)\partial x_1(t)} & \cdots & \frac{\partial^2 V(x(t))}{\partial^2 x_n(t)} \end{bmatrix}$, respectively. I_a denotes the identity matrix with dimension $a \times a$; $0_{a\times b}$ denotes the zero matrix with dimension $a \times b$.

II. PROBLEM DESCRIPTION

Consider the following nonlinear stochastic signal system

$$\begin{aligned} \dot{x}(t) &= A(x(t)) + B(x(t))v(t) \\ y(t) &= C(x(t)) + n(t) \end{aligned} \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $y(t) \in \mathbb{R}^m$ is the measurement output, $v(t) \in \mathbb{R}^{n_v}$ denotes the stochastic external disturbance, and $n(t) \in \mathbb{R}^{n_n}$ denotes the stochastic measurement noise. The system functions $A(x(t))$, $B(x(t))$, and $C(x(t))$ are nonlinear function of state vector $x(t)$ which satisfy Lipschitz condition.

Before the discussion of filter design of nonlinear stochastic signal system in (1), the observability of nonlinear stochastic signal system (i.e., the ability of observing $x(t)$ from output measurement $y(t)$) must be guaranteed at first. In general, the observability criterion of nonlinear stochastic signal system in (1) can not be expressed in a simple explicity condition (e.g., rank condition or matrix equality in the linear system). The following lemma is provided from [36] to test the observability of nonlinear stochastic signal system in (1).

Lemma 1 (See [36]): Define the augmented vector $Z(t) = [y^T(t), \dot{y}^T(t), \ddot{y}^T(t), \ldots, y^{(n)^T}(t)]$ *and the corresponding Hessian matrix* $H_z(x(t)) = \frac{\partial^2 z(t)}{\partial^2 x(t)}$. *Then the nonlinear stochastic signal system in (1) is observable at the equilibrium point* $x(t) = 0$ *if there exists a constant* $\varepsilon > 0$ *and a constant matrix* T *such that the absolute values of the leading principal minors* $\Delta_1(x(t)), \ldots, \Delta_n(x(t))$ *of* $TH_z(x(t))$ *satisfy the following conditions*:

$$\begin{aligned} \Delta_1(x(t)) &\geq \varepsilon \\ &\vdots \\ \Delta_n(x(t)) &\geq \varepsilon, \quad \forall x(t) \end{aligned}$$

where n *is the dimension of* $x(t)$ *and the principal minor* $\Delta_i(x(t))$ *is the determinant of the matrix by deleting the last* $n - i$ *columns and rows of Hessian matrix* $TH_z(x(t))$.

For the simplicity of design, we assume the nonlinear stochastic signal system in (1) is observable (i.e., the observable conditions in Lemma 1 are satisfied) and therefore the following conventional Luenberger-type filter can be employed to estimate the state of nonlinear stochastic signal system in (1)

$$\begin{aligned} \dot{\hat{x}}(t) &= A(\hat{x}(t)) + k(\hat{x}(t))(y(t) - \hat{y}(t)) \\ \hat{y}(t) &= C(\hat{x}(t)) \end{aligned} \tag{2}$$

where $\hat{x}(t)$ is the estimated state and $\hat{y}(t)$ is estimated measurement output.

In the above Luenberger-type filter, we will design nonlinear filter gain $k(\hat{x}(t))$ so that the estimate state $\hat{x}(t)$ could approach to $x(t)$ as possible despite the uncertain external disturbance $v(t)$ and measurement noise $n(t)$ in (1).

Let us denote the estimation error as

$$\tilde{x}(t) = x(t) - \hat{x}(t) \tag{3}$$

Then the estimation error dynamic system is given as follows:

$$\begin{aligned} \dot{\tilde{x}}(t) &= A(x(t)) - A(\hat{x}(t)) + B(x(t))v(t) \\ &\quad -k(\hat{x}(t))(y(t) - \hat{y}(t)) \\ &= A(x(t)) - A(\hat{x}(t)) - k(\hat{x}(t))(C(x(t)) - C(\hat{x}(t))) \\ &\quad +B(x(t))v(t) - k(\hat{x}(t))n(t) \end{aligned} \tag{4}$$

We assume the statistics of external disturbance $v(t)$ and measurement noise $n(t)$ are unavailable and therefore nonlinear Luenberger-type filter design in (2) is employed to treat the robust H_∞ state estimation design problem of nonlinear stochastic signal system in (1) to efficiently eliminate the effect of uncertain $v(t)$ and $n(t)$ on the state estimation.

The robust H_∞ filter design of (2) for the nonlinear stochastic signal system (1) is formulated as the following minmax Nash stochastic game problem [3], [7]

$$\min_{k(\hat{x}(t))} \max_{v(t), n(t)} \frac{E\int_0^{t_f} \tilde{x}^T(t)Q\tilde{x}(t)dt - V(x(0))}{E\int_0^{t_f} v^T(t)v(t) + n^T(t)n(t)dt} \leq \rho \tag{5}$$

where the symmetric weighting matrix $Q = Q^T \geq 0$ is on the estimation error $\tilde{x}(t)$, the Lyapunov function $V(\tilde{x}(0)) \in \mathbb{R}^1$ denotes the energy of the initial estimation error in (4), t_f denotes the final time of state estimation and ρ denotes the desired upper bound of filtering ability and $E(\cdot)$ denotes the expectation operation. The physical meaning of robust

H_∞ filter design strategy in (5) is that the worst-case effect of external disturbance $v(t) \in \mathcal{L}_2[0, t_f]$ and measurement noise $n(t) \in \mathcal{L}_2[0, t_f]$ on the estimation error $\tilde{x}(t)$ must be minimized by the Luenberger-type filter in (2) from the perspective of mean energy. In general, the initial Lyapunov energy function $V(\tilde{x}(0)) \geq 0$ in the numerator of (5) due to the initial error $\tilde{x}(0)$ must be extracted from the estimation error energy within $0 \sim t_f$ because it is independent on the design of H_∞ filter. Further, the worst-case effect of external disturbance $v(t)$ and measurement noise $n(t)$ on the estimation error of robust H_∞ filter must be equal to or less than a prescribed attenuation level ρ from the perspective of mean energy.

Theorem 1: The minmax robust H_∞ filter design strategy in (5) for nonlinear stochastic signal system in (1) and Luenberger-type filter in (2) is solved by the following worst-case $v^*(t)$ and $n^*(t)$ and H_∞ filter gain $k^*(\hat{x}(t))$

$$v^*(t) = \frac{1}{2\rho} B^T(x(t)) \left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)} \right) \tag{6}$$

$$n^*(t) = -\frac{1}{2\rho} k^{*T}(\hat{x}(t)) \left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)} \right) \tag{7}$$

$$k^*(\hat{x}(t)) = \frac{2\rho}{\left\| \left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)} \right) \right\|^2} \left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)} \right) \times (C(x(t)) - C(\hat{x}(t)))^T \tag{8}$$

where $\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)} \right) \triangleq \left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}_1(t)}, \frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}_2(t)}, \cdots, \frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}_n(t)} \right)^T$ *is the solution of following nonlinear partial differential HJIE,*

$$\begin{aligned} HJIE = {} & \tilde{x}^T(t) Q \tilde{x}(t) + \left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)} \right)^T \\ & \times (A(x(t)) - A(\hat{x}(t))) + \frac{1}{4\rho} \left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)} \right)^T B(x(t)) \\ & \times B^T(x(t)) \left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)} \right) - \rho (C(x(t)) - C(\hat{x}(t)))^T \\ & \times (C(x(t)) - C(\hat{x}(t))) = 0 \end{aligned} \tag{9}$$

and $\left\| \left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)} \right) \right\|^2$ *is defined as follows*:

$$\left\| \left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)} \right) \right\| \triangleq \left(\sum_{i=1}^{n} \left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}_i(t)} \right)^2 \right)^{\frac{1}{2}}$$

$$i.e., \left\| \left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)} \right) \right\|^2 = \left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)} \right)^T \left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)} \right) \tag{10}$$

Proof: See Appendix A. □

Remark 1: From HJIE in (9), it needs to solve a complex partial differential $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ from HJIE $= 0$, i.e., if the substitution of the output $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ of DNN into HJIE in (9) can let HJIE approach to zero, then DNN can solve HJIE and we can use the $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ to generate the H_∞ filter gain $k^*(\hat{x}(t))$ and the worst-case external disturbance $v^*(t)$ and measurement noise $n^*(t)$. In the conventional interpolation methods in [13]–[22], they assume the solution $V(\tilde{x}(t))$ of HJIE in (9) is of the quadratic Lyapunov form $V(\tilde{x}(t)) = \tilde{x}^T(t) P \tilde{x}(t)$ and nonlinear stochastic signal system can be interpolated by a set of local linearized systems through fuzzy interpolation method [13]–[15], global linearization method [16] or gain scheduling method [17] so that we can solve a set of Riccati-like algebraic equations for the filter gain $k^*(\hat{x}(t))$.

In general, it is very difficult at present to solve the nonlinear partial differential HJIE in (9) analytically or numerically for robust H_∞ filter design in (8) for nonlinear stochastic signal system in (1) because HJIE in (9) is a nonlinear differential function of $x(t)$, $\hat{x}(t)$ and $\tilde{x}(t)$. Especially, while the state vector $x(t)$ is unavailable and only output measurement $y(t)$ is available in (1), it almost impossible to directly solve $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ from HJIE due to the unavailable terms $A(x(t))$, $B(x(t))$ and $C(x(t))$ of unavailable $x(t)$ in (9) for H_∞ filter design in (8) by the conventional methods.

Remark 2: In the robust control design of nonlinear stochastic system [33], the HJIE of H_∞ control design problem is only the functional of $x(t)$, which is assumed available and can be measured directly. However, in the nonlinear H_∞ filter design problem, $x(t)$ is unavailable and needs to be estimated from output measurement $y(t)$. Further, the observer gain $k^*(t)$ in (8) also contains the term $C(x(t))$, which is unavailable too. Therefore, more efforts are needed to directly solve HJIE in (9) for the nonlinear H_∞ filter design problem because $A(x(t))$, $B(x(t))$ and $C(x(t))$ in HJIE are unavailable. In this study, to overcome this difficult, the estimation error dynamic model in (4) is needed to be embedded with Luenberger-type filter in H_∞ HJIE-embedded DNN-based filter scheme as shown in Fig. 1, but with unavailable $v(t)$ and $n(t)$ being replaced by $v^*(t)$ and $n^*(t)$ in the off-line training phase, respectively.

In the last decades, several approximation methods such as gain schedule method [17], [21], global linearization method [16] and fuzzy method [20] have been applied to interpolate some local linear stochastic signal systems to approximate nonlinear stochastic signal system in (1). In this situation, the nonlinear stochastic signal system is approximated by the following L local linear stochastic signal systems

$$\begin{aligned} \dot{x}(t) &= \sum_{i=1}^{L} h_i(\hat{x}(t))(A_i x(t) + B_i v(t)) \\ y(t) &= \sum_{i=1}^{L} h_i(\hat{x}(t))(C_i x(t) + n(t)) \end{aligned} \tag{11}$$

where $h_i(\hat{x}(t))$ is the ith local interpolation function such as fuzzy basis in fuzzy interpolation method [20], smooth function in global linearization method [22] or smoothing function in gain schedule method [17], A_i, B_i and C_i are local linearized system parameters of the ith local linear system.

In this situation, the Luenberger-type filter in (2) is approximated by the following form

$$\dot{\hat{x}}(t) = \sum_{i=1}^{L}\sum_{j=1}^{L} h_i(\hat{x}(t))h_j(\hat{x}(t))(A_i\hat{x}(t) + k_i(y(t) - C_j\hat{x}(t))) \tag{12}$$

and the estimation error dynamic equation in (4) can be approximated by

$$\dot{\tilde{x}}(t) = \sum_{i=1}^{L}\sum_{j=1}^{L} h_i(\hat{x}(t))h_j(\hat{x}(t))(A_i - k_iC_j)\tilde{x}(t) + B_iv(t) - k_in(t) \tag{13}$$

In this situation, the robust H_∞ filter design strategy in (5) for a local linear interpolation estimation error system in (13) needs to solve a set of L^2 Riccati-like algebraic equations [15]–[17], [20]–[22] for the design of filter gains k_i^*, $i = 1, \cdots, L$ in (12) if we assume the solution $V(\tilde{x}(t))$ in (9) is of the quadratic Lyapunov form $V(\tilde{x}(t)) = \tilde{x}^T(t)P\tilde{x}(t)$. If a system is very highly nonlinear, then a large number of L local linear syatems in (11) are needed to be interpolated to approximate the nonlinear stochastic signal system in (1). Consequently, a much larger number of L^2 Riccati-like algebraic equations are also needed to be solved for k_i^*, $i = 1, \cdots, L$ of H_∞ filter gains in (12), which will lead to a design complexity. Further, a complex Luenberger-type filter in (12) is needed to be computed at very time instant for the estimation of $\hat{x}(t)$ if L becomes large. Further, the limit selection of quadratic Lyapunov function $V(\tilde{x}(t)) = \tilde{x}^T(t)P\tilde{x}(t)$ as the solution of HJIE in (9) can make the design of robust H_∞ filter more conservative and the approximation error between (1) and (11) may also deteriorate the H_∞ filtering performance.

Recently, some deep learning algorithms have been widely used to learn some system behaviors by big data-driven methods with successful application to image pattern classification [24], language translation [25] and speech recognition [26] etc. Though these deep learning schemes may have led to a very good result, they have to employ a very large amount of experimental data for training deep neural network. Further, the conventional DNN approaches can not be applied to some specific filter design performance such as the optimal H_2 filtering performance and robust H_∞ filtering performance in nonlinear stochastic signal systems, which are very important in signal processing designs. In the nonlinear H_2 and H_∞ filter design, these exist a large number of nonlinear stochastic system models in practical applications. Further, we have accumulated many powerful theoretical nonlinear filter results in the last decades. Even these nonlinear models and theoretical results of nonlinear filter designs are complicated and difficult to solve, they can be employed to achieve as experts to train DNN to achieve nonlinear H_∞ filter design with a large amount of training data and time saving than the conventional big data-driven DNN learning algorithm.

In this study, a novel method of DNN will be proposed to directly solve $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ from the complex partial differential HJIE in (9) for the H_∞ filter gain $k^*(\hat{x}(t))$ in (8). Since the HJIE in (9) could guarantee the H_∞ filtering performance in (5), the proposed H_∞ DNN-based filter could not only efficiently attenuate the effect of external disturbance and measurement noise on the state estimation error but also significantly save a large amount of training data which are needed in the conventional big data-driven DNN approaches.

In the off-line training phase of the proposed H_∞ DNN-based filter scheme in Fig. 1, since $v(t)$ and $n(t)$ are unavailable, $y(t)$ is generated by nonlinear stochastic signal system model with $v(t)$ and $n(t)$ being replaced by the worst-case $v^*(t)$ in (6) and $n^*(t)$ in (7). This does not influence on the H_∞ filtering performance because the H_∞ filter gain $k^*(\hat{x}(t))$ in (8) is designed based on the worst-cast disturbance $v^*(t)$ and noise $n^*(t)$ as shown in Theorem 1. In the on-line operation phase of H_∞ DNN-based filter scheme, $y(t)$ is generated by the real nonlinear stochastic signal system in (1) through the true value $v(t)$ and $n(t)$.

Remark 3: In the previous works, in order to avoid solving a complex nonlinear partial differential HJIE of the robust H_∞ filter of nonlinear stochastic signal systems, the T-S fuzzy interpolation method in [19], [20], [22], the global linearization method [16] and gain scheduling method [17] have been employed to interpolate several local linearized systems to approximate the nonlinear stochastic signal system so that HJIE of robust nonlinear H_∞ filter design could be interpolated by a set of local Riccati-like equations or a set of equivalent local LMIs under the assumption of quadratic solution $V(\tilde{x}(t)) = \tilde{x}^T(t)P\tilde{x}(t)$ of HJIE, i.e., with different interpolation methods to interpolate some local linearized systems to approximate a nonlinear stochastic system so that the nonlinear partial HJIE can be transformed to a set of LMIs under the quadratic solution $V(\tilde{x}(t)) = \tilde{x}^T(t)P\tilde{x}(t)$, which could be easily solved by LMI Toolbox in Matlab. In this study, we focus on using DNN to directly sovle $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ from the complex nonlinear partial differential HJIE in (9) of H_∞ filter design of nonlinear stochastic signal systems so that DNN can output $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ to generate the H_∞ filter gain $k^*(\hat{x}(t))$ directly instead of solving $V(\tilde{x}(t)) = \tilde{x}^T(t)P\tilde{x}(t)$ by the interpolation schemes in the conventional design method. In this case, we can not only avoid the difficult calculation in solving HJIE but also save more training and computation time.

III. ROBUST H_∞ HJIE-EMBEDDED DNN-BASED FILTER VIA A CO-DESIGN OF H_∞ FILTER SCHEME AND DEEP LEARNING ALGORITHM

Due to the difficulty to directly solve $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ of HJIE in (9) for the filter gain $k^*(\hat{x}(t))$ in (8) by the conventional methods for the design of Luenberger-type filter in (2) to achieve minmax robust H_∞ filter strategy in (5), DNN is proposed to be trained to solve $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ of the HJIE in (9) for $k^*(\hat{x}(t))$,

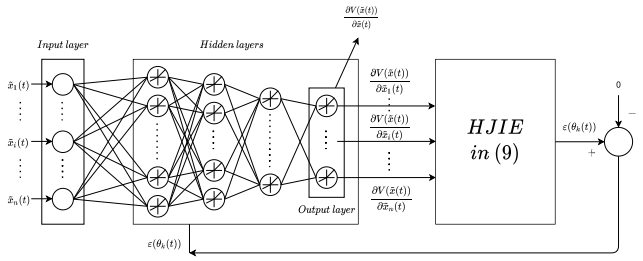

FIGURE 2. HJIE-embedded DNN architecture with input $\tilde{x}(t)$ to output $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ to solve HJIE in (9) after the training of Adam learning algorithm in (14) and (15) based on the error of $HJIE = \epsilon(\theta_k(t))$.

$v^*(t)$ and $n^*(t)$ in (6)-(8). Since $A(x(t))$, $B(x(t))$ and $C(x(t))$ in HJIE need the information of $x(t)$, which is unavailable and to be estimated, to overcome this difficulty, we need to synthesize $x(t)$ by $x(t) = \hat{x}(t) + \tilde{x}(t)$. Therefore, both Luenberger-type filter model in (2) and the estimation error dynamic in (4) are needed to be embedded as shown in Fig. 1 to obtain $\hat{x}(t)$ and $\tilde{x}(t)$, respectively. Therefore, the proposed H_∞ DNN-based filter can synthesize $x(t) = \hat{x}(t) + \tilde{x}(t)$ for the calculation of $A(x(t))$, $B(x(t))$ and $C(x(t))$ for solving HJIE in (9) as shown in the HJIE block of HJIE-embedded DNN in Fig. 1. As shown in Fig. 2, the proposed HJIE-embedded DNN architecture with input $\tilde{x}(t)$ is trained by the error $\epsilon(\theta_k(t))$ of solving $HJIE = 0$ to output $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ for H_∞ filter gain $k^*(\hat{x}(t))$. HJIE-embedded DNN in Fig. 2 consists of an input layer, several hidden layers, an output layer and an HJIE block. The neurons in hidden layers of DNN employ LeakyReLU as the activation function [27], which is the same as ReLU when input $\tilde{x}(t)$ is positive and is not equal to zero but a constant gradient when $\tilde{x}(t)$ is negative. Therefore, we have the advantage of ReLU and avoid the disadvantage of dead ReLU when input $\tilde{x}(t)$ is negative, i.e., LeakyReLU operates as follows [37]:

$$q(\tilde{x}(t)) = \begin{cases} \alpha_1 \tilde{x}(t) & \text{if } \tilde{x}(t) > 0 \\ \alpha_2 \tilde{x}(t) & \text{if } \tilde{x}(t) \le 0 \end{cases}$$

where α_1 and α_2 are some constant with $\alpha_1, \alpha_2 \in (0, 1)$.

In the off-line training phase, as shown in Fig. 1, we need the nonlinear stochastic signal system model in (1) to generate $y(t)$ for Luenberger-type filter in (2) and the estimation error dynamic system in (4). However external disturbance $v(t)$ and measurement noise $n(t)$ are unavailable in real nonlinear stochastic signal system in (1). In the robust H_∞ Luenberger-type filter design, the real external disturbance $v(t)$ and measurement noise $n(t)$ are not considered directly but replaced by their worst-case $v^*(t)$ and $n^*(t)$ in the design procedure from the minmax robust H_∞ filtering perspective. Therefore, we use $v^*(t)$ in (6) and $n^*(t)$ in (7) to replace real $v(t)$ and $n(t)$, respectively, without the influence on the estimation performance of robust H_∞ filter. From the flow chart of robust H_∞ HJIE-embedded DNN-based filter design in Fig. 1, $\tilde{x}(t)$ from the filter estimation error system in (4) is inputed to DNN to expect generating the desired output $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$, which is to be used to produce $v^*(t)$, $n^*(t)$ and $k^*(\hat{x}(t))$ according to (6), (7) and (8) in Theorem 1, respectively. Therefore, the nonlinear signal system model in (1) with $v^*(t)$, $n^*(t)$ and $u^*(t) = k^*(\hat{x}(t))$ is employed to produce $y(t)$ in (1). Luenberger-type filter in (2) can produce $\hat{x}(t)$, and estimation error system in (4) can generate $\tilde{x}(t)$ in the off-line training process of DNN simultaneously. The DNN output $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ is also sent to HJIE to check whether the output $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ approaches to its true value by checking $HJIE = 0$ or not in (9). If $HJIE = \epsilon(\theta_k(t)) \neq 0$, the error of HJIE will be fedback to train the weighting parameters in the hidden layers of DNN in Fig. 2 via Adam learning algorithm to minimize the objective funtion $\epsilon^2(\theta_k(t))$ as follows [28]:

$$\theta_k(t) = \theta_{k-1}(t) - \frac{l}{\sqrt{\hat{v}_k(t) + \tau}} \hat{m}_k(t), \ k = 1, \cdots, K \quad (14)$$

where $\theta_k(t)$ is the weighting parameter vector in the hidden layers of DNN which is to be trained for DNN to generate the output $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ at time t. l is the learning rate and K is the number of training steps at time t. The bias-corrected estimator $\hat{m}_k(t)$ and $\hat{v}_k(t)$ are given as [28]

$$\hat{m}_k(t) = \frac{m_k(t)}{1 - \lambda_1^k}, \ \hat{v}_k(t) = \frac{v_k(t)}{1 - \lambda_2^k} \quad (15)$$

where

$$m_k(t) = \lambda_1 m_{k-1}(t) + (1 - \lambda_1) g_k(t)$$
$$v_k(t) = \lambda_2 v_{k-1}(t) + (1 - \lambda_2) g_k^2(t)$$

and $g_k(t) = \frac{\partial}{\partial \theta_k(t)} \sqrt{\frac{1}{M} \sum^M \epsilon^2(\theta_k(t))}$ is the gradient vector of partial derivatives of objective function $\epsilon^2(\theta_k(t))$ with respect to $\theta_k(t)$ at the training step k at time t. M denotes the batch size. $\lambda_1, \lambda_2 \in [0, 1)$ in (15) are the degree of previous impact influence on the current direction and can be specified by the designer with the concept of momentum to avoid being trapped in a local minimum and speed up the learning rate [29]. λ_1^k and λ_2^k denote the kth power of λ_1 and λ_2, respectively. τ in (14) is a small number to be used to prevent the denominator from being zero. $m_k(t)$ and $v_k(t)$ in (15) are the moving average of gradient and squared gradient at time t. With $\hat{v}_k(t)$ in (15), we can take the advantage of the idea of adaptive learning rate, which is always large at the beginning and small near the minimum. In (14) and (15), the Adam learning algorithm can take both the advantages of momentum and RMSProp [30] as an efficient parameter-specific adaptive learning method with easy implementation and great performance, one of the most popular learning algorithm of neural networks of some optimization problems recently.

Remark 4: Unlike the conventional big data-driven DNN, with the help of nonlinear system model in (1), Luenberger-type filter in (2), estimation error dynamic system in (4) and HJIE of theoretical H_∞ filter of nonlinear stochastic system in (1), the proposed DNN-based filter scheme can converge quickly by Adam learning algorithm scheme in (14) and (15). The convergence of weighting parameter vector $\theta_k(t)$ of Adam

learning algorithm in (14) and (15) has been proven in [31]. If the number of hidden neurons of DNN and the number K of training steps is large enough, the updating weighting parameter vector $\theta_k(t)$ in (14) and (15) can converge to a globally optimal parameter vector at a linear convergent rate.

In the off-line training process of HJIE-embedded DNN in Fig. 1, the output $\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)_\epsilon$ of DNN is fed to HJIE to calculate the error of HJIE_ϵ as follows:

$$\begin{aligned} HJIE_\epsilon &= \tilde{x}^T(t)Q\tilde{x}(t) + \left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)_\epsilon^T \\ &\quad \times (A(x(t)) - A(\hat{x}(t))) + \frac{1}{4\rho}\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)_\epsilon^T \\ &\quad \times B(x(t))B^T(x(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)_\epsilon \\ &\quad -\rho(C(x(t)) - C(\hat{x}(t)))^T(C(x(t)) - C(\hat{x}(t))) \\ &= \epsilon(\theta_k(t)) \end{aligned} \tag{16}$$

where $x(t) = \hat{x}(t) + \tilde{x}(t)$ which are obtained from Luenberger-type H_∞ filter in (2) and the estimation error dynamic system in (4), respectively.

The error of HJIE in (16) will be fedback to train DNN by the Adam learning algorithm in (14) and (15). It is expected to produce the precise $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ for H_∞ filter gain $k^*(\hat{x}(t)) = \frac{2\rho}{\left\|\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right\|^2}\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)(C(x(t)) - C(\hat{x}(t)))^T$ in (8) and the worst-case external disturbance $v^*(t) = \frac{1}{2\rho}B^T(x(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)$ in (6) as well as the worst-case measurement noise $n^*(t) = \frac{1}{2\rho}k^*(\hat{x}(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)$ in (7) after the off-line training phase.

When the error $HJIE_\epsilon = \epsilon(\theta_k(t))$ of the embedded-HJIE DNN approaches to 0 by the Adam learning algorithm, it can be proven that the output $\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)_\epsilon$ of DNN can approach $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ of HJIE in (9) by the following theorem.

Theorem 2: If $\epsilon(\theta_k(t))$ approaches to 0 in (16) by Adam learning algorithm in (14) and (15), then $\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)_\epsilon$ in (16) will approach to $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ in (9) and $HJIE_\epsilon$ in (16) will approach to HJIE in (9), i.e., the filter gain $k^\epsilon(\hat{x}(t)) = \frac{2\rho}{\left\|\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right\|_\epsilon^2}\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)_\epsilon(C(x(t)) - C(\hat{x}(t)))^T$ based on the output $\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)_\epsilon$ of DNN of the HJIE-embedded DNN-based filter in Fig. 1 will approach to the H_∞ filter gain $k^*(\hat{x}(t))$ in (8) of the Theorem 1 for the Luenberger-type H_∞ filter in (2) for the nonlinear stochastic signal system in (1).

Proof: See Appendix B. □

Remark 5: (i) From Theorem 2 and Theorem 1, we could see that the proposed HJIE-embedded DNN-based filter in Fig. 1 will approach to the robust H_∞ filter in (8) as $\epsilon(\theta_k(t))$ of HJIE in (16) approaches to zero via Adam learning algorithm in (14) and (15). However, in the practical filtering applications, we will stop the training phase and transfer into the operation phase if $|\epsilon(\theta_k(t))| \le \delta$ for a small prescribed δ or the training steps achieve a given number K in (14). In the simulation example in the sequel, the number of training step at each time t is given by $K = 30$. (ii) In this study, unlike the conventional big data-driven DNN schemes, we could significantly save a much amount of training data and training time of HJIE-embedded DNN because the nonliner stochastic signal system model in (1), Luenberger-type H_∞ filter in (2), the estimation error dynamic system in (4) and theoretical H_∞ filtering result HJIE in (9) are employed to generate the necessary training data $\hat{x}(t)$, $\tilde{x}(t)$ and $x(t) = \hat{x}(t) + \tilde{x}(t)$ in the off-line training phase of H_∞ HJIE-embedded DNN-based filter in Fig. 1. Based on these training data generated by the corresponding system models and the theoretical HJIE result for the robust H_∞ filtering strategy, the H_∞ HJIE-embedded DNN-based filter design becomes possible for solving the very complex robust H_∞ filter design problem of nonlinear stochastic dynamic signal system in (1), unlike the most conventional big data-driven DNN schemes to be trained only for deciding yes or no (i.e., 0 or 1) in the conventional imaging classification and speech recognition problems [23]–[27].

For the convenience of training and practical design, the continuous nonlinear stochastic signal system in (1) can be represented by the following nonlinear stochastic sampling-data signal system as follows:

$$\begin{aligned} \frac{x(t+\Delta t) - x(t)}{\Delta t} &\approx A(x(t)) + B(x(t))v(t) \\ y(t) &= C(x(t)) + n(t) \end{aligned} \tag{17}$$

or

$$\begin{aligned} x(t+\Delta t) &\approx (x(t) + \Delta t A(x(t))) + \Delta t B(x(t))v(t) \\ y(t) &= C(x(t)) + n(t) \end{aligned} \tag{18}$$

where $\Delta t > 0$ denotes the sampling time.

Similarly, the Luenberger-type filter in (2) and the estimation error system in (4) are also represented by the following nonlinear sample-data stochastic signal systems, respectively.

$$\hat{x}(t+\Delta t) = (\hat{x}(t) + \Delta t A(\hat{x}(t))) + \Delta t k(\hat{x}(t))(y(t) - \hat{y}(t)) \tag{19}$$

$$\begin{aligned} \tilde{x}(t+\Delta t) &= (\tilde{x}(t) + \Delta t A(x(t)) - \Delta t A(\hat{x}(t))) \\ &\quad -\Delta t k(\hat{x}(t))(C(x(t)) - C(\hat{x}(t))) \\ &\quad +\Delta t B(x(t))v(t) - \Delta t k(\hat{x}(t))n(t) \end{aligned} \tag{20}$$

Then the flow chart of HJIE-embedded DNN-based H_∞ filter design in Fig. 1 is modified in Fig. 3 as follows:

At $t = 0$ in the off-line training phase in Fig. 3, we randomly select N sample vectors near the error $\tilde{x}(0)$ at first and then send them into DNN to expect to output $\left(\frac{\partial V(\tilde{x}(0))}{\partial \tilde{x}(0)}\right)_\epsilon$, which is used to generate $v^*(0)$, $n^*(0)$ and $k^*(\hat{x}(0))$ in (6)-(8) and sent to calculate $HJIE_\epsilon = \epsilon(\theta_k(0))$ in (16) to be fedback to train the weighting parameters of hidden neurons in DNN by Adam learning algorithm in (14) and (15). After finishing K training processes in the off-line training phase, the output $\left(\frac{\partial V(\tilde{x}(0))}{\partial \tilde{x}(0)}\right)_\epsilon$ of DNN is used to produce $v^*(0)$, $n^*(0)$ and $k^*(\hat{x}(0))$ in (6)-(8). With $v^*(0)$, $n^*(0)$ and $k^*(\hat{x}(0))$, $\hat{x}(0)$, $\tilde{x}(0)$ and $x(0) = \hat{x}(0) + \tilde{x}(0)$, we could compute $y(0)$ from (1) and

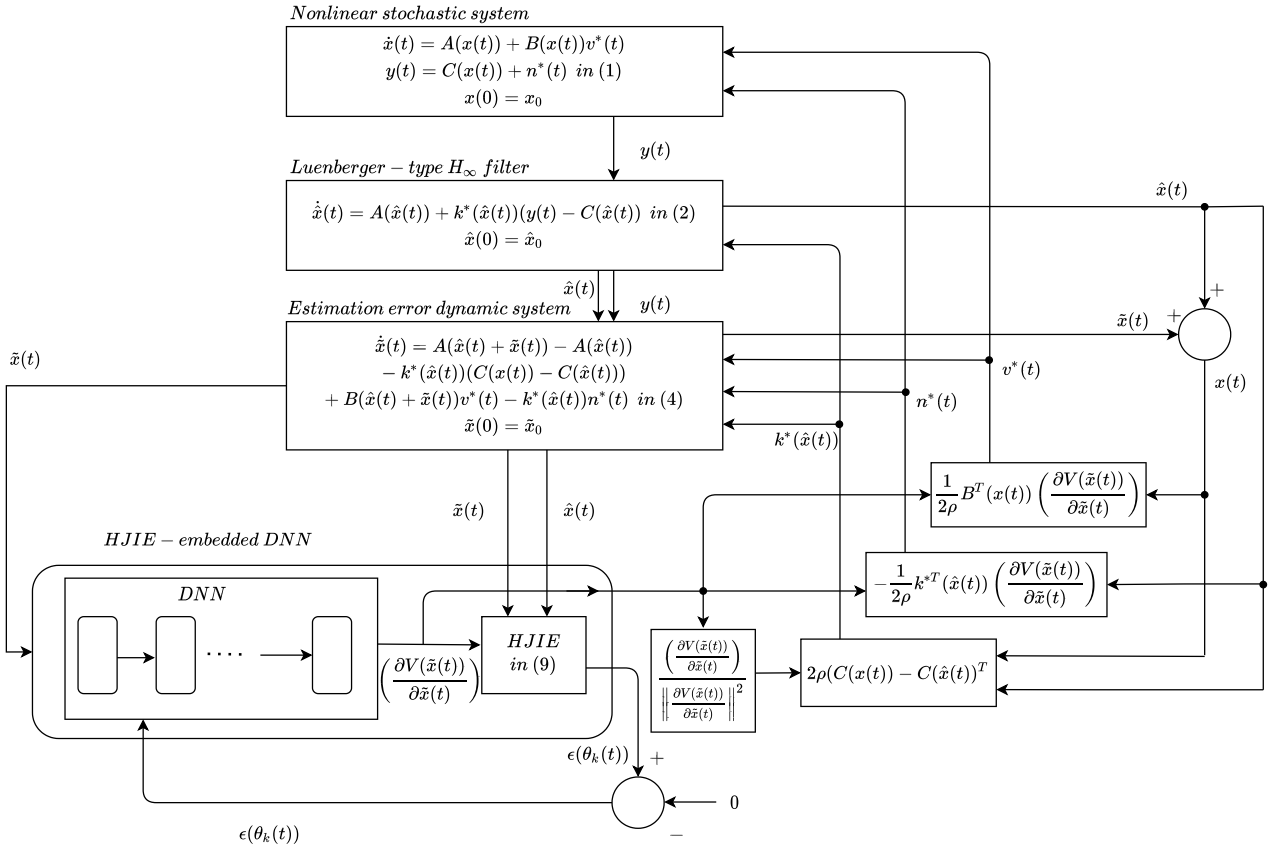

FIGURE 3. The flow chart of robust H_∞ HJIE-embedded DNN-based filter scheme of nonlinear stochastic signal system in sample-data system in (18). In the on-line operation phase, the well-trained HJIE-embedded DNN in off-line training phase is directly applied. In the on-line operation the output $y(t)$ is generated by real physical system in (1) with real external disturbance $v(t)$ and measurement noise $n(t)$. The other procedures are similar to the off-line training phase. In general, the DNN training algorithm is closed in the on-line operation phase except $\epsilon(\theta_k(t)) \geq \delta$ for a prescribed small positive value δ.

generate $\hat{x}(\Delta t)$ from (19) and $\tilde{x}(\Delta t)$ from (20) and $x(\Delta t) = \hat{x}(\Delta t) + \tilde{x}(\Delta t)$.

At $t = \Delta t$, we randomly select N sample vectors near the error $\tilde{x}(\Delta t)$ and then send them into DNN to output $\left(\frac{\partial V(\tilde{x}(\Delta t))}{\partial \tilde{x}(\Delta t)}\right)_\epsilon$, which is used to generate $v^*(\Delta t)$, $n^*(\Delta t)$ and $k^*(\hat{x}(\Delta t))$ in (6)-(8) and sent to calculate $HJIE_\epsilon = \epsilon(\theta_k(\Delta t))$ to feedback to train the weighting parameters of DNN by Adam learning algorithm. After K training steps, the output $\left(\frac{\partial V(\tilde{x}(\Delta t))}{\partial \tilde{x}(\Delta t)}\right)_\epsilon$ of DNN is used to produce $v^*(\Delta t)$, $n^*(\Delta t)$ and $k^*(\hat{x}(\Delta t))$ in (6)-(8). With $v^*(\Delta t)$, $n^*(\Delta t)$ and $k^*(\hat{x}(\Delta t))$ in (6)-(8), we could compute $y(\Delta t)$ and generate $\hat{x}(2\Delta t)$ from (19) and $\tilde{x}(2\Delta t)$ from (20) and $x(2\Delta t) = \hat{x}(2\Delta t) + \tilde{x}(2\Delta t)$.

Sequentially, at $t = t_f$, we also randomly select N sample vectors near the error $\tilde{x}(t_f)$ and then send them into DNN to output $\left(\frac{\partial V(\tilde{x}(t_f))}{\partial \tilde{x}(t_f)}\right)_\epsilon$, which is used to generate $v^*(t_f)$, $n^*(t_f)$ and $k^*(\hat{x}(t_f))$ in (6)-(8) and sent to calculate $HJIE_\epsilon = \epsilon(\theta_k(t_f))$ with the feedback of $\epsilon(\theta_k(t_f))$ to train the weighting parameters of DNN by Adam learning algorithm. After K training steps, the output of training model is used to produce $v^*(t_f)$, $n^*(t_f)$ and $k^*(\hat{x}(t_f))$ in (6)-(8). With $v^*(t_f)$, $n^*(t_f)$ and $k^*(\hat{x}(t_f))$ in (6)-(8), we could generate $\hat{x}(t_f + \Delta t)$.

After we have finished the above HJIE-embedded DNN training process of H_∞ HJIE-embedded filter, we begin the operation phase of DNN-based H_∞ filter. In the on-line operation phase as shown in Fig. 3, we do not need to use the information of $v^*(t)$ and $n^*(t)$ generated by DNN but the $v(t)$ and $n(t)$, which will occur themself in the real nonlinear stochastic signal system in (1). Therefore, every time step, we only need to input $\tilde{x}(t)$ to DNN to output $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ to compute filter gain $k^*(\hat{x}(t)) = \frac{2\rho}{\left\|\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right\|^2}\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)(C(x(t)) - C(\hat{x}(t)))^T$ for Luenberger-type H_∞ filter and estimation error system. However, the training of DNN is necessary if the absolute error $|\epsilon(\theta_k(t))| \geq \delta$ for a prescribed small value δ.

Remark 6: For the off-line training processes, there are some data processings usually being used before training DNN such as normalization or standardization in the deep learning [30]. If the error $\tilde{x}(t)$ is so large that all the random sample data will influence the training speed, we can standardize all the sample data before sending them into DNN. Hence, we can not only speed up the training speed but also improve the estimation accuracy in the off-line training phase.

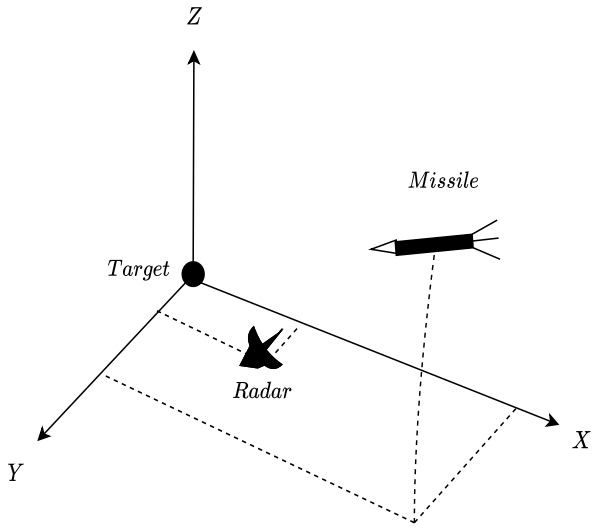

FIGURE 4. The 3-D geometry of incoming ballistic missile. x axis, y axis and z axis denotes the downrange, offrange and attitude of ballistic missile. The target of missile is set at the origin of the Cartesian coordinate and the radar is located at $(x, y, z) = (300, 200, 0)$.

IV. SIMULATION EXAMPLE

After the robust H_∞ HJIE-embedded DNN-based filter design in Fig. 1 is proposed in the above sections for the robust state estimation of the nonlinear stochastic signal system in (1), in this section, a simulation example is given to illustrate the design procedure based on Fig. 3 and to validate the filtering performance of the proposed H_∞ DNN-based filter of nonlinear stochastic signal system. In this study, an H_∞ robust state estimation problem of incoming ballistic missile by the measurement of radar system in Fig. 4 is given to confirm the robust state estimation performance of the proposed H_∞ HJIE-embedded DNN-based filtering scheme when compared with the extended Kalman filter [8] and particle filter [11].

Suppose an incoming ballistic missile will attack a target at the origin of Cartesian coordinate and is measured by the sensor of radar located at $(x, y, z) = (300, 200, 0)$. The dynamic motion equation of incoming ballistic missile is given as follows [20], [35]:

$$\begin{aligned}
\ddot{x}(t) &= \frac{-\rho(z(t))g\sqrt{\dot{x}^2(t)+\dot{y}^2(t)+\dot{z}^2(t)}}{2\beta}\dot{x}(t) \\
\ddot{y}(t) &= \frac{-\rho(z(t))g\sqrt{\dot{x}^2(t)+\dot{y}^2(t)+\dot{z}^2(t)}}{2\beta}\dot{y}(t) \\
\ddot{z}(t) &= \frac{-\rho(z(t))g\sqrt{\dot{x}^2(t)+\dot{y}^2(t)+\dot{z}^2(t)}}{2\beta}\dot{z}(t) - g
\end{aligned} \tag{21}$$

where $x(t)$, $y(t)$ and $z(t)$ are the target-centered Cartesian coordinates of the incoming ballistic missile, β is the ballistic coefficient, g is the gravity constant, $\rho(z(t))$ is the density of the atmosphere at the position of the incoming ballistic missile and is defined as follows [20], [35]:

$$\rho(z(t)) = \begin{cases} \rho_h e^{-\alpha_h z(t)}, \quad \rho_h = 1.75, \ \alpha_h = 1.49 \times 10^{-4}, \\ \text{if } \ z(t) \geq 9144 \ \text{ meters} \\ \rho_l e^{-\alpha_l z(t)}, \ \rho_l = 1.227, \alpha_l = 1.093 \times 10^{-4}, \\ \text{if } \ z(t) < 9144 \ \text{ meters} \end{cases}$$

Based on the state space dynamic model, the motion equations of incoming ballistic missile in (21) could be rewritten as the following nonlinear dynamic equation

$$\begin{aligned}
\dot{X}(t) &= \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \\ \frac{-\rho(z(t))g\sqrt{\dot{x}^2(t)+\dot{y}^2(t)+\dot{z}^2(t)}}{2\beta}\dot{x}(t) \\ \frac{-\rho(z(t))g\sqrt{\dot{x}^2(t)+\dot{y}^2(t)+\dot{z}^2(t)}}{2\beta}\dot{y}(t) \\ d\,\frac{-\rho(z(t))g\sqrt{\dot{x}^2(t)+\dot{y}^2(t)+\dot{z}^2(t)}}{2\beta}\dot{z}(t) - g \end{bmatrix} \\
&= f(X(t))
\end{aligned} \tag{22}$$

where state vector is defined as

$$X(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \\ \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \\ x_4(t) \\ x_5(t) \\ x_6(t) \end{bmatrix}$$

Remark 7: In order to avoid the symbol confusion between the state vector $x(t)$ of missile system in (1) and the coordinate $x(t)$, $y(t)$ and $z(t)$ of missile position, in the following, we use $X(t)$ to temporarily represent the state vector of missile system until (26).

Suppose that the incoming missile employs a guidance control law

$$\begin{aligned}
u(t) = f_{con}X(t) = [&-0.1x_1(t) - x_4(t), -0.1x_2(t) - x_5(t), \\
&-0.08x_3(t) - x_6(t), -0.01x_4(t), \\
&-0.01x_5(t), -0.01x_6(t)]^T
\end{aligned}$$

and suffers from stochastic external disturbance $v(t)$ as follows:

$$\dot{X}(t) = f(X(t)) + f_{con}X(t) + B(X(t))v(t) \tag{23}$$

Let $Y(t)$ denote the sensor measurement of radar system and $n(t)$ denote the stochastic measurement noise of radar sensor. Then the measurement equation of radar system is given by

$$Y(t) = C(X(t)) + n(t) \tag{24}$$

where $C(X(t)) = [x_1(t) - 300, \ x_2(t) - 200, \ x_3(t), \ x_4(t), \ x_5(t), \ x_6(t)]^T$.

Then the nonlinear stochastic signal system of radar detection of incoming missile can be rewritten as follows:

$$\begin{aligned} \dot{X}(t) &= A(X(t)) + B(X(t))v(t) \\ Y(t) &= C(X(t)) + n(t) \end{aligned} \tag{25}$$

where

$$A(X(t)) = \begin{bmatrix} -0.1x_1(t) \\ -0.1x_2(t) \\ -0.08x_3(t) \\ \dfrac{-\rho(x_3(t))g\sqrt{x_4^2(t)+x_5^2(t)+x_6^2(t)}}{2\beta}x_4(t) \\ -0.01x_4(t) \\ \dfrac{-\rho(x_3(t))g\sqrt{x_4^2(t)+x_5^2(t)+x_6^2(t)}}{2\beta}x_5(t) \\ -0.01x_5(t) \\ \dfrac{-\rho(x_3(t))g\sqrt{x_4^2(t)+x_5^2(t)+x_6^2(t)}}{2\beta}x_6(t) \\ -0.01x_6(t) - g \end{bmatrix},$$

$$B(X(t)) = [0_{1\times 3}\ 1\ 1\ 1]^T$$

Based on the flow chart of H_∞ HJIE-embedded DNN-based filter scheme for radar detection of incoming missile stochastic system in Fig. 3, for practical design, the nonlinear stochastic radar detection system is modified to a nonlinear stochastic sample-data signal system. Then from (18) we obtain the following sample-data stochastic signal system of radar detection of incoming missile system,

$$\begin{aligned} X(t+\Delta t) &= (X(t) + \Delta t A(X(t))) + \Delta t B(X(t))v(t) \\ Y(t) &= C(X(t)) + n(t) \end{aligned} \tag{26}$$

where the sampling time Δt is $0.01s$.

Suppose we want to design the H_∞ HJIE-embedded DNN-based filter in Fig. 3 for the sample-data radar detection system in (26) to achieve the minmax H_∞ state estimation performance in (5) of incoming missile with a desired filtering ability $\rho = 0.03$. According to Theorem 1, we need to solve $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ of the following HJIE for Luenberger gain $k^*(\hat{x}(t))$ in (8) of robust H_∞ Luenberger-type filter in (2),

$$\begin{aligned} HJIE = {} & \tilde{x}^T(t)Q\tilde{x}(t) + \left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)^T \\ & \times (A(x(t)) - A(\hat{x}(t))) + \frac{1}{4\rho}\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)^T B(x(t)) \\ & \times B^T(x(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right) - \rho(C(x(t)) - C(\hat{x}(t)))^T \\ & \times (C(x(t)) - C(\hat{x}(t))) = 0 \end{aligned} \tag{27}$$

where $\hat{x}(t)$ and $\tilde{x}(t)$ are obtained from Luenberger-type filter in (19) and the estimation error system in (20) of the sample-data missile system in (26). In general, it is almost impossible to directly solve $V(\tilde{x}(t))$ or $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ analytically or numerically from the above HJIE for $k^*(\hat{x}(t))$ in (8) since the HJIE in (27) is a highly nonlinear partial differential equation containing unavailable $x(t)$, $\hat{x}(t)$ and $\tilde{x}(t)$ which need to be generated from their corresponding dynamic systems in (26), (19) and (20), respectively. Therefore, the proposed robust H_∞ HJIE-embedded DNN-based filter design in Fig. 3 is employed to solve $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ from HJIE in (27) for $v^*(t)$, $n^*(t)$ and $k^*(t)$ in (6), (7) and (8), respectively.

From the flow chart of H_∞ HJIE-embedded DNN-based filter scheme in Fig. 3, HJIE-embedded DNN is trained off-line via Adam learning algorithm in (14) and (15) by the error $HJIE_\epsilon = \epsilon(\theta_k(t))$ in (16) to achieve $HJIE = 0$ in (27) for DNN to output $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$. In this situation the robust H_∞ filter gain $k^*(\hat{x}(t))$, $v^*(t)$ and $n^*(t)$ can be obtained, which are sent to nonlinear stochastic missile detection system in (26), Luenberger-type filter in (19) and estimation error dynamic system in (20) to generate the corresponding signals $y(t)$, $\hat{x}(t)$ and $\tilde{x}(t)$ as well as $x(t) = \hat{x}(t) + \tilde{x}(t)$ for the next step $t + \Delta t$ of Adam training process as shown in Fig. 3.

Remark 8: At present, there exists no study to combine the theoretical result of nonlinear H_∞ filter and deep learning algorithm to implement the H_∞ deep neural network-based filter of nonlinear signal systems. It is not easy to implement this co-design of nonlinear H_∞ filtering algorithm and DNN learning algorithm for the state estimation of nonlinear stochastic signal system because the conventional deep learning algorithms need lots of empirical data to train DNN as image classifier or recognizer. Further, in the nonlinear H_∞ filter design, we need to solve HJIE, which is a function of the state $x(t)$ of nonlinear stochastic system. However, the real state $x(t)$ is unavailable, so we could not train the DNN model directly. We need filter model to generate $\hat{x}(t)$ and estimation error model to generate $\tilde{x}(t)$. Therefore, we could obtain $x(t) = \hat{x}(t) + \tilde{x}(t)$ for solving HJIE. Further, since external disturbance $v(t)$ and measurement noise $n(t)$ are unavailable. Therefore, it is very difficult to generate $y(t)$ by signal system model in (1) and estimation error model in (4) to generate $\tilde{x}(t)$. Therefore, we use the worst-case $v^*(t)$ in (6) and $n^*(t)$ in (7) to replace $v(t)$ and $n(t)$ for signal system model in (1) to generate $y(t)$ for filter model in (2) to generate $\hat{x}(t)$, and estimation error dynamic system in (4) to generate $\tilde{x}(t)$ and therefore $x(t) = \hat{x}(t) + \tilde{x}(t)$ for HJIE for training DNN by Adam learning algorithm at each time t in the off-line training phase. In addition, in order to train DNN by $\tilde{x}(t)$ to output $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ for solving HJIE for filter gain $k^*(\hat{x}(t))$ in (8), $v^*(t)$ in (6) and $n^*(t)$ in (7), an HJIE block is embedded in the DNN structure to approach to our desired output $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$. Therefore, we can use the output of DNN to produce the filter gain $k^*(\hat{x}(t))$, the worst-case external disturbance $v^*(t)$ and measurement noise $n^*(t)$ as shown in Fig. 1.

In this example, the architecture of DNN in Fig. 2 contains input layer, four hidden layers, one HJIE layer and output layer. The input layer consists of 6 inputs, four hidden layers consist of 256, 128, 32 and 6 hidden neural units sequentially which can be regarded as the concept of data compression. We want to train DNN to fit the nonlinear function $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ of HJIE for H_∞ filter gain $k^*(\hat{x}(t))$ through the fewer and fewer

hidden neurals in each hidden layer sequentially. Further, if the learning rate l in (14) is assumed to be too large at the beginning, the gradient may be trapped in the local minimum, otherwise, the training speed will be too slow. The parameters λ_1, λ_2 in Adam learning algorithm in Eq. (15) can also represent the impact in current direction from previous direction to avoid trapping in the local minimum and speed up the learning rate. These parameters can not be assumed to be small due to the important information in the previous direction. Finally, these parameters l, λ_1, λ_2 and τ in Adam learning algorithm in (14) and (15) are selected as $0.001, 0.9, 0.999$ and 10^{-7}. The number of training steps and batch size are $K = 30$, $M = 800$, and the number of random sample data is $N = 2000$ as the number of training inputs. In the H_∞ filtering design strategy of the radar detection system in (5), the weighting matrix Q and ρ in (5) are specified as $Q = 10^{-5} I_6$ and $\rho = 0.03$, respectively. The sampling time Δt used in this example is $0.01s$. The external disturbance $v(t)$ and measurement noise $n(t)$ are both assumed to be Gaussian random noise of $20N(0, 1)$. After H_∞ DNN-based filter being trained by Adam learning algorithm, the trajectory $x_1(t), \cdots, x_6(t)$ of the incoming missile in (25) is simulated with the initial condition $X(0) = [150000, 210000, 120000, -2500, -2500, -2500]^T$.

Remark 9: In the real application with DNN method, at the beginning of off-line training phase, we need to randomly select the initial training data near the error $\tilde{x}(t)$ which will significantly affect the training performance. If the domain is limited by the random selection in the off-line training phase, the state $x(t) = \hat{x}(t) + \tilde{x}(t)$ may be far from the training data during the on-line operation phase. Hence, it will limit the domain of $\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)_\epsilon$ to approach to the real $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ of HJIE. In this situation, the error $\epsilon(\theta_k(t))$ will be larger than a small prescribed δ, (i.e., $|\epsilon(\theta_k(t))| > \delta$) so that we need to start on Adam learning algorithm again for updating weighting parameters of DNN or change the method of random selections to train the weighting parameters of DNN again.

Remark 10: In this simulation, we have tested three different numbers of training data with 2000, 20000 and 200000 initial conditions, respectively. The largest one needs more computation time than others. However, the error of DNN in 2000 initial conditions and 20000 initial conditions are almost the same in the training performance. Further, when the number of hidden layers with 2000 training inputs exceeds 4, the error of DNN will not decrease anymore due to the overfitting in the DNN. To avoid this overfitting problem, more initial conditions as training inputs are required and will increase the computation time, i.e., there is a tradeoff between the loss and the computation time. Finally, we select 4 hidden layers and 2000 initial conditions as training inputs in our simulation.

Remark 11: For the weighting matrix Q of robust H_∞ filter design strategy in (5), if the estimation error $\tilde{x}_i(t)$ in the ith state is more significant for our incoming ballistic missile simulation, then we will set a larger parameter in the term (i, i) of Q. In contrast, the less parameter in the term (j, j) of Q if we consider the estimation error $\tilde{x}_j(t)$ in the jth state is insignificant. In this simulation, we consider the estimation error in each element as equal importance with $Q = 10^{-5} I_6$ in the incoming ballistic missile.

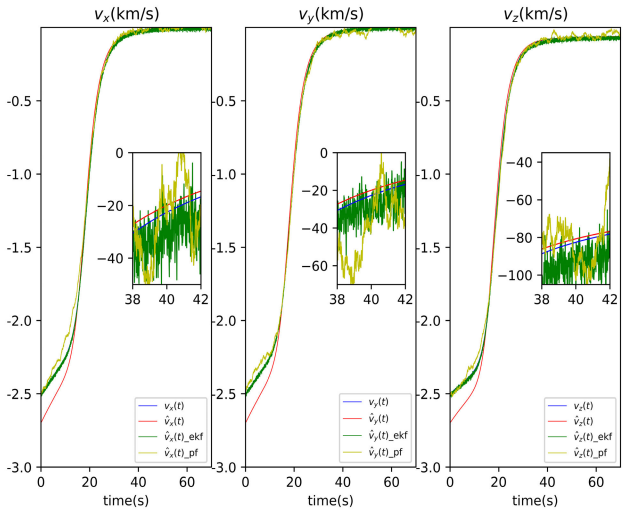

FIGURE 5. The downrange, offrange and vertical velocities and their corresponding estimations by the proposed H_∞ HJIE-embedded DNN-based filter and other comparison methods. The blue line is the real velocity of the incoming ballistic missile and the red line is the velocity of the proposed H_∞ HJIE-embedded DNN-based filter. The green and yellow lines are the velocities of the extended Kalman filter in [8] and particle filter in [11], respectively, in comparison.

Based on the proposed H_∞ HJIE-embedded DNN-based filter, the location and the velocity of the incoming ballistic missile system and their corresponding estimations are shown in Fig. 5 and Fig. 6, respectively. From the simulation results in Fig. 5 and Fig. 6, although the state estimation of incoming missile has a transient response in the beginning, we can see that it still achieves the robust H_∞ filtering performance under the effect of the uncertain external disturbance and measurement noise. The real H_∞ state estimation performance of the incoming missile by the proposed robust H_∞ DNN-based filter is calculated as follows:

$$\frac{\int_0^{70} \tilde{x}^T(t) Q \tilde{x}(t) dt - V(\tilde{x}(0))}{\int_0^{70} \left(v^T(t)v(t) + n^T(t)n(t)\right) dt} \approx 0.012 \leq 0.03$$

It is seen that the proposed HJIE-embedded DNN filter scheme in Fig. 1 could achieve the prescribed H_∞ filtering performance to efficiently attenuate the effect of random external disturbance and measurement noise. Obviously, the filtering performance of the proposed robust H_∞ HJIE-embedded DNN filter is better than the prescribed $\rho = 0.03$. The reason is that the H_∞ filter is designed based on minimizing the effect of the worst-case external disturbance $v^*(t)$ in (6) and the worst-case measurement noise $n^*(t)$ in (7), which may not exist in the real situation. However, in this simulation, we use $v(t)$ and $n(t)$ of stochastic process of $20N(0, 1)$ as external disturbance and measurement noise, respectively, which are not the worst-case $v^*(t)$ and $n^*(t)$. From the simulation results in Fig. 5 and Fig. 6, it is seen that

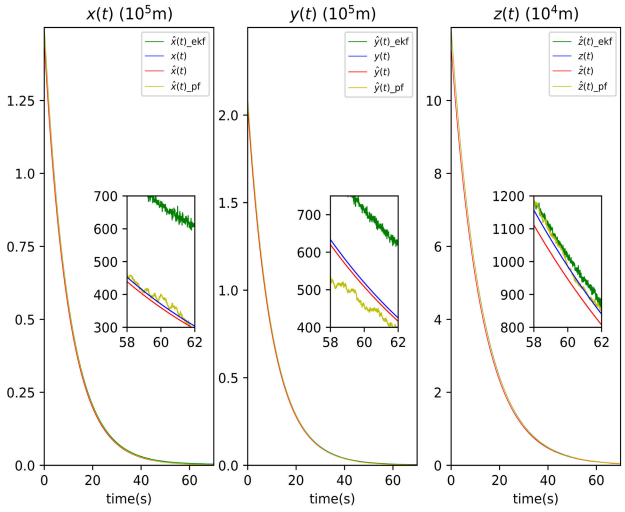

FIGURE 6. The downrange, offrange and vertical trajectories and their corresponding estimations by the proposed H_∞ HJIE-embedded DNN-based filter and other comparison methods. The blue line is the real trajectory of the incoming ballistic missile and the red line is the trajectory of the proposed H_∞ HJIE-embedded DNN-bsaed filter. The green and yellow lines are the trajectories of the extended Kalman filter in [8] and particle filter in [11], respectively, in comparison. Obviously, the estimation performance of extended Kalman filter is very poor in the trajectory estimation of incoming ballistic missile. The mean error and standard deviation (μ, σ) in the trajectory estimation of H_∞ DNN-based filter and particle filter at the estimation of $x(t)$, $y(t)$ and $z(t)$ are given as follows: H_∞ DNN-based filter :$(-11.11, 1.32)$, $(-11.11, 1.32)$, $(-38.78, 3.61)$, and particle filter :$(7.25, 10.51)$, $(-59.42, 21.86)$, $(11.79, 11.75)$. Obviously, particle filter has a good mean trajectory estimation performance but with a very large random fluctuation, which is not suitable for the high precise and robust trajectory estimation of incoming missile.

the extended Kalman filter has a poor trajectory estimation performance of incoming missile and the particle filter has a good mean trajectory estimation but with a large random fluctuation at every time step of estimation because its estimation is based on the condition probability of state $x(t)$ under measurement $y(t)$. Comparing with the results of extented Kalman filter with exact statistical knowledge of $v(t)$ and $n(t)$ [8] and particle filter with exact statistical knowledge of $v(t)$ and $n(t)$ [11] in Fig. 5 and Fig. 6 for the trajectory estimation of incoming missile, we found that the proposed robust H_∞ HJIE-embedded DNN-based filter without the knowledge of external disturbance and measurement noise has better estimation performance than the conventional extented Kalman filter and particle filter for nonlinear stochastic signal systems with exact covariance matrices of external disturbance and measurement noise.

Remark 12: In the filter design field at present, there exists no study to use the conventional big data-driven DNN method to deal with this nonlinear filter design problem due to the unavailable empirical data for training DNN. In general, the conventional DNN is based on the input/output empirical data pairs to train the weighting parameters of hidden layers in DNN by optimizer. However, in the nonlinear H_∞ filter design problem, $x(t)$ is unavailable and needs to be estimated from output measurement $y(t)$. Therefore, we can not compare with the conventional big data-driven DNN method in nonlinear H_∞ filter design. Instead, we show the two nonlinear filtering methods of extended Kalman filter and particle filter to compare the filtering performance with the proposed robust H_∞ HJIE-embedded DNN filter.

In the extended Kalman filter [8], at every time step, we need to update the linearization of the nonlinear functions $A(x(t))$, $B(x(t))$ and $C(x(t))$ of nonlinear stochastic signal system in (1), update the estimation error covariance prediction for state prediction and then update the filtering error covariance for state estimation (filtering). There are a large amount of computations to be performed at every time step. If the stochastic signal system is highly nonlinear, the linearization error will propagate to the subsequent step. Further, if the covariance matrices of $v(t)$ and $n(t)$ are unknown or uncertain, it will deteriorate the performance of extended Kalman filter. For the particle filter design [11], we need to predict the conditional probability of state $x(t)$ with the observation according to $P(x_t|y_{1:t-1}) = \int_{dx_{t-1}} P(x_t|x_{t-1})P(x_{t-1}|y_{1:t-1})dx_{t-1}$ where $y_{1:t-1}$ denotes the partial observations $y_1, \ldots, y_{t-1}$, and update the filtering of particle filter according to the conditional probability $P(x_t|y_{1:t}) = \frac{P(y_t|x_t)P(x_t|y_{1:t-1})}{P(y_t|y_{1:t-1})}$. Further, we still need to update the importance distribution for particle filter. There are a lot of complicated computations in every step of particle filter. However, in the proposed robust H_∞ HJIE-embedded DNN-based filter scheme, after the HJIE-embedded DNN has been trained by Adam learning algorithm in the off-line training phase, the proposed H_∞ DNN-based filter could perform with a quite good trajectory estimation performance in the incoming missile detection system without the knowledge of external disturbance and measurement noise. The better filtering performance of the proposed method lies in: (i) The proposed H_∞ DNN-based filter is based on the global nonlinear solution $V(\tilde{x}(t))$ of HJIE while the conventional design in extended Kalman filter [8], particle filter [11] are limited to the quadratic solution $V(\tilde{x}(t)) = \tilde{x}^T(t)P\tilde{x}(t)$. (ii) The nonlinear H_∞ filter could efficiently attenuate the effect of uncertain external disturbance and measurement noise on the state estimation performance. (iii) The DNN could universally approximate any nonlinear function like $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ for H_∞ filter gain $k^*(\hat{x}(t))$ after an efficient training algorithm like Adam learning algorithm.

V. CONCLUSION

In this paper, in order to overcome the complexity and difficulty in the design of robust H_∞ filter of nonlinear stochastic signal systems with uncertain external disturbance and measurement noise, an HJIE-embedded DNN-based filter design in Fig. 1 is proposed to employ artificial neural network to directly solve HJIE of H_∞ filtering design via deep learning algorithm to simplify the robust nonlinear H_∞ filter design. Based on system model, filter model and estimation error model, we could generate $\hat{x}(t)$, $\tilde{x}(t)$ and $x(t) = \hat{x}(t) + \tilde{x}(t)$ to train DNN to solve $\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ of $x(t)$-$\hat{x}(t)$-$\tilde{x}(t)$-coupled partial

differential HJIE by Adam learning algorithm by HJIE error $\epsilon(\theta_k(t))$ for H_∞ filter gain $k^*(\hat{x}(t))$, the worst-case external disturbance $v^*(t)$ and measurement noise $n^*(t)$. Therefore, with the help of system model and theoretical nonlinear H_∞ filtering result of HJIE, DNN is not only the conventional big data-driven classification and recognition schemes but also a system model-based H_∞ filtering scheme in nonlinear stochastic signal system with external disturbance and measurement noise. By the proposed nonlinear H_∞ DNN-based filter scheme, which can be considered as a co-design of H_∞ filtering scheme and DNN learning algorithm, we can save a large amount of training data and training time of DNN in the off-line training phase for the H_∞ HJIE-embedded DNN-based filter design of nonlinear stochastic signal systems. Therefore, the proposed DNN-based filter scheme could achieve the robust H_∞ filter performance of nonlinear stochastic signal system in (1) by solving HJIE in (9) directly, which is very difficult to solve analytically and numerically through conventional methods at present. We have also proven that when HJIE error of H_∞ filter approaches to 0, the proposed DNN-based filter scheme can approach to the theoretical H_∞ filter performance. For the practical design, an H_∞ HJIE-embedded DNN-based filter scheme is also proposed for sample-data nonlinear stochastic signal system in Fig. 3. Since the HJIE-embedded DNN could solve the more general nonlinear solution $V(x(t))$ of HJIE than the conventional quadratic optimal filters with the special quadratic solution $V(x(t)) = x^T(t)Px(t)$, the proposed H_∞ DNN-based filter could achieve a quite well filtering performance in the trajectory estimation performance of missile detection system by radar when compared with extended Kalman filter and particle filter. In future work, for more practical application, the proposed H_∞ DNN-based filter design will be extended to the DNN-based H_∞ state estimator-based output feedback reference tracking control of nonlinear stochastic signal system under external disturbance and output measurement noise.

APPENDIX A
PROOF OF THEOREM 1

Since the selection of $v(t)$ and $n(t)$ is independent on the specification of filter gain $k(\hat{x}(t))$ in (5), the minmax Nash game problem is equivalent to the following minmax problem [33], [34]

$$\min_{k(\hat{x}(t))}\max_{v(t),n(t)} E\left\{\int_0^{t_f}\left(\tilde{x}^T(t)Q\tilde{x}(t)-\rho v^T(t)v(t) -\rho n^T(t)n(t)\right)dt\right\} \le EV(\tilde{x}(0)) \quad \text{(A1)}$$

Let us denote

$$J = \min_{k(\hat{x}(t))}\max_{v(t),n(t)} E\left\{\int_0^{t_f}\left(\tilde{x}^T(t)Q\tilde{x}(t)-\rho v^T(t)v(t) -\rho n^T(t)n(t)\right)dt\right\} \quad \text{(A2)}$$

Then the robust H_∞ filter design problem needs to solve the minmax problem in (A2) at first and then the result J must be less than or equal to $EV(\tilde{x}(0))$.

By chain rule, we get

$$\frac{\partial V(\tilde{x}(t))}{\partial t} = \left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)^T \dot{\tilde{x}}(t) \quad \text{(A3)}$$

Integrating both sides of (A3) from 0 to t_f, we get

$$\int_0^{t_f}\left(-\frac{\partial V(\tilde{x}(t))}{\partial t}+\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)^T\dot{\tilde{x}}(t)\right)dt = 0 \quad \text{(A4)}$$

or

$$V(\tilde{x}(0)) - V(\tilde{x}(t_f)) + \int_0^{t_f}\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)^T\dot{\tilde{x}}(t)dt = 0 \quad \text{(A5)}$$

Substituting (A5) into (A2), we get

$$\begin{aligned} J = \min_{k(\hat{x}(t))}\max_{v(t),n(t)} & E\{V(\tilde{x}(0)) - V(\tilde{x}(t_f)) \\ & + \int_0^{t_f}(\tilde{x}^T(t)Q\tilde{x}(t)+\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)^T\dot{\tilde{x}}(t) - \rho v^T(t)v(t) \\ & - \rho n^T(t)n(t))dt\} \end{aligned} \quad \text{(A6)}$$

Then, by substituting (4) into (A6), we get

$$\begin{aligned} J = \min_{k(\hat{x}(t))}\max_{v(t),n(t)} & E\{V(\tilde{x}(0)) - V(\tilde{x}(t_f)) \\ & + \int_0^{t_f}(\tilde{x}^T(t)Q\tilde{x}(t)+\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)^T(A(x(t)) - A(\hat{x}(t)) \\ & + B(x(t))v(t) - k(\hat{x}(t))(C(x(t)) - C(\hat{x}(t))) \\ & - k(\hat{x}(t))n(t)) - \rho v^T(t)v(t) - \rho n^T(t)n(t))dt\} \end{aligned} \quad \text{(A7)}$$

and (A7) can be written as following compact form

$$\begin{aligned} J = \min_{k(\hat{x}(t))}\max_{v(t),n(t)} & E\{V(\tilde{x}(0)) - V(\tilde{x}(t_f)) \\ & + \int_0^{t_f}(\tilde{x}^T(t)Q\tilde{x}(t)+\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)^T(A(x(t)) - A(\hat{x}(t))) \\ & + \left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)^T B(x(t))v(t) - \left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)^T k(\hat{x}(t)) \\ & \times (C(x(t)) - C(\hat{x}(t))) - \left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)^T k(\hat{x}(t))n(t)) \\ & - \rho v^T(t)v(t) - \rho n^T(t)n(t))dt\} \end{aligned} \quad \text{(A8)}$$

By completing square technique of $v(t)$ and $n(t)$, we get

$$\begin{aligned} J = \min_{k(\hat{x}(t))}\max_{v(t),n(t)} & E\{V(\tilde{x}(0)) - V(\tilde{x}(t_f)) \\ & + \int_0^{t_f}(\tilde{x}^T(t)Q\tilde{x}(t)+\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)^T(A(x(t)) - A(\hat{x}(t))) \\ & - \frac{1}{\rho}(\rho v(t) - \frac{1}{2}B^T(x(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right))^T \\ & \times (\rho v(t) - \frac{1}{2}B^T(x(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)) \end{aligned}$$

$$+\frac{1}{4\rho}\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right)^T B(x(t))B^T(x(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right)$$
$$-\frac{1}{\rho}(\rho n(t)+\frac{1}{2}k^T(\hat{x}(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right))^T$$
$$\times(\rho n(t)+\frac{1}{2}k^T(\hat{x}(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right))$$
$$+\frac{1}{4\rho}\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right)^T k(\hat{x}(t))k^T(\hat{x}(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right)$$
$$-\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right)^T k(\hat{x}(t))(C(x(t))-C(\hat{x}(t)))dt\} \quad \text{(A9)}$$

By completing square techniques of $k(\hat{x}(t))$, we get

$$J=\min_{k(\hat{x}(t))}\max_{v(t),n(t)} E\{V(\tilde{x}(0))-V(\tilde{x}(t_f))$$
$$+\int_0^{t_f}(\tilde{x}^T(t)Q\tilde{x}(t)+\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right)^T(A(x(t))-A(\hat{x}(t)))$$
$$+\frac{1}{4\rho}\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right)^T B(x(t))B^T(x(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right)$$
$$-\frac{1}{\rho}(\rho v(t)-\frac{1}{2}B^T(x(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right))^T$$
$$\times(\rho v(t)-\frac{1}{2}B^T(x(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right))$$
$$-\frac{1}{\rho}(\rho n(t)+\frac{1}{2}k^T(\hat{x}(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right))^T$$
$$\times(\rho n(t)+\frac{1}{2}k^T(\hat{x}(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right))$$
$$+\rho(\frac{1}{2\rho}k^T(\hat{x}(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right)-(C(x(t))-C(\hat{x}(t)))^T$$
$$\times(\frac{1}{2\rho}k^T(\hat{x}(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right)-(C(x(t))-C(\hat{x}(t)))$$
$$-\rho(C(x(t))-C(\hat{x}(t)))^T(C(x(t))$$
$$-C(\hat{x}(t)))dt\} \quad \text{(A10)}$$

By the HJIE in (9), we get

$$J=\min_{k(\hat{x}(t))}\max_{v(t),n(t)} E\{V(\tilde{x}(0))-V(\tilde{x}(t_f))$$
$$+\int_0^{t_f}(\rho(\frac{1}{2\rho}k^T(\hat{x}(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right)$$
$$-(C(x(t))-C(\hat{x}(t)))^T(\frac{1}{2\rho}k^T(\hat{x}(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right)$$
$$-(C(x(t))-C(\hat{x}(t))))-\frac{1}{\rho}(\rho v(t)-\frac{1}{2}B^T(x(t))$$
$$\times\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right))^T(\rho v(t)-\frac{1}{2}B^T(x(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right))$$
$$-\frac{1}{\rho}(\rho n(t)+\frac{1}{2}k^T(\hat{x}(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right))^T$$
$$\times(\rho n(t)+\frac{1}{2}k^T(\hat{x}(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right))dt\} \quad \text{(A11)}$$

Then we obtain the worst-case $v^*(t)$ and $n^*(t)$ as well as the optimal H_∞ filter gain $k^*(\hat{x}(t))$ in (6)-(8) and

$$J=E\left\{V(\tilde{x}(0))-V(\tilde{x}(t_f))\right\}\le EV(\tilde{x}(0))$$

which satisfies with (A1). Q.E.D.

APPENDIX B
PROOF OF THEOREM 2

Suppose there exists an error function $e(\tilde{x}(t))$ between $\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right)_\epsilon$ of $HJIE_\epsilon$ in (16) and $\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}$ of $HJIE$ in (9) as follows

$$\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right)_\epsilon=\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}+e(\tilde{x}(t)) \quad \text{(B1)}$$

By the fact $HJIE=0$ in (9), from (16)

$$\epsilon(\theta_k(t))$$
$$=HJIE_\epsilon-HJIE$$
$$=\left(\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right)_\epsilon^T-\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right)^T\right)$$
$$\times(A(x(t))-A(\hat{x}(t)))$$
$$+\frac{1}{4\rho}\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right)_\epsilon^T B(x(t))B^T(x(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right)_\epsilon$$
$$-\frac{1}{4\rho}\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right)^T B(x(t))B^T(x(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right) \quad \text{(B2)}$$

Substituting (B1) into (B2), we get

$$\epsilon(\theta_k(t))=e^T(\tilde{x}(t))(A(x(t))-A(\hat{x}(t)))$$
$$+\frac{1}{4\rho}e^T(\tilde{x}(t))B(x(t))B^T(x(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right)$$
$$+\frac{1}{4\rho}\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right)^T B(x(t))B^T(x(t))e(\tilde{x}(t))$$
$$+\frac{1}{4\rho}e^T(\tilde{x}(t))B(x(t))B^T(x(t))e(\tilde{x}(t)) \quad \text{(B3)}$$

By the symmetry of

$$e^T(\tilde{x}(t))B(x(t))B^T(x(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right)$$
$$=\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right)^T B(x(t))B^T(x(t))e(\tilde{x}(t)) \quad \text{(B4)}$$

Then $\epsilon(\theta_k(t))$ in (B3) becomes

$$\epsilon(\theta_k(t))=e^T(\tilde{x}(t))(A(x(t))-A(\hat{x}(t)))$$
$$+\frac{1}{2\rho}e^T(\tilde{x}(t))B(x(t))B^T(x(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial\tilde{x}(t)}\right)$$
$$+\frac{1}{4\rho}e^T(\tilde{x}(t))B(x(t))B^T(x(t))e(\tilde{x}(t))$$

$$= e^T(\tilde{x}(t))[(A(x(t)) - A(\hat{x}(t))) + \frac{1}{2\rho}B(x(t))B^T(x(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right) + \frac{1}{4\rho}B(x(t))B^T(x(t))e(\tilde{x}(t))] \quad \text{(B5)}$$

As $\epsilon(\theta_k(t)) \to 0$, then

$$e^T(\tilde{x}(t))[(A(x(t)) - A(\hat{x}(t))) + \frac{1}{2\rho}B(x(t))B^T(x(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right) + \frac{1}{4\rho}B(x(t))B^T(x(t))e(\tilde{x}(t))] \to 0 \quad \text{(B6)}$$

Since the term in $[\cdot]$ in the above equation is different from $HJIE = 0$, i.e.,

$$(A(x(t)) - A(\hat{x}(t))) + \frac{1}{2\rho}B(x(t))B^T(x(t))\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right) + \frac{1}{4\rho}B(x(t))B^T(x(t))e(\tilde{x}(t)) \neq HJIE = 0$$

it will be not equal to zero for all $x(t)$, $\hat{x}(t)$ and $\tilde{x}(t)$. Therefore, from (B6) we can conclude that $e(\tilde{x}(t)) \to 0$ as $\epsilon(\theta_k(t)) \to 0$. From (B1), it implies that $\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)_\epsilon \to \frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}$ as $\epsilon(\theta_k(t)) \to 0$ in the Adam learning algorithm. From $HJIE_\epsilon$ in (16) and $HJIE$ in (9), it is seen that $HJIE_\epsilon \to HJIE$ as $\epsilon(\theta_k(t)) \to 0$. According to Theorem 1, the filter gain $k^\epsilon(\hat{x}(t)) = \frac{2\rho}{\left\|\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right\|^2_\epsilon}\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)_\epsilon (C(x(t)) - C(\hat{x}(t)))^T$ based on the output $\left(\frac{\partial V(\tilde{x}(t))}{\partial \tilde{x}(t)}\right)_\epsilon$ of DNN of HJIE-embedded DNN-based filter in Fig. 1 will approach to the H_∞ filter gain $k^*(\hat{x}(t))$ in (8) of the Luenberger-type H_∞ filter in (2) for the nonlinear stochastic signal system in (1). Q.E.D.

REFERENCES

[1] M. J. Grimble and A. El Sayed, "Solution of the H_∞ optimal linear filtering problem for discrete-time systems," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 7, pp. 1092–1104, Jul. 1990.
[2] W. M. Mceneaney, "Robust H_∞ filtering for nonlinear system," *Syst. Control Lett.*, vol. 33, no. 5, pp. 315–325, Apr. 1998.
[3] W. Zhang, B. S. Chen, and C. S. Tseng, "Robust H_∞ filtering for nonlinear stochastic systems," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 589–598, Feb. 2005.
[4] X. Zhang, S. He, V. Stojanovic, X. Luan, and F. Liu, "Finite-time asynchronous dissipative filtering of conic-type nonlinear Markov jump systems," *Sci. China Inf. Sci.*, vol. 64, pp. 152206:1–152206:12, May 2021.
[5] P. Cheng, M. Chen, V. Stojanovic, and S. He, "Asynchronous fault detection filtering for piecewise homogenous Markov jump linear systems via a dual hidden Markov model," *Mech. Syst. Signal Process.*, vol. 151, Apr. 2021, Art. no. 107353.
[6] P. Cheng, S. He, V. Stojanovic, X. Luan, and F. Liu, "Fuzzy fault detection for Markov jump systems with partly accessible hidden information: An event-triggered approach," *IEEE Trans. Cybern.*, early access, Jan. 29, 2021, doi: 10.1109/TCYB.2021.3050209.
[7] W. Zhang, B. S. Chen, L. Sheng, and M. Gao, "Robust H_2/H_∞ filter design for a class of nonlinear stochastic systems with state-dependent noise," *Math. Problems Eng.*, vol. 2012, pp. 1–15, Oct. 2012.
[8] C. Antoniou, M. Ben-Akiva, and H. N. Koutsopoulos, "Nonlinear Kalman filtering algorithms for on-line calibration of dynamic traffic assignment models," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 4, pp. 661–670, Dec. 2007.
[9] S. Sarkka, "On unscented Kalman filtering for state estimation of continuous-time nonlinear systems," *IEEE Trans. Autom. Control*, vol. 52, no. 9, pp. 1631–1641, Sep. 2007.
[10] V. Filipovic, N. Nedic, and V. Stojanovic, "Robust identification of pneumatic servo actuators in the real situations," *Forschung Ingenieurwesen*, vol. 75, no. 4, pp. 183–196, Dec. 2011.
[11] E. E. Tsakonas, N. D. Sidiropoulos, and A. Swami, "Optimal particle filters for tracking a time-varying harmonic or chirp signal," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4598–4610, Oct. 2008.
[12] S. Yin and X. Zhu, "Intelligent particle filter and its application on fault detection of nonlinear system," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3852–3861, Jun. 2015.
[13] S. K. Nguang and W. Assawinchaichote, "H_∞ filtering for fuzzy dynamical systems with D stability constraints," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 50, no. 11, pp. 1503–1508, Nov. 2003.
[14] W. Assawinchaichote and S. K. Nguang, "H_∞ filtering for fuzzy singularly perturbed systems with pole placement constraints: An LMI approach," *IEEE Trans. Signal Process.*, vol. 52, no. 6, pp. 1659–1667, Jun. 2004.
[15] H. J. Gao, Y. Zhao, L. Lam, and K. Chen, "H_∞ fuzzy filtering of nonlinear systems with intermittent measurements," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 2, pp. 291–300, Apr. 2009.
[16] B. S. Chen, W. H. Chen, and H. L. Wu, "Robust H_2/H_∞ global linearization filter design for nonlinear stochastic systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 7, pp. 1441–1454, Jul. 2009.
[17] B.-S. Chen and C.-F. Wu, "Robust scheduling filter design for a class of nonlinear stochastic Poisson signal systems," *IEEE Trans. Signal Process.*, vol. 63, no. 23, pp. 6245–6257, Dec. 2015.
[18] C. S. Tseng, "Robust fuzzy filter design for a class of nonlinear stochastic systems," *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 2, pp. 261–274, Apr. 2007.
[19] C. S. Tseng and B. S. Chen, "H_∞ fuzzy estimation for a class of nonlinear discrete-time dynamic systems," *IEEE Trans. Signal Process.*, vol. 49, no. 11, pp. 2605–2619, Nov. 2001.
[20] B. S. Chen, H. C. Lee, and C. F. Wu, "Pareto optimal filter design for nonlinear stochastic fuzzy systems via multiobjective H_2/H_∞ optimization," *IEEE Trans. Fuzzy Systems.*, vol. 23, no. 2, pp. 387–399, Apr. 2015.
[21] H. J. Gao, J. Lam, L. H. Xie, and C. H. Wang, "New approach to mixed H_2/H_∞ filtering for polytopic discrete-time systems," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 3183–3192, Aug. 2005.
[22] B. S. Chen, M. Y. Lee, and X. H. Chen, "Security-enhanced filter design for stochastic systems under malicious attack via smoothed signal model and multiobjective H_2/H_∞ estimation method," *IEEE Trans. Signal Process.*, vol. 68, pp. 4971–4986, 2020.
[23] A. L. Maas, P. Qi, Z. Xie, and A. Y. Hannun, "Building DNN acoustic models for large vocabulary speech recognition," *Comput. Speech Lang.*, vol. 41, pp. 195–213, Jan. 2017.
[24] 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.
[25] 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, Feb. 2016.
[26] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3642–3649.
[27] 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.*, Jaipur, India, 2017, pp. 162–167.
[28] A. E. Eshratifar, A. Esmaili, and M. Pedram, "BottleNet: A deep learning architecture for intelligent mobile cloud computing services," in *Proc. ISLPED*, Lausanne, Switzerlan, 2019, pp. 1–6.
[29] 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. MICRO*, Fukuok, Japan, 2018, pp. 189–202.
[30] M. Z. Alom, T. M. Taha, and C. Yakopcic, "The history began from AlexNet: A comprehensive survey on deep learning approaches," in *Proc. CVPR*, Salt Lake City, UT, USA, Mar. 2018, pp. 1–39.
[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, San Diego, CA, USA, Dec. 2014, pp. 1–15.

[32] Y. Dauphin, H. De Vries, and Y. Bengio, "Equilibrated adaptive learning rates for non-convex optimization," 2015, *arXiv:1502.04390*. [Online]. Available: https://arxiv.org/abs/1502.04390
[33] W. Zhang, L. Xie, and B. S. Chen, *Stochasitc H_2/H_∞ Control: A Nash Game Approach*. Boca Raton, FL, USA: CRC Press, 2017.
[34] B. S. Chen, *Stochastic Game Strategies and Their Applications*. Boca Raton, FL, USA: CRC Press, 2020.
[35] G. M. Siouris, G. Chen, and J. Wang, "Tracking an incoming ballistic missile using an extended interval Kalman filter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 33, no. 1, pp. 232–240, Jan. 1997.
[36] S. R. Kou, D. L. Elliott, and T. J. Tarn, "Observability of nonlinear systems," *Inf. Control*, vol. 22, pp. 89–99, Feb. 1973.
[37] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," Nov. 2015, *arXiv:1505.00853*.
[38] X. Wang, D. Ding, X. Ge, and Q. Han, "Neural-network-based control for discrete-time nonlinear systems with denial-of-service attack: The adaptive event-triggered case," *Int. J. Robust Nonlinear Control*, to be published, doi: 10.1002/rnc.5831.

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, Chungli, 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 at the Tatung Institute of Technology, from 1973 to 1987. He is currently the Tsing Hua Distinguished Chair Professor of electrical engineering and computer science at National Tsing Hua University, Hsinchu, Taiwan. His current research interests include control engineering, signal processing, and systems biology. He was a recipient of the Distinguished Research Award from the National Science Council of Taiwan four times. He is the National Chair Professor of the Ministry of Education of Taiwan.

PO-HSUN WU received the B.S. degree in electrical engineering from National Chung Cheng University, Chiayi, Taiwan, in 2018. He is currently pursuing the master's degree in electrical engineering with National Tsing Hua University, Hsinchu, Taiwan. His current research interests include robust control, fuzzy control, and non-linear stochastic systems.

MIN-YEN LEE received the B.S. degree in bio-industrial mechatronics engineering from National Chung Hsing University, Taichung, 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 non-linear stochastic systems.