Two-Stage Neural Observer for Mechanical Systems
Juan Resendiz, Wen Yu, and Leonid Fridman

Abstract—This paper proposes a novel velocity observer which uses neural network and sliding mode for unknown mechanical systems. The neural observer in this paper has two stages: 1) a dead-zone neural observer assures that the observer error is bounded and 2) a super-twisting second-order sliding-mode is used to guarantee finite time convergence of the observer. With sliding mode compensation, the two-stage neural observer ensures finite time convergence, and reduces the chattering during its discrete realization.

Index Terms—Finite time convergence, neural observer, second-order sliding mode.

I. INTRODUCTION

STATE estimation is one of the most important problems in control theory. Because of Coulomb friction, linear observer does not guarantee a zero steady-state error for mechanical systems. While model-based nonlinear observer can remove this error, it is usually restricted to the case that the model is exactly known. When the plant parameters are unknown, nonlinear adaptive observer can be applied [1]. High-gain differentiators was realized if the plant is completely unknown [2]. But it is not exact with any fixed finite gain and features peaking effect. To avoid these problems, sliding mode observers are developed [see [3] and references therein]. In order to obtain finite-time convergence, robustness with respect to uncertainties and a possibility of uncertainty estimation, usually sliding mode observers require partial knowledge and relative degree of the system with respect to the unknown inputs [4].

A new generation of observers based on second-order sliding-mode algorithms has been recently developed by [5]. In [6], super-twisting and robust exact differentiator techniques were used to guarantee finite time convergence. In [7] an observer basing on “broken” super-twisting algorithm, reconstructing the velocity of mechanical systems, was proposed. But they require to know the nominal part of the system and an upper bound for the acceleration.

When we do not have complete information, a model-free nonlinear observer is needed. If the nonlinear system is given in normal linearized form, high-gain observers may estimate the derivative of the output [2], but such filters loose their capabilities in presence of unmeasurable perturbations in the outputs. More recently, in [8], first-order sliding-mode observer, which does not requires any knowledge of the system, can obtain infinite time convergence. The sliding mode gain should be bigger than the upper bound of the unknown nonlinear function. Neural networks can be considered as an alternative model-free observer because they offer potential benefit for nonlinear modeling [9]. For example, dynamic neural networks have been applied to design a Luenberger-like observer [10]. Due to neural modeling error, neural observers are not asymptotically stable. Normal combinations of neural networks and sliding mode methods are to apply them at same time, where the sliding mode observer is used to compensate modeling error of the neural observer [4]. This type of neural observers cannot assure finite time convergence [10]. In this paper, neural observer and sliding mode compensator are connected serially, it is called two-stage neural observer. The neural network is used to approximate the nonlinear function of the mechanical system. A dead-zone training algorithm is applied for the neural observer. After the observer error enters the dead-zone, a super-twisting second-order sliding-mode is used to guarantee finite time convergence of the neural observer. This type of observer can ensure finite time convergence of neural observers and less chattering of sliding-mode observers.

II. PROBLEM STATEMENT

Generally, a second-order mechanical system has the form

\[ M(q)\ddot{q} + C(q, \dot{q})\dot{q} + P(\dot{q}) + G(q) + d(t, q, \dot{q}) = \tau \] (1)

where \( q \in \mathbb{R}^n \) is the state vector, \( M(q) \) is the inertia matrix, \( C(q, \dot{q}) \) is the Coriolis and centrifugal forces matrix, \( P(\dot{q}) \) is the Coulomb friction, \( G(q) \) is the term of gravitational forces, \( d(t, q, \dot{q}) \) is an uncertainty disturbance and \( \tau \) is the torque produced by the actuators. Introducing \( x_1 = q \in \mathbb{R}^n, x_2 = \dot{q} \in \mathbb{R}^n \), this system can be rewritten in the state-space form

\[ \dot{x}_1 = x_2 \]
\[ \dot{x}_2 = f(t, x, u) + \xi(t, x, u) \]
\[ y = Cx \] (2)

where \( y \in \mathbb{R}^m, C \in \mathbb{R}^{m \times 2n} \) are known. For observer case, we let \( m = n \times \bar{x} = [x_1^T, x_2^T]^T \in \mathbb{R}^{2n}, u = \tau \) is a scalar, and

\[ f(t, x, u) = -M(x_1)^{-1}(C(x)x_2 + P(x_2) + G(x_1) - u) \] (3)

while the uncertainties are concentrated in the term \( \xi(t, x, u) \in \mathbb{R}^n \). The solutions of the system (2) are understood in Filippov’s sense [6]. For our purpose, the following assumptions are required.

A1) The system states are bounded for all time.
A2) \( f \) and the uncertainty term \( \xi = \xi(t, x, u) \) are Lebesgue-measurable and uniformly bounded in any compact region of the state space \( x_1, x_2 \), with

\[ ||\xi(t)||_{\mathcal{L}_\xi}^2 = \xi_T \Lambda_\xi \xi \leq \bar{\xi} < \infty \quad \Lambda_\xi = \Lambda_\xi^T > 0 \] (4)
where the normalizing matrix $\Lambda_C$ is introduced to represent different physical natures, it is given in advance. The upper bound $\xi_t$ can be regarded as a power of the corresponding perturbation.

When $f(t,x,u)$ is known, the sliding mode observer proposed in [7] has a form of

$$
\dot{\hat{x}}_1 = \dot{\hat{x}}_2 + \lambda |\hat{x}_1|^{1/2} \text{sign}(\hat{x}_1), \\
\dot{\hat{x}}_2 = f(t,\hat{x}_1, \hat{x}_2,u) + \alpha \text{sign}(\hat{x}_1)
$$

(5)

where $\dot{\hat{x}}_1$ and $\dot{\hat{x}}_2$ are the state estimations, $\hat{x}_1 = x_1 - \hat{x}_1$. The sliding mode gain $\alpha$ should be big enough, such that it can cancel the uncertainty $\xi_t$. Since it requires the complete knowledge about $f$ defined in (3), it is very difficult, especially for the Coulomb friction $F(x_2)$. A big advantage of this observer is that it can arrive finite time convergence.

When $f(t,x,u)$ is unknown, the uncertainty $\xi_t$ should include $f(t,x,u)$, the sliding mode observer has the form as in [8]. It is similar with (5), but without the term of $f(t,x,u)$. The sliding mode gain $\alpha$ and the chattering are much bigger than [7]. The model-free sliding-mode observer does not require the model $f$. But it cannot arrive finite time convergence.

Nevertheless it is not always possible to have a good knowledge on the system, so the chattering phenomenon is inevitable. If we have incomplete information about the nominal nonlinear function $f(t,x,u)$, and of course $\xi_t$, the chattering will be decreased. It seems to be natural to construct its estimate $\hat{f}(t,\hat{x}_1,\hat{x}_2)W_t$ depending on a parameter $W_t$, which can be adjusted online by means of an updating law

$$
\dot{W}_t = \Phi(t,\hat{x}_1,\hat{x}_2,W_t, y).
$$

(6)

In this paper, we will use neural networks to approximate $f(t,x,u)$ and construct a neural observer. In fact, many estimators can be used, e.g., traditional linear estimator has nice linearity-in-the-parameters property. But [11] shows that worst case errors in linear approximation are larger than those in approximation by neural networks, although neural nets are more complex and have problems of non-convexity and local minima. Another advantage of neural estimator is the minimal number of neural units can be estimated to guarantee a desired approximation accuracy. Reference [12] proved that the feedforward networks with one layer of sigmoidal nonlinearities, which will be applied in this paper, achieve integrated squared error of order $O((1)/(p))$, where $p$ is the number of neurons. Reference [13] proved that the convergence rates of the neural estimator are the better, the smoother the activation functions are. In this paper, we do not care about the exact accuracy of neural estimator, the remaining error of the neural observer will be compensated by a sliding mode observer.

The object of this paper is to design a neural observer for mechanical systems, which use a neural estimator and a sliding mode observer sequentially, to estimate the unmeasurable velocity based on the position measurement. The sliding mode observer ensures finite time convergence and the neural network will approximate the nominal nonlinear function of the mechanical system.

III. TWO-STAGE NEURAL OBSERVER

The neural observer proposed in this paper has the following form:

$$
\dot{\hat{x}} = A\hat{x} + BW_t\sigma(\hat{x}) + (1 - s_t)Z_t, \hat{y} = C\hat{x}
$$

(7)

where $\hat{x} \in \mathbb{R}^{2n}$ is the state of the estimation vector, $A \in \mathbb{R}^{2n \times 2n}$ is a stable fixed matrix which will be specified later.

The model-free sliding-mode observer does not require the model $f$. By proper choice of the vectors $a, b \in \mathbb{R}^n$, the matrix $A$ can be Hurwitz stable. Here, $s_t$ is a switch variable, it will switch between the neural estimator (7) and the second-order sliding mode observer (10), based on the output error $e_t = y - \hat{y} \in \mathbb{R}^n$.

$$
s_t = \begin{cases} 1, & \text{if } \|e_t\|^2_{Q_1} \geq \gamma \\
0, & \text{if } \|e_t\|^2_{Q_2} < \gamma
\end{cases}
$$

(9)

where $Q_1 = Q_1^T > 0$ is a known matrix which will be specified later, $\gamma$ is known upper bound of the neural modeling error, which will be defined in (19). The second-order sliding mode observer is

$$
z_1 = k_1 |x_1 - \hat{x}_1|^{1/2} \text{sign}(x_1 - \hat{x}_1),
$$

$$
z_2 = k_2 \text{sign}(x_1 - \hat{x}_1)
$$

(10)

where $k_1$ and $k_2$ are the sliding mode gains, they will be determined by the theorem in the next section. The learning algorithm (6) is dead-zone one

$$
\dot{W}_t = s_t \Phi(t,\hat{x}_1,\hat{x}_2,W_t, y).
$$

(11)

If $\|e_t\|^2_{Q_1} \geq \gamma, s_t = 1$, the observer is pure neural observer, (7) becomes

$$
\dot{\hat{x}} = A\hat{x} + BW_t\sigma(\hat{x}), \hat{y} = C\hat{x}.
$$

(12)

With neural learning law (6), $BW_t\sigma(\hat{x})$ will approximate $f(t,x,u) + \xi(t,x,u)$, and $\|e_t\|^2_{Q_1}$ will be decrease. If after time $t_0, \|e_t\|^2_{Q_1} < \gamma, s_t = 0$. From (11), we know $\dot{W} = W_t$. The observer (7) become pure sliding mode

$$
\dot{\hat{x}} = A\hat{x} + BW_t\sigma(\hat{x}) + Z_t, \hat{y} = C\hat{x}.
$$

(13)

(7) can be also rewritten as

$$
\dot{\hat{x}}_1 = \dot{\hat{x}}_2 + (1 - s_t)z_1
$$

$$
\dot{\hat{x}}_2 = A_1\hat{x} + \dot{W}_t\sigma(\hat{x}) + (1 - s_t)z_2
$$

$$
\hat{y} = C\hat{x}
$$

(14)
with \( A_1 = [a \ b] \). We define observer error \( \Delta = x - \hat{x} \in \mathbb{R}^{2n} \). So the output error is \( e_t = C\Delta \), which implies that

\[
CTe_t = CTC\Delta - \delta I \Delta + \delta I \Delta
\]

\[
CTe_t = (CTC + \delta I)\Delta - \delta I \Delta
\]

\[
\Delta = C\delta A_1 \hat{e}_t + \delta N_\delta \Delta
\]  

(15)

where \( C \in \mathbb{R}^{m\times 2n}, C^+ \in \mathbb{R}^{2n\times m}, N_\delta \in \mathbb{R}^{2n\times 2n}, \delta \) is a small positive scalar, \( C^+ \) and \( N_\delta \) are defined as

\[
C^+ = (CTC + \delta I)^{-1}CT \quad N_\delta = (CTC + \delta I)^{-1}. \]  

(16)

Because sigmoid function \( \sigma(\cdot) \) satisfies Lipschitz condition

\[
\sigma^T A_1 \sigma \leq \Delta^T A_\sigma \Delta \]  

(17)

where \( \sigma = \sigma(x_t) - \sigma(\hat{x}_t), A_1 = A_1^T > 0, A_\sigma = A_\sigma^T > 0 \), and \( A_1 \) and \( A_\sigma \) are any positive definite matrices which can be selected by users. Adding and subtracting \( Ax \) to system (2), we have

\[
\dot{x}_1 = x_2 \quad \dot{x}_2 = A_1 x + g(t, x, u) \]  

(18)

where \( g(t, x, u) = f(t, x, u) - A_1 x + \xi \). According to the Stone–Weierstrass theorem \([14]\), the linear smooth function \( g(t, x, u) \) can be written as [15]

\[
g(t, x, u) = W^0 \sigma(x) + \eta_t\]  

where \( W^0 \) is a fixed weight matrix of the neural network. Then, we can rewrite (18) as

\[
\dot{x}_1 = x_2 \quad \dot{x}_2 = A_1 x + W^0 \sigma(x) + \gamma_t. \]  

(19)

By the assumption A1, \( x \) is restricted to a compact set \( S \) of \( x \in \mathbb{R}^n \). By the assumption A2, \( f_t \) and \( e_t \) are uniformly bounded. Taking into account that the sigmoid function \( \sigma(\cdot) \) is uniformly bounded, there exists a known positive constant \( \gamma \), such that

\[
||\gamma||^2_{A_2} = \gamma^T A_2 \gamma \leq \gamma \quad A_2 = A_2^T > 0. \]  

(20)

Now (19) can be expressed in matrix form

\[
\dot{x}_t = A_1 x + B W_0 \sigma(x) + \Gamma \gamma_t \]  

(21)

where \( A \) and \( B \) are defined as in (8), and \( \Gamma = [0 \ 1]^T \). It is well known [16] that if the matrix \( A \) is stable, for a matrix \( R \) the pair \( (A, R^{1/2}) \) is controllable, the pair \( (Q^{1/2}, A) \) is observable, and local frequency condition is fulfilled, then the algebraic Riccati equation

\[
A^T P + PA + PRP = -Q \]  

(22)

has a positive solution \( P \). We can make the following assumption:

**A3:** For a given stable matrix \( A \) there exists a strictly positive defined matrix \( Q \) such that the Riccati equation (22) with \( R = \hat{W} + \Gamma + \Lambda_N \delta, \hat{W} = BW_0 A_1^{-1}(W_0^T)BT \) has solution, and

\[
Q_0 = Q - A_\sigma > 0 \]  

(23)

here \( A_1 \) and \( A_\sigma \) are defined in (17), \( \Gamma = \Lambda_{N_\delta}^{-1} \Gamma_\sigma, A_2 \) is defined in (20), \( \Gamma \) is defined in (21), \( A_N \delta = \delta^2 N_\delta B A_{N_\delta}^{-1} BT N_\delta \Lambda \) and \( A_3 \) are positive definite matrices, \( N_\delta \) is defined in (16).

### IV. Stability Analysis

The stable learning law for the neural network is given by the following matrix differential equation

\[
\dot{W}_t = -s_t K \left[ PB \sigma^T(\hat{x}) C^+ e_t + \frac{\delta^2}{2} \sigma(\hat{x}) \hat{W}_t B A_3 B \sigma(\hat{x}) \right] \]  

(24)

where \( K \in \mathbb{R}^{2m\times 2m}, P \in \mathbb{R}^{2n\times 2n} \) is the solution of equation (22), \( C^+ \in \mathbb{R}^{2n\times m}, \hat{W}_t, \hat{W}_t \in \mathbb{R}^{2n}, \sigma \in \mathbb{R}^{2n}, A_3 \in \mathbb{R}^{2n\times 2n}, \delta_1 \) and \( s_t \) are scalars. In fact, we do not need to solve the Riccati (22), we can define the learning gain \( K \) as \( K_1 \), and select a positive defined matrix \( K_1 \) as the learning gain. The following theorem gives the stability and finite time convergence for the observer (7).

**Theorem 1:** Under the assumptions A1 and A2, the switch policy (9) and neural training law (24), if the sliding mode gains of the two-stage neural observer (7) satisfy

\[
k_2 > f^+, k_1 > (k_2 + f^+) \sqrt{\frac{2}{k_2 - f^+} \frac{1+p}{1-p}} \]  

(25)

where \( p \) is some chosen constant, \( 0 < p < 1, f^+ \) is the upper bound of the neural modeling error when the weight of the neural networks is fixed as \( W_1, W_1 = W_{t_0}, \) at time \( t = t_0, ||e_t||_{Q_2} < \gamma \)

\[
Q_1 = (I - \delta N_\delta)^{-1}C^+ T Q_0 C^+ (I - \delta N_\delta)^{-1} \]  

then the observer is stable, i.e., \( ||\Delta t|| \in L_{\infty}, W_t \in L_{\infty}, \) and the states of the observer (7) converge to the states of system (2) in finite time.

**Proof:** The two-stage neural observer switches between two models: (12) and (13). Now we discuss these two cases: I) if \( ||x_t||_{Q_2} \geq \gamma \), the observer (7) becomes (12). From (19) and (12), the estimation error can be expressed as

\[
\Delta_1 = \Delta_2 \quad \Delta_2 = A_1 \Delta + \hat{W}_t \sigma(\hat{x}) + W^0 \sigma + \gamma_t \]  

(26)

where \( \hat{W}_t = W^0 + \hat{W}_t \). Or in matrix form

\[
\dot{\Delta}_t = A \Delta_t + B \hat{W}_t \sigma(\hat{x}) + BW^0 \sigma + \Gamma \gamma_t \]  

(27)

where \( \Gamma \) is defined in (21). We define the Lyapunov function candidate as

\[
V = \Delta^T_t \ P \Delta_t + \text{tr} \{ \hat{W}^T K_1^{-1} \hat{W}_t \} \]  

(28)

where \( K_1 = K_1^T > 0 \). Calculating its time derivative, we obtain

\[
\dot{V} = 2\Delta^T \dot{P} \Delta_t + 2\text{tr} \left\{ \hat{W}_t^T K_1^{-1} \hat{W}_t \right\} \]  

(29)

where

\[
2\Delta^T \dot{P} \Delta = \Delta^T (A^T P + PA) \Delta
\]

\[+ 2\Delta^T \dot{P} \hat{W}_t \sigma(\hat{x}) + 2\Delta^T \dot{P} BW^0 \sigma + 2\Delta^T \dot{P} T \gamma_t. \]  

(30)

Using the matrix inequality [10]

\[
X^T Y + (XT Y)^T \leq X^T A^{-1} X + Y^T A Y \]  

(31)
which is valid for any $X, Y \in \mathbb{R}^{n \times d}$, and for any positive defined matrix $\Lambda = \Delta^T \in \mathbb{R}^{n \times d}$, we can estimate $2\Delta^T P B W^0 \tilde{\sigma}$ as

$$2\Delta^T P B W^0 \tilde{\sigma} \leq \Delta^T (P W \tilde{W} + \Lambda \sigma) \Delta$$  \hspace{1cm} (32)

where $\tilde{W} = B W^0 \Lambda_1^{-1} (W^0)^T B^T$. Similar from A2, we can estimate $2\Delta^T P T \Gamma t$ as

$$2\Delta^T P T \Gamma t \leq \Delta^T P T P \Delta + \gamma$$  \hspace{1cm} (33)

where $\Gamma = \Gamma \Lambda_2^{-1}$. Since $\Delta$ is not available, only $e_t$ can be used in the updating law. $2\Delta^T P B W_t \tilde{\sigma}(\hat{x}_t)$ becomes

$$2(C^t + e_t)^T P B W_t \tilde{\sigma}(\hat{x}_t) + 2\delta \Delta^T P N \delta B W_t \tilde{\sigma}(\hat{x}_t).$$

Using again (31), $2\delta \Delta^T P N \delta B W_t \tilde{\sigma}(\hat{x}_t)$ is estimated as

$$\Delta^T P \Lambda_3 P \Delta + \frac{\delta^2 \sigma^2(\hat{x}_t)}{\tilde{W}_t} B^T \Lambda_3 B W_t \tilde{\sigma}(\hat{x}_t),$$

where $\Lambda_3 = \delta^2 \Lambda_3 B \Lambda_3^{-1} B^T \Lambda_3$. Since in (32) and (33), (29) can be rewritten as

$$\dot{V} \leq \Delta^T (A T + P A + P (W + \Gamma + \Lambda_{N_t}) + \Lambda \sigma) \Delta + 2(C^t + e_t)^T P B W_t \tilde{\sigma}(\hat{x}_t) + \frac{\delta^2 \sigma^2(\hat{x}_t)}{\tilde{W}_t} B^T \Lambda_3 B W_t \tilde{\sigma}(\hat{x}_t)$$

$$+ 2t \tau [\tilde{W}_t K_{1}^{-1} \tilde{W}_t] + \gamma.$$  \hspace{1cm} (34)

Taking account of A3 and using the learning law (24), now $s_t = 1$, we have

$$\dot{V} \leq -\Delta^T Q_0 \Delta + \gamma$$  \hspace{1cm} (35)

where $Q_0 = Q - \sigma \Lambda$. Now we estimate the bound of $e_t$. Using again (15), $(I - \delta N \sigma) \Delta = C^t e_t$, we can rewrite (35) as

$$\dot{V} \leq -e_t Q_1 e_t + \gamma$$  \hspace{1cm} (36)

where $Q_1 = (I - \delta N \sigma)^{-1} C^t Q_0 C^t (I - \delta N \sigma)^{-1}$. Since in this case $||e_t||_{Q_1} \geq \gamma$, then $\dot{V} \leq 0$, $V$ is bounded. So $||\Delta||$ and $||W_t||$ are bounded. If we integrate (36) from 0 up to $T$, it yields

$$V_T - V_0 \leq \int_0^T e_t Q_1 e_t dt + \gamma T.$$  \hspace{1cm} (37)

Because $V_T \geq 0$, we have

$$\frac{1}{T} \int_0^T ||e_t||_{Q_1}^2 \, dt \leq \gamma + \frac{V_0}{T}.$$  \hspace{1cm} (37)

So, $||e_t||_{Q_1}$ will converge to the ball with radius $\gamma$. II) if at time $t = t_0$, $||e_t||_{Q_1} < \gamma$, from (24) we know the weights become constants, $W_1 = W_1$. Equation (19) can be written as

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = A_1 \dot{x}_t + W_1 \tilde{\sigma}(\dot{x}_t) + F_t$$  \hspace{1cm} (38)

where $F_t$ is neural modeling error when the weight of the neural networks is fixed as $W_1$. The error dynamics is obtained from

$$\dot{\Delta}_1 = \Delta_2 - z_1$$  \hspace{1cm} (39)

$$\dot{\Delta}_2 = F_t - z_2.$$  \hspace{1cm} (39)

We define the upper bound of $F_t$ as $f^+$, that is $|F_t| \leq f^+$. By (10), (39) can be rewritten as

$$\dot{\Delta}_1 = \Delta_2 - k_1 |\Delta_1|^{1/2} \text{sign}(\Delta_1)$$

$$\dot{\Delta}_2 = F_t - k_2 \text{sign}(\Delta_1).$$  \hspace{1cm} (40)

All differential inclusions are understood in the Filippov sense [6], which means that the right hand side is enlarged in some points in order to satisfy the upper semi-continuity property. So, $
abla_2 \in [-f^+, f^+] - k_2 \text{sign}(\Delta_1)$, and $
abla_2 \in [-k_2 - f^+, k_2 + f^+]$ with $\Delta_1 = 0$. Using the trivial identity $(d|f|)/(dt) = \text{sign}(x)$, and computing the derivative of $\Delta_1$ with $\Delta_1 \neq 0$ obtain

$$\dot{\Delta}_1 \in [-f^+, f^+] - \left(\frac{1}{2} k_1 |\Delta_1|^{1/2} + k_2 \text{sign}(\Delta_1) \right).$$  \hspace{1cm} (41)

At the moments $\Delta_1 = 0$, taking into account that $\Delta_2 = F_t - k_2 \text{sign}(\Delta_1)$ and (25)

$$0 < k_2 - f^+ < |\Delta_2| < k_2 + f^+,$$  \hspace{1cm} (42)

Similar with our previous results in [7] and the form in [5], $\Delta_1(t_i) \geq (k_2 - f^+)_t$, $t_i$ is the time intervals between the successive intersection of the trajectory with the axis $\Delta_1 = 0$. Hence $t_i \leq (\Delta_1(t_i))/(k_2 - f^+)$, the total convergence time is estimated by $T \leq \sum (\Delta_1(t_i))/(k_2 - f^+)$. Therefore, $T$ is finite and the estimated states converge to the real states in finite time. Then the convergence of $(\dot{x}_1, \dot{x}_2)$ arrow $(x_1, x_2)$ in finite time is assured.

Remark 1: The two-stage neural observer (7) requires two design parameters: switch constant $\gamma$ and the upper bound of neural modeling error $f^+$ when start the sliding mode compensation. $\gamma$ decides when we stop neural networks learning and start sliding mode observer. How to choose this user-defined parameter is a trade-off. The bigger $\gamma$ is, the shorter training time the neural observer has. In this case, the neural modeling error is bigger, so $f^+$ should be bigger. If $\gamma$ is too small, the unmodeled dynamic prevent the condition $||e_t||_{Q_1} \leq \gamma$ is established, so the two-stage neural observer cannot enter sliding mode compensation, the finite time convergence cannot arrive.

Remark 2: Usually $f^+ > \gamma$, because $\gamma$ corresponds to the modeling error with the optimal weight, while $f^+$ corresponds to the modeling error when $||e_t||_{Q_1} < \gamma$. Since $f^+$ is unknown, how to choose $k_1$ and $k_2$ is also a trade-off problem. If we chose the sliding mode gains $k_1$ and $k_2$ very large to satisfy (25), the chattering becomes big. If we chose the sliding mode gains $k_1$ and $k_2$ smaller, (25) is not satisfied, then (41) and (42) cannot be established. Sliding mode observer will not converge, the observer error becomes bigger such that $||e_t||_{Q_1} > \gamma$. Now neural training is re-started again, until $||e_t||_{Q_1} < \gamma$ and enter sliding mode compensation again, see Fig. 1. But, if $k_1$ and $k_2$ are too small, (25) will never be right, the two-stage neural observer cannot converge in finite time. Another possible method is to used off-line training, such that $k_1$ and $k_2$ can be chosen small values.

V. SIMULATIONS

Consider a pendulum system with Coulomb friction and external, which was studied by [6]–[8]

$$\dot{x}_1 = x_2, y = x_1, x(0) = x_2(0) = 0$$

$$\dot{x}_2 = \frac{1}{J} \tau - \frac{g}{L} \sin x_1 - \frac{F_0}{J} \text{sign}(x_2) + v$$

where $M = 1.1, J = MT^2 = 0.891, g = 9.815, L = 0.9, V_0 = 0.18, F_0 = 0.45, v$ is an uncertain external pertur-
is the slowest and the observer error is big. As pointed out in Remarks 1 and 2, the design parameters $\eta$ and $k_2$ (or $k_3$) entail two trade-off problems. Fig. 3 shows how these parameter effect the two-stage neural observer. For the cases $s_1$ and $s_2$, $\eta = 2$, the sliding mode observer starts at $t = 2$, and the neural training re-starts. In the case $s_3$, $k_3 = k_2 = 6$, it finally enters the sliding mode at $t = 3$, while for $s_2$ $k_3 = k_2 = 3$, it finally enters the sliding mode until $t = 6$. Here for the cases $s_3$ and $s_4$, $\eta = 0.1$, they start sliding mode observer later. Also $k_3 = k_2 = 6$ for $s_3$, $k_3 = k_2 = 3$ for $s_4$.

VI. CONCLUSION

Although there exist some neural observers, sliding mode observers and neural sliding mode observers, two-stage neural sliding-mode observer is not applied in the literature. The stability and finite time convergence of the proposed observer are proven. Further works will be done on multilayer neural estimator and discrete time observer.

REFERENCES

of nonlinear systems using different hig-gain observer design,” Syst.
daplications of sliding mode observers,” in Variable Structure Sys-
tems: Towards XXIst Century Lecture Notes in Control and Informa-
253–282.
law for dynamic neural network observer,” IEEE Trans. Circuits Sys-
dynamic systems with application to a Coulomb friction oscillator,”
observer for mechanical systems,” IEEE Trans. Autom. Control,
work-based observer with application to flexible-joint manipulators,”
approximation science of sliding mode observers,” in Lecture Notes in
networks are universal approximators,” Neural Netw., vol. 2, pp.