Finite-time adaptive neural dynamic surface control for non-linear systems with unknown dead zone

Lian Chen | Qing Wang

School of Automation Science and Electrical Engineering, Beihang University, Beijing, China

Abstract
This paper proposes a new adaptive neural network finite-time dynamic surface control scheme for inaccurate non-linear systems subjected to unknown dead zones. The ‘explosion of differentiation’ is eliminated by the first-order filter in backstepping design. The parameter normalization scheme updates the coefficients of activation functions in the radial basis function neural networks. The dead zone inverse method estimates the dead zone parameters. It is proved that the proposed controller achieves the faster tracking performance and the boundedness of all signals in finite time. The contribution of this paper is that the presented controller not only has the advantages of transient performance and program execution time in comparison with traditional backstepping methods, but also its computational cost is lower than command filtered methods. Simulation experiment results are included to illustrate the effectiveness of the proposed scheme.

1 INTRODUCTION

At present, fuzzy logic systems (FLSs) [1, 2] and neural networks [3–5] are regarded as two kinds of intelligent approximators with different structures, because they exist excellent characteristics which can approximate the unknown non-linear smooth functions with great accuracy [6, 7]. For example, an intelligent fuzzy approximator was presented for a class of single-input-single-output (SISO) strict-feedback systems with a triangular structure to solve the singularity problem of the control boundary [1, 8]. Liu et al. [9] utilized universal approximators to approximate the differential functions and greatly simplified the derivation process of the control law for a class of non-affine systems. Radial basis function neural networks (RBFNNs) were employed to model the unavailable smooth functions in non-linear quantized systems [10]. In conclusion, the main role of an intelligent approximator in a control system is to approximate the unknown dynamic function [11] or estimate the uncertain parameters in real time [12]. This paper will employ RBFNNs to approximate uncertain or unavailable non-linear functions during the design procedure of controller.

Dead zones may be an unknown and unmeasurable physical phenomenon in practical industrial environments including hydraulic servo control [13], flexible joint industrial robots [14] and pneumatic servo driving systems [15]. Therefore, it is of great significance to improve the robustness and anti-interference ability of the control system when considering the influence of the dead zone in the controller design process. Shahriari and Rahmani [16] established a type-2 network system to identify the key parameters (the left and right breakpoint positions) of dead zone models. In addition to using neural networks to deal with dead zone phenomenon, there also exist other effective methods. Tao and Kokotovic [17] proposed a dead zone inverse to model the dead zone phenomenon for the first time. Then, the vibration problem of an actual control signal was tackled by introducing a deformed Sigmoid function to smooth the dead zone inverse [18]. Furthermore, time-varying integral functions [19], indirect adaptive updating parameters [20] and other schemes which are utilized to overcome the dead zones are not listed here. It can be concluded that combining different control algorithms with the dead zone model can improve the response performance of the control system and suppress the vibration of the tracking trajectory under the guidance of the asymptotic Lyapunov stability theory. On the basis of the proposed theories, above-mentioned research results [8, 13–20] cannot quantitatively estimate when the control system can enter the desired steady state [21].

In order to satisfy the requirements of real-time control, some specific industrial occasions must limit the total time of transient response [22]. Therefore, a finite-time stability theory...
which defines the specific solution formula for the time required to enter the steady state came into being [23]. Lee et al. [24] combined finite-time theory with a sliding mode technique to establish the precise control of piezoelectric actuators. In [25], state observers and FLSs have been utilized to design a finite-time controller for multiple-input-multiple-output (MIMO) systems as state variables of the control system are not measurable. The backstepping technology, as a popular control method, also has wide application [26–33]. For example, the backstepping technique was used to control linear systems or time-variant strict feedback systems. The shortcoming of the backstepping approach is ‘explosion of differentiation’ due to the repeated differentiations of immediate control functions, which has received considerable attention. To overcome the above drawback, Wang et al. [34] introduced a class of FLSs to approximate continuous functions which contain immediate control signals, so a finite-time adaptive fuzzy backstepping controller was designed for non-linear systems. Then, the backstepping controller was further successfully extended to non-affine systems with dead zones [35], which expands its application field. There also exist other methods to deal with the ‘explosion of differentiation’ in the process of backstepping design, for example, the first-order filters replace the virtual control signals; therefore the dynamic surface control (DSC) scheme appeared [36–43]. Ling et al. [44] normalized the adaptive parameters by using the maximum value of norm such that the running efficiency of the standard DSC algorithm is improved for flexible manipulators. Some command filtered control technology mainly utilizes Levant differentiators to approximate and compensate the intermediate control signals [45–50], which can also deal with the above ‘explosion of differentiation’ problem. Yu et al. [51] developed a finite-time command filtered method (the control scheme of [51] is abbreviated FTCF) to achieve the high-precision tracking of reference trajectory in finite time. In summary, finite-time controller design for different dynamic systems mainly combines finite-time stability theory with sliding mode, state observer or backstepping techniques. However, within our knowledge, there is not a finite-time DSC which has been applied to non-linear systems with unknown dead zones in the previous references.

Based on the improved DSC algorithm of [44], we turn to the finite-time tracking control issues for a class of inaccurate non-linear systems with unknown dead zones. First, the RBFNNs model the unknown functions of the dynamic system. Then the first-order filter is employed to overcome the ‘explosion of differentiation’. Next, the key parameters of the dead zone model are introduced into the existing control algorithm [44] such that a new finite-time DSC scheme that considers the dead zone is proposed. Compared with traditional backstepping methods, this proposed control scheme has the advantages of lower computational burden and faster transient response.

1. The first derivative of a reference signal is only needed in the process of control signal design; therefore, the described algorithm has obvious strengths when the reference signal exists no high-order derivative.

2. The first-order filter accurately approximates the derivative of the intermediate virtual control signal so that the ‘differential explosion’ problem in backstepping design is solved.

3. The first-order filter accurately approximates the derivative of the intermediate virtual control signal so that the ‘differential explosion’ problem in backstepping design is solved.

4. In the control design process, since the first-order filter and the adaptive parameter normalization scheme are introduced such that the number of differential equations greatly reduced. Thus compared with FTCF [51], the proposed algorithm significantly degrades the computational burden and accelerates the efficiency of online tuning parameters.

Note that the organization of this paper is constructed as follows. Section 2 represents the problem formulation and preliminaries. The detailed design and stability proof of the finite-time dynamic surface control scheme (the proposed control scheme is abbreviated as FTDSC) considering dead zones are given in Section 3. Simulation comparison examples are presented to verify the effectiveness of the FTDSC in Section 4. Finally, the conclusions are given in Section 5.

2 | PROBLEM STATEMENT AND PRELIMINARIES

2.1 | Description of the system with unknown dead zone

In the subsequent derivation, let estimation error \( \hat{\xi} = (\xi) - \xi \), where \( \hat{\xi} \) is the estimation value, \( \xi \) denotes the true value.

Throughout the brief, \( \| \xi \| = \sqrt{\sum_{i=1}^{n} \xi_i^2} \), where \( \xi=[\xi_1, ..., \xi_n]^T \), \( \mathbb{R}^+ \) is a positive real number, \( \mathbb{R}^n \) is the real space and its superscript \( n \) represents dimension.

Consider a class of \( n \) order non-linear systems whose general model is given as following form:

\[
\begin{align*}
  \dot{x}_i &= F_i(x) + P_i(x)u_T, & 1 \leq i \leq n-1 \\
  \dot{x}_n &= F_n(x) + P_n(x)u_T \\
  y &= x_1,
\end{align*}
\]

(1)

where \( x = [x_1, ..., x_n]^T \in \mathbb{R}^n \) is the state vector, \( y \) denotes the measured output signal, \( u_T \) is the actual control signal (the output of dead zone). Due to the inaccuracy of physical system models, \( F_i(x) \) and \( F_n(x) \) are difficult to obtain, so we only assume that they are bounded unknown continuous differentiable functions. Notice that \( P_i(x) \), \( P_n(x) \) are continuously bounded and often used in strict-feedback systems [51], for example.
The dead zone model is described by the following equation [18]:

\[
u_T = D(u_C) = \begin{cases} d_r(u_C - d_+) & u_C \geq d_+ \\ 0 & d_- < u_C < d_+ \\ d_l(u_C - d_-) & u_C \leq d_-,
\end{cases}
\tag{2}
\]

where \( u_C \) is plant signal (the input of dead zone), and \( D(u_C) \) is the dead zone function. \( d_r > 0, d_l > 0, d_- < 0, d_+ > 0 \) denote right slope, left slope, right breakpoint and left breakpoint of the dead zone, respectively, which are assumed unknown.

Symbol \( D^{-1}(u_C) \) represents the inverse function of \( D(u_C) \); then one will obtain

\[
u_C = D^{-1}(u_T(t)) = \frac{u_T(t) + d_+ d_-}{d_r} S_r(t) + \frac{u_T(t) + d_+ d_-}{d_l} S_l(t)
\tag{3}
\]

with

\[
S = \begin{bmatrix} S_r(t)u_C & -S_r(t) & S_l(t)u_C & -S_l(t) \end{bmatrix}^T,
\tag{4}
\]

\[
S_r(t) = \begin{cases} 1 & u_T \geq 0 \\ 0 & u_T < 0 \end{cases},
\tag{5}
\]

\[
S_l(t) = \begin{cases} 1 & u_T \leq 0 \\ 0 & u_T > 0 \end{cases}
\]

where \( S_r(t) \) and \( S_l(t) \) are membership functions of the true dead zone model.

Now, the parameter vector set \( \theta = [\theta_{d1} \ \theta_{d2} \ \theta_{d3} \ \theta_{d4}]^T \) is defined to estimate the key parameters of the dead zone model, where \( \theta_{d1} = d_r, \theta_{d2} = d_+, \theta_{d3} = d_l, \theta_{d4} = d_- \). Then (2) can be redefined as

\[
u_T(t) = \theta_d^T \hat{\mathbf{S}}.
\tag{6}
\]

Based on the definitions of (6) and (7), (3) can be rewritten as

\[
u_C = \frac{\hat{\nu}_T(t) + \theta_{d1} S_r(t) + \theta_{d2} S_l(t)}{\theta_{d3}}
\tag{7}
\]

\[
u_C = \frac{\hat{\nu}_T(t) + \theta_{d1} S_r(t) + \theta_{d2} S_l(t)}{\theta_{d3}}
\tag{8}
\]

According to the designed \( \hat{\nu}_T(t) \) and \( \theta_{d}^T \), \( u_C \) can be obtained.

Now, subtract (6) from (5) to get the estimation error \( \tilde{u}_T(t) = \theta_d^T \hat{\mathbf{S}} - \theta_d^T \hat{\mathbf{S}} \), and then \( \tilde{u}_T(t) \) can be further described as

\[
\tilde{u}_T(t) = \theta_d^T (\mathbf{S} + \hat{\mathbf{S}}) - \theta_d^T \hat{\mathbf{S}} = \theta_d^T \mathbf{S} + \theta_d^T \hat{\mathbf{S}}.
\tag{9}
\]

Then, by means of deforming (9), \( u_T(t) \) can be expressed as

\[
u_T(t) = \hat{\nu}_T(t) + \tilde{u}_T(t) = \nu_T(t) + \theta_d^T \mathbf{S} + \theta_d^T \hat{\mathbf{S}}.
\tag{10}
\]

Substituting (10) into (1), the \( n \) order system is further described as

\[
\begin{cases}
\dot{S}_i = F_i(x_i) + P_i(x_i) \zeta_i(n+1) \\
\dot{S}_i = F_i(x_i) + P_i(x_i) (\hat{\nu}_T(t) + \theta_d^T \hat{\mathbf{S}}) + \theta_d^T \hat{\mathbf{S}}
\end{cases},
\tag{11}
\]

where \( i = 1, 2, \ldots, n \).

Simplify \( F_i(x_i), P_i(x_i) \) for a dynamic system, whose time derivative can be tracked the reference signal \( y \) in a finite period of time.

\subsection{2.2 Lemmas and assumptions}

To cope with the design and stability analysis of controller, the following Lemmas and Assumptions should be included.

\begin{lemma}
Assume that there exists a smooth positive definite quadratic function \( L(x) \) for a dynamic system, whose time derivative can be expressed as

\[
\dot{L}(x) = -\rho_1 L(x) + c_0, \quad t \geq 0.
\tag{12}
\]

If three constants satisfy \( \rho_0 > 0, 0 < a < 1, c_0 > 0 \), then the system is semi-global practical finite-time stable (SGPFS), and the finite time \( T_R \) of which the output signal enters the steady state can be described as

\[
T_R = \frac{L^1 - \epsilon(0) - \left( \frac{c_0}{(1 - a)\rho_1 \rho_0} \right) (t, \epsilon)}{(1 - a)\rho_1 \rho_0}
\tag{13}
\]

with \( \rho_1 \in (0, 1) \).
\end{lemma}
CONTROLLER DESIGN AND ANALYSIS

In this section, the design process of the FTDSC controller for the $n$ order non-linear system (1) with unknown dead zone are divided into three steps. The proof of SGFPS for this system is given after the above steps are completed.

Define a general formula for the first-order filter:

$$
\dot{Q}_j, \tau_j = \alpha_{j-1}, j = 2, ..., n,
$$

where $Q_j, \tau_j$ denote the output of first-order filter and the filter constant, respectively. $\alpha_{j-1}$ represents the intermediate control signal that needs to be designed. Then, the transformations of coordinates can be described as follows:

$$
e_j = x_j - y_j, e_j = x_j - Q_j,
$$

where $e_j$ are the errors of state variable, and $e_j$ denotes the filter error.

Remark 1. The derivative of intermediate control signal $\alpha_{j-1}$ is approximated by the output $Q_j, j = 2, 3, ..., n$ of the first-order filter to deal with the ‘explosion of differentiation’ problem in the backstepping design. In contrast, above signals are modelled by Levant differentiators in [51]. Compared with the FTCF, the FTDSC diminishes significantly the number of differential equations that need to be solved; therefore it has the advantage of low time complexity.

To ensure that the closed-loop control system converges in finite time, we introduce the following control signal $\alpha_i$ for $i = 1, 2, ..., n$ and adaptive functions:

$$
\alpha_i = \frac{-\lambda_i \gamma^{d-2}}{P_i} - \frac{e_i}{2P_i} E_i(Z_i)^T E_i(Z_i) - \frac{\epsilon_i}{2P_i},
$$

$$
\dot{\theta} = \rho_1 \sum_{i=1}^n \frac{\dot{E}_i(Z_i)^T E_i(Z_i)}{2y_i^2} - \phi \dot{\theta},
$$

where adjustable parameters $\lambda_i \in \mathbb{R}^+, \gamma_i \in \mathbb{R}^+, \rho_i \in \mathbb{R}^+, \phi \in \mathbb{R}^+, \dot{\theta} \in \mathbb{R}^+$, and the input vectors of RBFNNSs are $Z_1 = [x_1, Q_2, y_{r1}], Z_k = [x_1, ..., x_k, Q_{k+1}, \dot{\theta}]^T, k = 2, 3, ..., n - 1, Z_n = [x_1, ..., x_n, Q_{n+1}, \dot{\theta}]^T$. It can be inferred from (21) that $\epsilon_i^{2^{d-1}}$ will be infinite as $2^{d-1} < 1$, in order to avoid singularity [33], we assume that $\beta \in (0, 1)$. Notice that sometimes let $E_i(Z_i) = E_i$, for the simplicity of presentation.

Remark 2. On the one hand, the robust compensator $\dot{\theta}_d$ is employed to deal with the dead zone, which improves the
disturbance-rejection ability of controller, on the other hand, the control signal $\alpha_i$ whose design procedure is based on the framework of the SGPFS, therefore the transient performance of the FTDS is better than traditional backstepping schemes.

**Step 1** ($i = 1$): Taking the derivative of $\xi_1 = x_1 - y_r$, and according to the transformation of coordinates (20), one has

$$\dot{\xi}_1 = F_1 + P_1(\xi_2 + \xi_2 + \alpha_1) - \dot{y}_r. \quad (23)$$

Substituting (21) into (23), and multiplying by $\xi_1$, one can obtain

$$\xi_1 \dot{\xi}_1 = -\lambda_1 \xi_1^2 - \frac{\xi_1^2}{2} + e_1 \dot{P}_1 \xi_2$$

$$+ e_1(P_1 \xi_2 + F_1 - \dot{y}_r). \quad (24)$$

The input function of the first RBFNN is selected as

$$\vec{F}_1(\xi_1) = P_1 \xi_2 + F_1 - \dot{y}_r. \quad (25)$$

Note that the input functions of the RBFNN of $i$th and $n$th subsystem are $\vec{F}_i(\xi_1) = F_i + P_i \xi_2 + e_{i-1} P_{i-1} - \dot{Q}_i$ and $\vec{F}_n(\xi_2) = P_n \theta^T_S + F_n + e_{n-1} P_{n-1} - \dot{Q}_n$, respectively. Substituting (25) into (24) produces

$$\xi_1 \dot{\xi}_1 = \xi_1 \vec{F}_1(\xi_1) - \lambda_1 \xi_1 \xi_1^2 - \frac{\xi_1^2}{2} + e_1 \dot{P}_1 \xi_2. \quad (26)$$

It can be inferred from (17) that there is an infinitesimal constant $\delta_1$ that satisfies the following equilibrium:

$$\vec{F}_1(\xi_1) = W_1^T E_1 + \delta_1(\xi_1), \quad |\delta_1(\xi_1)| \leq \delta_1, \quad (27)$$

where $\delta_1(\xi_1)$ which is simplified as $\delta_1$ represents the approximation error of first RBFNN. Now, substituting (27) into (26), $\xi_1 \dot{\xi}_1$ can be expressed as

$$\xi_1 \dot{\xi}_1 = \xi_1 W_1^T E_1 + e_1 \delta_1 - \lambda_1 \xi_1 \xi_1^2 - \frac{\xi_1^2}{2} + e_1 \dot{P}_1 \xi_2. \quad (28)$$

To facilitate controller design, (28) is introduced a tuning parameter $\gamma_1$, so $\xi_1 \dot{\xi}_1$ can be rewritten as

$$\xi_1 \dot{\xi}_1 = \frac{\xi_1 W_1^T E_1}{\gamma_1} + e_1 \delta_1 - \lambda_1 \xi_1 \xi_1^2 - \frac{\xi_1^2}{2} + e_1 \dot{P}_1 \xi_2. \quad (29)$$

Based on the Young's inequality and (18), the following inequalities hold:

$$\frac{e_1 W_1^T E_1}{\gamma_1} + e_1 \delta_1 \leq \frac{e_1^2}{2} + \frac{\delta_1^2}{2}. \quad (30)$$

Substituting (30) into (29), one can get

$$\dot{\xi}_1 \xi_1 \leq \frac{e_1^2}{2} + \frac{\delta_1^2}{2} + e_1 \dot{P}_1 \xi_2. \quad (31)$$

According to (19) and (20), we have

$$\dot{\xi}_1 \xi_1 = \dot{Q}_2 - \dot{\xi}_1 = \frac{e_1^2}{2} + \frac{\delta_1^2}{2} + e_1 \dot{P}_1 \xi_2. \quad (32)$$

The designed intermediate control signal $\alpha_1$ is a bounded function; thus $-\dot{\alpha}_1$ is also a bounded function, namely, there is a positive number $\theta_1$ which satisfies the inequality $| - \dot{\alpha}_1 | \leq \theta_1$. Substituting this inequality into (32); then multiplying both sides by $\xi_1$, and according to the Young's inequality, the following inequality holds:

$$\xi_1 \dot{\xi}_1 \xi_1 \leq \frac{e_1^2}{2} + \frac{\delta_1^2}{2} + e_1 \dot{P}_1 \xi_2. \quad (33)$$

Choose a positive definite Lyapunov function $L_1 = \frac{1}{2}(\xi_1^2 + \xi_1^2)$, and its derivative follows:

$$L_1 = \xi_1 \dot{\xi}_1 + \xi_1 \dot{\xi}_1. \quad (34)$$

Substituting (31) and (33) into (34) results in

$$L_1 \leq \frac{e_1^2}{2} + \frac{\delta_1^2}{2} + e_1 \dot{P}_1 \xi_2. \quad (35)$$

**Step 2** ($i = 2, \ldots, n - 1$): With the help of (20) and (21), the tracking error of the $i$th order subsystem can be described as

$$\dot{\xi}_i \xi_i = -\lambda_i \xi_i \xi_i^2 - \frac{e_i^2}{2} - e_i \xi_1 \xi_{i+1} - \frac{\xi_i^2}{2} + e_i \dot{P}_1 \xi_2$$

$$+ e_i (P_i \xi_{i+1} + F_i - \dot{Q}_i). \quad (36)$$

Similar to the derivation process of (27)–(35), we have

$$L_i \leq \frac{e_i^2}{2} + \frac{\delta_i^2}{2} + \left(1 - \frac{1}{r_{i+1}}\right) e_i^2$$

$$- \lambda_i \xi_i \xi_i^2 + e_i \xi_1 \xi_{i+1} + \frac{\theta_i^2}{4} - e_{i-1} \dot{P}_1 \xi_i. \quad (37)$$
Step 3 \((i = n)\): Taking the time derivative of \(e_n = x_n - \hat{x}_n\); then substituting \(\dot{x}_n\) of (11) into above equation, we get
\[
\dot{e}_n = \dot{x}_n - \dot{\hat{x}}_n = F_n + P_n \left(\mathbf{v}_n(t) + \mathbf{b}_n^T \dot{\mathbf{S}} + \mathbf{e}_n^T \mathbf{R}_n^T \mathbf{S}\right) - \dot{Q}_n.
\]
(38)

The \(\mathbf{a}_n\) is viewed as \(\dot{\mathbf{v}}_n(t)\), then we can obtain the error dynamic of the \(n\)-th order subsystem:
\[
e_n^T \mathbf{a}_n^T - \frac{c_e^2}{2} \frac{E_n^T E_n}{2y_n^2} + \frac{c_n}{2} + e_n P_n \mathbf{b}_n^T \dot{\mathbf{S}}
+ e_n P_n \mathbf{b}_n^T \dot{\mathbf{S}} - F_n - \bar{Q}_n.
\]
(39)

Similar to the derivation procedure from (27) to (31), one has
\[
e_n^T \mathbf{a}_n^T - \frac{c_e^2}{2} \frac{E_n^T E_n}{2y_n^2} + \frac{c_n}{2} + \frac{d_n}{2} e_n P_n \mathbf{b}_n^T \dot{\mathbf{S}} - e_{n-1} P_{n-1} e_n.
\]
(40)

Introduce a smooth Lyapunov function as follows:
\[
L_n = \frac{1}{2} \left(\dot{e}_n^T \mathbf{a}_n^T - \frac{d_n}{2} e_n P_n \mathbf{b}_n^T \dot{\mathbf{S}} - e_{n-1} P_{n-1} e_n\right),
\]
(41)

where \(\mathbf{A} \) is a \(4 \times 4\) positive definite diagonal matrix, \(\mathbf{A}^{-1}\) represents the inverse of \(\mathbf{A}\). The derivative of (41) is
\[
\dot{L}_n = e_n \dot{e}_n + \frac{\partial \mathbf{S}}{\partial \mathbf{a}} \dot{\mathbf{a}} + \frac{\partial \mathbf{S}}{\partial \mathbf{b}} \dot{\mathbf{b}} - \frac{c_n}{2} e_n P_n \mathbf{b}_n^T \dot{\mathbf{S}},
\]
(42)

where \(\dot{\mathbf{S}} = \mathbf{S} - \mathbf{a} \dot{\mathbf{a}}\). It can be inferred from (18) that \(\mathbf{S}\) is a constant value and its derivative is 0; so the following equation holds:
\[
\dot{\mathbf{S}} = \mathbf{S} - \mathbf{a} \dot{\mathbf{a}} = -\dot{\mathbf{a}} \dot{\mathbf{a}}.
\]
(43)

Similarly, \(\dot{\mathbf{b}} = \dot{\mathbf{b}}\), \(\dot{\mathbf{b}}\), \(\dot{\mathbf{b}}\), \(\dot{\mathbf{b}}\) is a key constant parameter vector of the dead zone model, and its derivative is also 0; thus
\[
\dot{\mathbf{b}} = \dot{\mathbf{b}} = \dot{\mathbf{b}} = \dot{\mathbf{b}} = -\dot{\mathbf{b}} \dot{\mathbf{b}}.
\]
(44)

Substituting (40), (43) and (44) into (42), one can get
\[
L_n \leq -\frac{\lambda_n^2}{2} \dot{e}_n^T \mathbf{a}_n^T - \frac{c_e^2}{2} \frac{E_n^T E_n}{2y_n^2} + \frac{c_n}{2} + e_n P_n \mathbf{b}_n^T \dot{\mathbf{S}} - e_{n-1} P_{n-1} e_n - \frac{\partial \mathbf{S}}{\partial \mathbf{a}} \dot{\mathbf{a}} - \frac{\partial \mathbf{S}}{\partial \mathbf{b}} \dot{\mathbf{b}}.
\]
(45)

Now, define the final smooth quadratic Lyapunov function for entire non-linear control system as follows:
\[
L_{\text{all}} = \sum_{i=1}^{n} L_i = \sum_{i=1}^{n} \frac{c_i^2}{2} + \sum_{j=2}^{n} \frac{c_j^2}{2} + \frac{\mathbf{b}_n^T \mathbf{a}_n^T \mathbf{L}_n^{-1} \tilde{\mathbf{b}}_n}{\mathbf{a}_n^T}.
\]
(46)

The derivative of \(L_{\text{all}}\) with respect to time can be described as
\[
\dot{L}_{\text{all}} = \sum_{i=1}^{n} \dot{L}_i = \sum_{i=1}^{n} \dot{e}_i^2 + \sum_{j=2}^{n} \dot{e}_j = -\frac{\partial \mathbf{S}}{\partial \mathbf{a}} \dot{\mathbf{a}} - \frac{\partial \mathbf{S}}{\partial \mathbf{b}} \dot{\mathbf{b}}.
\]
(47)

The summation of (45), (37) and (35) into (47) produces
\[
\dot{L}_{\text{all}} \leq \sum_{i=1}^{n} \dot{e}_i^2 + \sum_{j=2}^{n} \dot{e}_j^2 + \frac{\Theta^2}{2} + \frac{\Theta^2}{4}.
\]
(48)

Substituting (22) into (48) results in
\[
\dot{L}_{\text{all}} \leq \sum_{i=1}^{n} \dot{e}_i^2 + \sum_{j=2}^{n} \dot{e}_j^2 + \frac{\Theta^2}{2} + \frac{\Theta^2}{4}.
\]
(49)

Remark 3. We choose only one adaptive law to update the coefficients of activation functions in RBFNNs. Compared with the traditional backstepping algorithm [32], the computational burden of FTDS is less than the backstepping scheme, based on the parameter normalization algorithm (18).

It can be inferred from Lemma 3 that the following inequalities are satisfied (the detailed proof is shown in the Appendix.):
\[
\frac{\dot{\mathbf{S}}}{\partial \mathbf{a}} \dot{\mathbf{a}} \leq (1 - \beta) q_u + \frac{\dot{\mathbf{S}}}{\partial \mathbf{a}} \dot{\mathbf{a}} \leq \frac{(\mathbf{S}^T \dot{\mathbf{a}})^2}{2\mathbf{a}}.
\]
(50)
is a bounded signal. Because 

\[ f\in \mathbb{R}_{+} \]

and dead zone compensator following inequalities can be obtained:

\[
\begin{align*}
\alpha &\leq \min(1) \\
\nu &\in [0, \infty) \quad \text{with} \quad \alpha < 1, \quad \nu(x) = \min\{2\alpha_1, \ldots, 2\alpha_d\}. \quad \text{According to the Lemma 2, the following inequalities can be obtained:}
\end{align*}
\]

Let \( \min(\frac{1}{r_j} - 1) \geq \frac{\phi_{0}}{2}, j = 2, 3, \ldots, n \) and \( \rho_{0} = \min\{\phi_{0}, \phi_{d}\} \), where \( \phi = \min\{2\alpha_1, \ldots, 2\alpha_d\} \). According to the Lemma 2, the following inequalities can be obtained:

\[
L_{all} \leq -\rho_{0}L_{all}^\beta + \epsilon_0 
\]

with

\[
\epsilon_0 = 2(1 - \beta)q_{\rho} + \frac{\phi_{0}^2}{2\rho_1} + \frac{\phi_{d}^2}{2\rho_2} + \sum_{j=2}^{n} \frac{\theta_{j}^2}{2} + \sum_{j=2}^{n} \frac{\theta_{j}^2}{2}. \quad (53)
\]

According to above-mentioned derivation process, \( 0 < \beta < 1, \epsilon_0 > 0, \rho_{0} \geq 0 \), and \( L_{all} \) is a positive definite function, which satisfies the four conditions of the proposed Lemma 1, in addition, if the \( (1) \) also satisfies Assumptions 1–3; then all signals will be SGPFs, i.e. \( \epsilon_1, \epsilon_2, \ldots, \epsilon_{n}, \tau_{2}, \tau_{1}, \ldots, \tau_{n}, \theta_{j} \) and \( \theta_{d} \) are semi-global practical finite-time bounded. Furthermore, we can define the finite time \( T_R \) that the control system enters the steady state

\[
T_R = \frac{\left( L_{all}^{1-\beta}(\epsilon_{0}^{2}) \right)^{1/(1-\beta)}}{(1-\beta)\rho_{0}^\beta}. \quad (54)
\]

Furthermore, the tracking error of the dynamic system satisfies

\[
\epsilon_1 = |y - y_r| \leq 2\left( \frac{\epsilon_0}{(1 - \omega_0)\rho_0^\beta} \right) \quad (55)
\]

with \( \omega \in (0, 1) \). For \( \forall \omega \geq T_R \), under the virtual control signal \( \alpha_1, \alpha_2, \ldots, \alpha_{n-1} \), actual control signal \( \alpha_n \), adaptive updating law \( \dot{\theta} \) and dead zone compensator \( \dot{\theta}_{d} \), and then the tracking error \( \epsilon_1 \) approaches an infinitesimal domain of the origin. It can be inferred from the Assumption 1 that the reference trajectory \( y_r \) is a bounded signal. Because \( \epsilon_1 = y - y_r \), the boundedness of state \( x_1 \) is proven. The dynamic model \( (1) \) implies that \( R_1(x_1) \) is a bounded continuous function. Since \( \alpha_1 \) is a function about \( R_1(x_1) \) and \( \epsilon_1 \), the \( \alpha_1 \) is a bounded virtual control law. It can be observed from \( \epsilon_2 = Q_2 - \alpha_1 \) that \( Q_2 \) is also a bounded signal. Because \( \epsilon_2 = x_2 - Q_2 \), the boundedness of state \( x_2 \) is proven.

Following the similar analysis process, the boundedness of all the signals in the closed-loop systems will be proven.

## 4 Simulation Examples

In order to show that the described algorithm can overcome the dead zone phenomenon and exhibit excellent tracking performances, we assume that the plant signal \( u_c \) passes the following dead zone model:

\[
u_T = D(u_c) = \begin{cases} 
(u_c - 0.5) & u_c \geq 0.5 \\
0 & -0.5 < u_c < 0.5 \\
(u_c + 0.5) & u_c \leq -0.5.
\end{cases} \quad (56)
\]

To reduce the number of parameters that needs to be debugged in the simulation [18], the breakpoint parameters \( \theta_{d1} \) and \( \theta_{d4} \) of the dead zone model are only estimated. Furthermore, other constant parameters of dead zone are chosen as: \( \theta_{d1} = \theta_{d3} = 1, A = \text{diag}(12, 12), \rho = 20, \phi = 0.5 \).

Simulation 1. Consider a simulation example of a third-order electromechanical system as follows:

\[
\begin{align*}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= (x_3 - 0.0181x_2 - 2.2816 \sin(x_1))/0.0642 \\
\dot{x}_3 &= (u_T - 0.9x_2 - 5x_3)/15 \\
y &= x_1.
\end{align*} \quad (57)
\]

Comparing the state equations of the electromechanical system with the general formula of non-linear system (1), we can conclude: \( P_1 = 1, P_2 = 1/15, P_2 = 1/0.0642, F_1 = 0, F_2 = -0.0181x_2 - 2.2816 \sin(x_1), F_3 = -0.9x_2 - 5x_3/15, F_4 = 0.0642, F_5 = 0.0642, F_6 = 0.0642, F_7 = 0.0642, F_8 = 0.0642, F_9 = 0.0642 \).

The correlation coefficient of each variable in (57) refers to the simulation example given in [57], but the difference is that we will use RBFNNs to tackle with the \( E_j(x_k), i = 1, 2, 3 \). The initial conditions are defined as \( x_1(0) = 0.5, x_2(0) = 0.5, x_3(0) = 0.5, \dot{\theta}_0(0) = 0.1, \dot{Q}_2(0) = x_1(0), \dot{Q}_2(0) = x_2(0), \dot{\theta}_d(0) = 0.5, \dot{\theta}_2(0) = 0.5, \dot{\theta}_d(4)(0) = -0.5. \) A sinusoidal-like function can be selected as the reference signal (its curve is shown in Figure 1): \( y_r = 0.5 \sin(\tau) + 0.5 \sin(5\tau) \). The design parameters are chosen as \( \beta = 0.975, \lambda_1 = 22, \lambda_2 = 22, \lambda_3 = 120, \gamma_1 = 15, \gamma_2 = 15, \gamma_3 = 15, \phi = 0.5, \tau_2 = 0.001, \tau_3 = 0.004, \rho_1 = 20 \).

Taking approximation accuracy and program running efficiency into account, the centre position vector of the Gaussian function is designed as \( \epsilon = [-3.5, -2.5, -1.5, -0.5, 0.5, 1.5, 2.5, 3.5, 4.5] \), and the Gaussian width \( \beta_m = 3, m = 1, 2, \ldots, 9 \) for every RBFNN [35]. Notice that above RBFNNs will be used in subsequent backstepping scheme.

Remark 4. It can be known from the definitions of (54) and (55) that there exists a serious coupling relationship between adjustable parameters \( (\beta, \lambda, \gamma) \) and integrated system performances \( (T_R \) and \( e_1) \) such that the specific values of above
parameters can be comprehensively weighed by simulations. On the one hand, the tracking error $e_1$ is mainly impacted by $\lambda_i$, $i = 1, 2, ..., n$, therefore these design parameters can be determined first. On the other hand, the design parameter $\beta$ affects the time $T_p$ for the controller to enter a steady state. Specifically, when a larger $\beta$ is selected, the controller can enter a steady state more quickly. Therefore, on the basis of ensuring the convergence of the tracking error, we can fine-tune $\beta$ to make the controller exhibit more excellent transient performance.

Simulation 2. The traditional backstepping algorithm [32] is applied to the same actual electromechanical system (57), we can obtain the following control laws and adaptive parameter update functions (in order to save space, we have omitted the specific design process of backstepping controller here.):

$$\alpha_i = \frac{-\lambda_i \gamma_i}{\rho_i} \frac{\partial^2 E_i^T(Z_i)E_i(Z_i)}{2\lambda_i^2} - \phi_i \hat{\gamma}_i \hat{\theta}_i = \Lambda(\rho_i \lambda_i \gamma_i - \phi_i \hat{\gamma}_i \hat{\theta}_i).$$

According to backstepping algorithm, the input vectors of all RBFNNs can be, respectively, defined as $Z_1 = [x_1, y_1, \dot{y}_1]^T$, $Z_2 = [x_1, x_2, \hat{x}_1, \dot{y}_1, \dot{y}_2]^T$, $Z_3 = [x_1, x_2, x_3, \hat{x}_1, \dot{x}_2, y_2, \dot{y}_1, \dot{y}_2]^T$, it contains the derivative information of the reference signal, and its derivative order is the same as the system order. It is worth mentioning that the proposed FTDSC only needs the first derivative of the reference signal, which may be more suitable in real applications.

The control parameters are chosen as: $\lambda_1 = 20, \lambda_2 = \lambda_3 = 35, \gamma_1 = \gamma_2 = \gamma_3 = 15, \rho_1 = \rho_2 = \rho_3 = 20, \phi_1 = \phi_2 = \phi_3 = 0.2$. The initial values of adaptive parameters are chosen as: $\hat{\gamma}_1(0) = \hat{\gamma}_2(0) = \hat{\gamma}_3(0) = 0.1$, and other initial conditions are the same as Simulation 1.

Remark 5. It should be pointed out that the backstepping approach requires three adaptive parameters. However, the presented controller needs only an adaptive parameter, which has the advantage of the lower computational burden. The above backstepping approach does not use finite-time theory such that its response speed is slower than FTDSC, which can be verified by Figure 2.

Figure 1 shows that both the backstepping approach and FTDSC have fascinating tracking performance. The tracking error curves of the two algorithms are displayed in Figure 2. The tracking error of FTDSC has stopped oscillating at time $t = 0.5086$, namely, has entered a steady state, while the tracking error of the backstepping approach is still unstable at this moment, and the tracking error stop oscillating as $t = 1.06$. If above time which the tracking error stop oscillating is regarded as a transient performance index, then, the transient tracking performance of developed controller increase 200% in comparison with the traditional backstepping approach.

Figures 3–6 show the time histories of other state variables, adaptive laws, plant signal and actual control signal. It can be inferred that all curves are bounded and approach a neighbourhood of the origin in finite time, which further validates the correctness of the stability analysis process in Section 3. Figure 5 implies that the output amplitude of the proposed controller at the initial moment is reduced by 3.21% in comparison with the control output of backstepping method, which is more suitable in industrial application. It is worth pointing out that the dead
zone results in the roughness (inflection point) of the curve, and its detailed description is shown in Figure 6.

Based on Remark 1, the first-order filter in FTDSC only needs one differential equation to describe, while a Levant differentiator in FTCF needs two differential equations to describe; therefore, we can further infer that for the $n$-order systems the number of differential equations involved in FTDSC is reduced by $(n - 1)/2$, compared with FTCF. It is worth noting that in order to save space, the design process of the FTCF is detailed in [51] and will not be repeated here. To ensure the single variable principle, the parameters of FTCF are debugged, and the FTCF has the same tracking performance as the FTDSC. Under the above conditions, a simulation of time complexity is performed to prove the correctness of the conclusion mentioned in Remark 1.

Simulation 3. Set the same total sampling time in the MATLAB2016a software environment. Then run programs corresponding to two different algorithms and record the average running time. Next replace the total sampling time and repeat the above experimental steps. Finally, the simulation results are shown in Table 1.

Hardware configuration of the computer on which MATLAB2016a depends: operating system: Windows 8.1; CPU: Intel Core i5-6500@3.2 GHz; RAM: 8 GB. To avoid relative errors, the data in Table 1 are obtained by recording the results of five replicate experiments and calculating their average values under the same simulation environment.

Remark 6. It can be inferred from the data in Table 1 that the average solution time of FTDSC is reduced by 47% with respect to FTCF, and above-mentioned strengths are more obvious, as the total sampling time increase.

### Table 1  Solving time consumption comparison

| Total sampling time | FTDSC | FTCF |
|---------------------|-------|------|
| 5                   | 1.05  | 1.37 |
| 10                  | 1.67  | 2.61 |
| 15                  | 2.38  | 3.71 |
| 20                  | 3.16  | 4.87 |
| 25                  | 3.89  | 6.38 |
| 30                  | 4.64  | 7.18 |

5 | CONCLUSION

This paper proposes a new adaptive neural network finite-time DSC approach to solve the tracking problem for uncertain nonlinear systems with unknown dead zones. The first-order filters are employed to approximate the derivatives of virtual control signals, the RBFNNs model the uncertain system functions, and the unknown dead zone is replaced by the adaptive dead zone inverse function. Simulations 1 and 2 imply that the transient tracking performance of the developed controller increases 200% in comparison with the traditional backstepping approach. The results of Simulation 3 show that the average solution time of FTDSC reduces 47%, compared with FTCF. It is worth noting that the proposed FTDSC needs to utilize some knowledge of dynamics systems; therefore future work will mainly account for how to extend FTDSC to a class of totally unknown non-affine non-linear systems.
ACKNOWLEDGEMENT
This work is supported by the National Natural Science Foundation of China (Grant nos. 61622308, 61933010, 61833016 and 61873295).

ORCID
Lian Chen https://orcid.org/0000-0003-3569-2222

REFERENCES
1. Peng, J.Z., Dubay, R.: Adaptive fuzzy backstepping control for a class of uncertain nonlinear strict-feedback systems based on dynamic surface control approach. Expert Syst. Appl. 120(1), 239–252 (2019)
2. Chen, L., Wang, Q.: Finite-time adaptive fuzzy command filtered control for nonlinear systems with indifferentiable non-affine functions. Nonlinear Dyn. 100, 493–507 (2020)
3. Wang, D., Huang, J.: Neural network-based adaptive dynamic surface control for a class of uncertain nonlinear systems in strict-feedback form. IEEE Trans. Neural Networks 16(1), 195–202 (2005)
4. Liu, B., et al.: Adaptive dynamic wavelength and bandwidth allocation algorithm based on error-back-propagation neural network prediction. Opt. Commun. 457(1), 276–284 (2019)
5. Zhao, H., Liu, D.R.: Event-triggered decentralized tracking control of modular reconfigurable robots through adaptive dynamic programming. IEEE Trans. Ind. Electron. 67(4), 3054–3064 (2020)
6. Zhao, B., et al.: Reinforcement learning-based optimal stabilization for unknown nonlinear systems subject to inputs with uncertain constraints. IEEE Trans. Neural Networks Learn. Syst. 30(10), 4330–4340 (2019)
7. Yang, Y.L., et al.: Hamiltonian-driven adaptive dynamic programming for continuous nonlinear dynamical systems. IEEE Trans. Neural Networks Learn. Syst. 28(8), 1929–1940 (2017)
8. Zhu, G., et al.: Command filtered robust adaptive NN control for a class of uncertain strict-feedback nonlinear systems under input saturation. J. Franklin Inst. 355(15), 7548–7569 (2018)
9. Liu, Z., et al.: Adaptive fuzzy control for pure-feedback nonlinear systems with nonaffine functions being semibounded and indifferentiable. IEEE Trans. Fuzzy Syst. 26(2), 395–408 (2018)
10. Wang, F., et al.: Adaptive neural network finite-time output feedback control of quantized nonlinear systems. IEEE Trans. Cybern. 48(6), 1839–1848 (2018)
11. Ling, S., et al.: Adaptive fuzzy tracking control of flexible-joint robots based on command filtering. IEEE Trans. Ind. Electron. 67(5), 4046–4055 (2020)
12. Zhu, G., Du, J.: Global robust adaptive trajectory tracking control for surface ships under input saturation. IEEE J. Oceanic Eng. 45(2), 442–450 (2018)
13. Gu, W.W., et al.: Robust adaptive control of hydraulic system with input saturation and valve dead-zone. IEEE Access 6(1), 53521–53532 (2018)
14. Wang, H.Q., Kang, S.J.: Adaptive neural command filtered tracking control for flexible robotic manipulator with input dead-zone. IEEE Access 7(1), 22675–22683 (2019)
15. Meng, D.Y., et al.: Adaptive dynamic surface control of pneumatic servo systems with valve dead-zone compensation. IEEE Access 6(1), 71378–71388 (2018)
16. Shahriari-kahkeshi, M., Rahmani, S.: Adaptive interval type-2 fuzzy dynamic surface control for uncertain nonlinear systems with unknown asymmetric dead-zone input. Trans. Inst. Meas. Control 41(2), 516–531 (2019)
17. Tao, G., Kokotovic, P.V.: Adaptive control of plants with unknown dead-zones. IEEE Trans. Autom. Control 39(1), 59–68 (1994)
18. Deng, W.X., et al.: Robust adaptive asymptotic tracking control of a class of nonlinear systems with unknown input dead-zone. J. Franklin Inst. 352(12), 5686–5707 (2015)
19. Zhang, Z.Q., et al.: Asymptotic tracking control of uncertain nonlinear systems with unknown actuator nonlinearity. IEEE Trans. Autom. Control 59(5), 1336–1341 (2014)
20. Wu, X.J., et al.: Indirect adaptive neural network dynamic surface control for non-linear time-delay systems with prescribed performance and unknown dead-zone input. IET Control Theory Appl. 12(4), 1895–1906 (2018)
21. Li, H., et al.: Adaptive finite-time tracking control of full state constrained nonlinear systems with dead-zone. Automatica 100(1), 99–107 (2019)
22. Ling, S., et al.: Fixed-time adaptive event-triggered tracking control of uncertain nonlinear systems. Nonlinear Dyn. 100, 3381–3397 (2020)
23. Bhat, S.P., Bernstein, D.S.: Continuous finite-time stabilization of the translational and rotational double integrators. IEEE Trans. Autom. Control 43(5), 678–682 (1998)
24. Lee, J., et al.: Inversion-free force tracking control of piezoelectric actuators using fast finite-time integral terminal sliding-mode. Mechatronics 57(1), 39–50 (2019)
25. Li, Y., et al.: Finite-time adaptive fuzzy output feedback dynamic surface control for MIMO nonstrict feedback systems. IEEE Trans. Fuzzy Syst. 27(1), 96–110 (2019)
26. Zhang, T., et al.: Adaptive neural network control for strict-feedback nonlinear systems using backstepping design. Automatica 36(12), 1835–1846 (2000)
27. Li, Y.H., et al.: Robust and adaptive backstepping control for nonlinear systems using RBF neural networks. IEEE Trans. Neural Networks 15(3), 693–701 (2004)
28. Zhou, J., et al.: Adaptive backstepping control of a class of uncertain non-linear systems with unknown backlash-like hysteresis. IEEE Trans. Autom. Control 49(10), 1751–1757 (2004)
29. Chang, C.W., et al.: Backstepping-based finite-time adaptive fuzzy control of unknown nonlinear systems. Int. J. Fuzzy Syst. 20(8), 2545–2555 (2018)
30. Deutscher, J., Kerschbaum, S.: Backstepping control of coupled linear parabolic PIDEs with spatially varying coefficients. IEEE Trans. Autom. Control 63(12), 4218–4233 (2018)
31. Zhang, X.F., et al.: Finite-time stabilization by state feedback control for a class of time-varying nonlinear systems. Automatica 45(5), 499–504 (2012)
32. Li, H.Y., et al.: Adaptive fuzzy backstepping tracking control for strict-feedback systems with input delay. IEEE Trans. Fuzzy Syst. 25(3), 642–652 (2017)
33. Zeghache, S., et al.: Fault tolerant control for modifield quadrotor via adaptive type-2 fuzzy backstepping subject to actuator faults. ISA Trans. 95(1), 330–345 (2019)
34. Wang, F., et al.: Finite-time adaptive fuzzy tracking control design for nonlinear systems. IEEE Trans. Fuzzy Syst. 26(3), 1207–1216 (2018)
35. Bao, J.L., et al.: Fuzzy finite-time tracking control for a class of non-affine nonlinear systems with unknown dead zones. IEEE Trans. Syst. Man Cybern. Syst. 1–12 (2018)
36. Zhang, T.P., Ge, S.S.: Adaptive dynamic surface control of nonlinear systems with unknown dead zone in pure feedback form. Automatica 44(7), 1895–1903 (2008)
37. Swaroop, D., et al.: Dynamic surface control for a class of nonlinear systems. IEEE Trans. Autom. Control 45(10), 1893–1899 (2000)
38. Xu, B., et al.: Composite neural dynamic surface control of a class of uncertain nonlinear systems in strict-feedback form. IEEE Trans. Cybern. 44(12), 2626–2634 (2014)
39. Li, Y.M., et al.: Adaptive fuzzy output feedback dynamic surface control of interconnected nonlinear pure-feedback systems. IEEE Trans. Cybern. 45(1), 138–149 (2015)
40. Elmi, M., et al.: Robust adaptive dynamic surface control of nonlinear time-varying systems in strict-feedback form. Int. J. Control Autom. Syst. 17(6), 1432–1444 (2019)
41. Liu, W., et al.: An improved adaptive neural dynamic surface control for pure-feedback systems with full state constraints and disturbance. Appl. Math. Comput. 358(1), 37–50 (2019)
42. Liu, F., et al.: Adaptive fault-tolerant time-varying formation tracking for multi-agent systems under actuator failure and input saturation. ISA Trans. 104, 145–153 (2019)
43. Baldini, A., et al.: Dynamic surface fault tolerant control for underwater remotely operated vehicles. ISA Trans. 78(1), 10–20 (2018)
APPENDIX A

The detailed proof of inequalities (50): It is known from (22) that $\rho_1, \phi, \rho$ and $\phi$ are positive real numbers greater than zero; thus according to Young’s inequality, one can get:

$$\frac{\phi \hat{\rho} \hat{\theta}}{\rho_1} \leq \frac{\phi}{\rho_1} \left( -\frac{\hat{\theta}^2}{2} + \frac{\theta^2}{2} \right)$$

(A.1)

The auxiliary terms is introduced in the right of above inequality, we have:

$$\frac{\phi \hat{\rho} \hat{\theta} d}{\rho} \leq \frac{\phi}{\rho} \left( -\frac{\hat{\theta}^2}{2} + \frac{\theta^2}{2} \right)$$

(A.2)

Let $q_1 = 1 - \beta, q_2 = \beta, q_w = \beta, l_1 = 1, l_2 = \frac{\beta}{\rho_1}$, one has the following inequality (see Lemma 3):

$$\left( \frac{\hat{\theta}^2}{\rho_1} \right)^\beta \leq (1-\beta)q_w + \frac{\hat{\theta}^2}{\rho_1}. \quad (A.3)$$

Similarly:

$$\left( \frac{\hat{\theta}^2}{\rho} \right)^\beta \leq (1-\beta)q_w + \frac{\hat{\theta}^2}{\rho}. \quad (A.4)$$

Substituting (A.4) and (A.3) into (A.2) yields (50).