HIGH-ACCURACY MOTION CONTROL OF A MOTOR SERVO SYSTEM WITH DEAD-ZONE BASED ON A SINGLE HIDDEN LAYER NEURAL NETWORK

J. Hu, *S. Cao,* *C. Xu,* *J. Yao,* *Z. Xie**

Abstract: There always exist parametric uncertainties, bounded disturbances and some other unknown nonlinearities such as the input dead-zone in physical motor servo systems, which can degrade the system's control performance. In this paper, a composite control strategy is proposed for high-accuracy motion control of a torque-controlled motor servo system with dead-zone. A smooth and continuous mathematical model is used to provide an approximate inverse transformation of the input-output dead-zone needed for feedback linearization. A single-layer neural network capable of on-line learning is designed to compensate for the inversion error, which comes from the approximate inversion. A stable weights adaption law for the on-line neural network is derived. In addition, a parameter adaptation law is also derived for handling the parametric uncertainty, and a nonlinear robust feedback term is designed to inhibit the influence of the imperfect modeling, compensation error or other disturbances. Lyapunov theorem is used to prove the stability of the proposed control algorithm with the weights and parameters adaptation law. Extensive comparative simulation results are used to illustrate the effectiveness and advancement of the proposed controller compared with several other main-stream controllers.

1. Introduction

High-accuracy motion control of motor servo systems has been catching much attention of many scientists and engineers, since it has been widely used in industrial application [1–4]. However, designers usually find it difficult to design a high-performance controller for the servo systems, since there always exist various uncertainties. We could divide them into two kinds, one is called parametric uncertainty,

---

*Jian Hu – Corresponding author; Shupeng Cao; Chenchen Xu; Jianyong Yao; Zhiwei Xie; School of Mechanical Engineering, Nanjing University of Science and Technology, Xiaolingwei200$^{\#}$, Nanjing, China, E-mail: hujiannjust@163.com

such as the unknown friction coefficient [5,6], load inertia [7] and so on [8–11]. The other is called uncertain nonlinearity, such as the external disturbance [12, 25]. These uncertainties could result in badly deteriorative control performance, bringing about undesirable control effect such as limit-cycle vibration, or even worse, leading to system instability [13]. In order to solve these problems, lots of related research has been done and scholars have acquired many achievements in this field. During the past few decades, adaptive control has been proposed and experienced rapid development with abundant papers and publications [14–17]. Based on the specific assumption that the nonlinear systems have only parametric uncertainties, systematic approaches of adaptive control were proposed in [14], which can achieve asymptotic tracking performance. As a matter of fact, modeling errors and external disturbances definitely exist in real motion systems, no matter how accurate the practical system model and identified parameter can be. However, the control performance of adaptive controller would reduce when faced with such condition. Therefore, some robust adaptive controllers have been developed in consideration of parametric uncertainties together with disturbances [18–20]. Moreover, Yao et al proposed an adaptive robust control (ARC) strategy in [21] for uncertain nonlinear systems. The ARC strategy has been effectively applied in many application occasions [22–25].

Apart from the parametric uncertainties and external disturbances discussed above, dead-zone nonlinearity usually exists in many motion control systems. Since dead-zone often leads to deteriorative tracking performance [26] and even instability, essential attention should be paid to solving this problem effectively. To attenuate the dead-zone effect well, a dead-zone inverse model combined with adaptive control was first proposed in [27] for linear systems with unmeasured dead-zone outputs. However, the approach mentioned above used discontinuous dead-zone inverse, which lead to unavoidable chattering in the control input. Hence, a smooth dead-zone inverse was designed in [12,28] and it was used in the design of an adaptive output feedback controller for motion control systems with dead-zone nonlinearity. This method can guarantee uniformly bounded tracking performance. Another way is to treat the dead-zone as a time-varying disturbance [26, 29–31]. Based on this idea, robust controllers can be used to overcome this disturbance-like term, such as sliding mode control [32, 36] and robust integral of the error control [33, 34]. In [35], the author introduced a workable adaptive controller to control nonlinear systems with dead-zone and the controller can achieve asymptotic tracking performance. However, without considering the explicit characteristics of the dead-zone, it might result in undesired tracking performance [28], especially when encountering severe dead-zone. According to the discussion above, we know it hard but worthwhile to make further research on the controller design for motion systems with dead-zone, parametric uncertainties and disturbances so as to obtain an ideal tracking performance.

In this paper, a smooth and continuous mathematical model is used to approximate inverse transformation of the dead-zone needed for controller design. A single-layer neural network capable of on-line learning is designed to compensate for the inversion error. A weights adaption law for the on-line neural network is derived. In addition, a parameters adaptation law is also derived for handling the parametric uncertainty, and a nonlinear robust feedback term is designed to inhibit

the influence of the imperfect modeling, compensation error or other disturbances. Lyapunov theorem is used to prove the stability of the proposed control algorithm. Applying the proposed controller to the nonlinear system with dead-zone, we can obtain better tracking performance compared with other different controllers.

This paper is organized as follows. Section 2 gives problem description and dynamic model. Section 3 gives the design procedure of proposed controller and its theoretical results. Section 4 presents comparative simulation results, and some conclusions are described in Section 5.
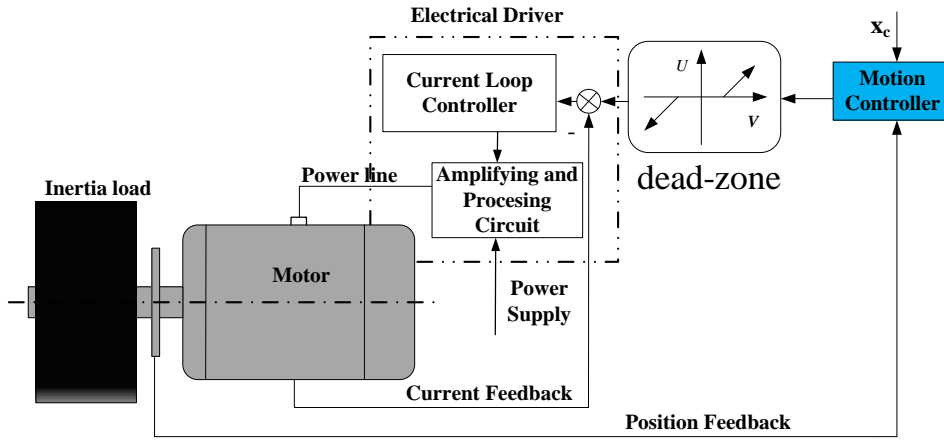
2. Problem description and dynamic model

Fig. 1 *Architecture of the motor servo system.*

The motor in the servo system considered here is a torque-controlled servo motor driven by a commercial servo electrical driver. It is connected with an inertia load through some mechanical connector such as a reducer. The system is presented in Fig. 1. The aim is to have the inertia load track any specified smooth motion trajectory x_c as close as possible. Considering the dead-zone of the actuator, the dynamic equation of the inertia load could be given as follows:

$$J\ddot{x} = k_u u(v) - B\dot{x} - (t), \tag{1}$$

where J is the inertia moment of the motor, k_u, B and $d(t)$ represent the motor torque constant, the viscous friction coefficient and the unmodeled disturbance (e.g., external disturbances, unmodeled dynamics, nonlinear friction and so on) respectively, $x, \dot{x}, \ddot{x} \in R$ mean the position, velocity and accelerator of the motor respectively. v represents the input of the actuator while u represents the output of the actuator. The characteristic of the dead-zone could be described as follows [27]:

$$u(t) = f(v(t)) = \begin{cases} m_r(v - b_r), & v \geq b_r \\ 0, & b_l < v < b_r \\ m_l(v - b_l), & v \leq b_l \end{cases}, \tag{2}$$

29

where $f(\bullet)$ represents the mapping relation of the dead-zone, $m_r > m_r > 0$, $b_r < 0, m_l < 0$ and $b_l < 0$ are the right slope, the right break-point, the left slope and the left break-point. In this paper we regard these parameters as known constants.

It is required to compute the inverse transformation in Eq. (2) to implement the control strategy. However, it is difficult to compute the inversion of Eq. (2) since it is not continuous and not smooth. Then we introduce a mapping $\hat{f}$ to approximate the actual one f. The input-output mapping of the actuator's dead-zone f and its approximate function $\hat{f}$ are depicted in Fig. 2. The exact inverse form of $\hat{f}$ could be given as follows:

$$
\begin{aligned}
v(t) &= \hat{f}^{-1}(u(t)) \\
&= \frac{u(t) + m_r b_r}{m_r} \Phi_r(u) + \frac{u(t) + m_l b_l}{m_l} \Phi_l(u),
\end{aligned}
\tag{3}
$$

where $\varphi_r(u)$ and $\varphi_l(u)$ are smooth continuous indicator functions which are defined as

$$
\Phi_r(u) = \frac{\mathrm{e}^{u/\varepsilon}}{\mathrm{e}^{u/\varepsilon} + \mathrm{e}^{-u/\varepsilon}}, \quad \Phi_l(u) = \frac{\mathrm{e}^{-u/\varepsilon}}{\mathrm{e}^{u/\varepsilon} + \mathrm{e}^{-u/\varepsilon}},
\tag{4}
$$

where ε is a positive constant needing to be selected.
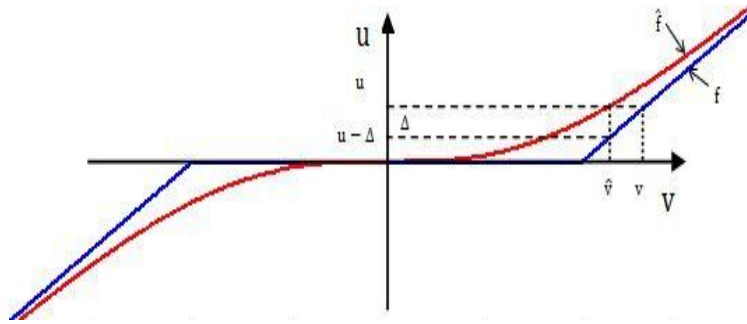

Fig. 2 *Input-output mapping of the actuator's dead-zone.*

Now we could design the control input u using feedback linearization method and then get the actual control input v by computing the inversion of u. However, since we approximate inverse transformation of the input-output dead-zone, there exists an output error of the actuator arising from the approximate inversion. We could get $\hat{v} = \hat{f}^{-1}(u)$ after we design u and the actual output of the actuator is $f(\hat{v})$. Thus the output error of the actuator could be expressed as follows.

$$
\Delta = f(v) - f(\hat{v}),
\tag{5}
$$

where $v = f^{-1}(u)$. The output error has been shown in Fig. 2.

Thus we could rewrite the system model as follows:

$$
J\ddot{x} = k_u(u(v) - \Delta) - B\dot{x} - d(t).
\tag{6}
$$

Substituting Δ into (6), we could get

$$J\ddot{x} = k_u f(\hat{v}) - B\dot{x} - d(t) = k_u u(\hat{v}) - B\dot{x} - d(t). \tag{7}$$

In fact, the Eq. (7) is equivalent to Eq. (1) in nature.
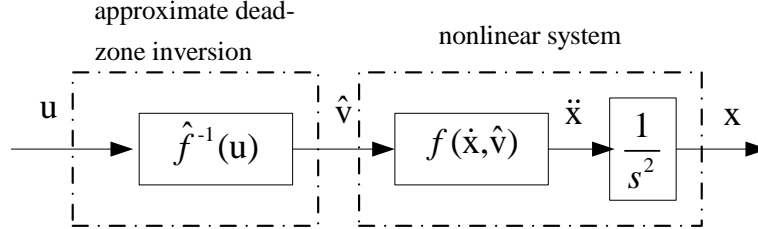

Fig. 3 *Implementation of the approximate inversion process.*

The block diagram shown in Fig. 3 depicts how to implement the approximate inversion process. For the new form of system dynamic function (6), we can utilize the approximate function $\hat{f}^{-1}$ to determine the actual control input $\hat{v}$ after designing the motor control input u.

In order to compensate for the output error of the actuator, a single hidden-layer neural network is designed later to observe Δ.

Dividing two sides of Eq. (6) by J, we can get a new form as follows:

$$\ddot{x} = \theta_1 u - \theta_2 \dot{x} - \Delta' - \tau, \tag{8}$$

where $\theta_1 = \frac{k_u}{J}, \theta_2 = \frac{B}{J}, \Delta' = \frac{k_u}{J}\Delta, \tau = \frac{d(t)}{J}$.

We can also rewrite Eq. (8) in a state-space form as follows:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \theta_1 u - \theta_2 x_2 - \Delta' - \tau, \end{cases} \tag{9}$$

where x_1 is the system's position and x_2 is the system's velocity. Define $\mathbf{x}[x_1, x_2]^{\mathrm{T}}$ as the state vector of the motor servo system.

To make the design of controller convenient, some assumptions should be made as follows.

Assumption A

1) All the systematic parameters are slowly time-varying or invariant unknown variables, that is $\dot{J} = \dot{k}_u = \dot{B} = 0$.

2) $\tau(t, x, \dot{x})$ and Δ' are time-varying unknown values, but they are bounded and the upper/lower bounds are known.

3) All the systematic parameters are bounded and the upper/lower bounds are known.

3.   Controller design

As mentioned above, there exist parametric uncertainties, bounded disturbances and dead-zone in the motor servo system. Thus a composite control strategy is proposed for high-accuracy motion control of the motor servo system. Considering the universal approximation property of neural network, it is used to approximate Δ. A parameter adaptation law is derived for handling the parametric uncertainty. Both of them could be compensated for through the feed-forward cancellation technique. A nonlinear robust feedback term is also designed to inhibit the influence of the imperfect modeling, compensation error or other disturbances. The overall control strategy diagram is shown in Fig. 4 and the design process of the controller is presented as follows in detail.
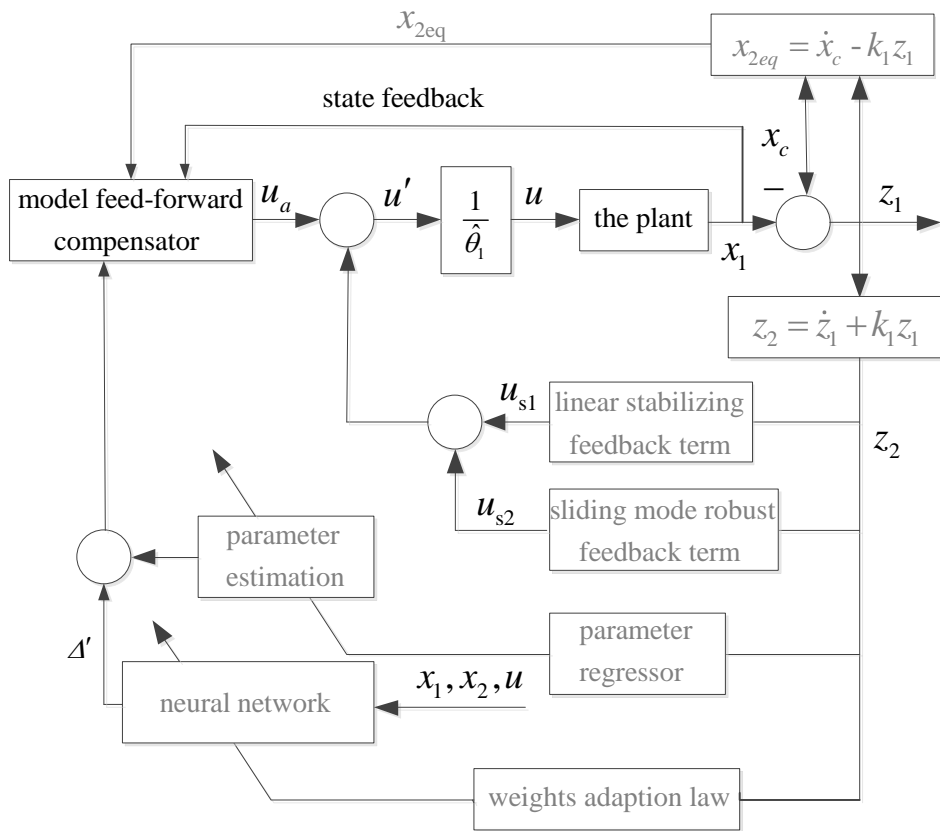

Fig. 4 *The neural network based control strategy diagram.*

Step 1: Design the desired virtual control input x_{2eq}

In this step, x_2 is treated as a virtual control input. Then a control function x_{2eq} should be designed for the virtual control input x_2 such that the output tracking

performance is guaranteed. Let x_c be the desired position and $z_1 = x_1 - x_c$ denote the position tracking error. Then we could obtain the error dynamic equation as follows:

$$\dot{z}_1 = \dot{x}_1 - \dot{x}_c = x_2 - \dot{x}_c. \tag{10}$$

If $\dot{z}_1 = \dot{x}_1 - \dot{x}_c = -k_1 z_1$, where k_1 is a positive feedback gain, we can guarantee system stability based on Lyapunov stability theorem as follows:

$$\begin{cases} V_1 = \frac{1}{2} z_1^2 > 0 \\ \dot{V}_1 = z_1 \dot{z}_1 = -k_1 z_1^2 < 0. \end{cases} \tag{11}$$

Thus let's design the desired virtual control input $x_{2\mathrm{eq}}$ as follows:

$$x_{2\mathrm{eq}} = \dot{x}_c - k_1 z_1. \tag{12}$$

However, there exists some error between x_2 and $x_{2\mathrm{eq}}$. Let's define this error as

$$z_2 = x_2 - x_{2\mathrm{eq}} = \dot{x}_1 - \dot{x}_c + k_1 z_1 = \dot{z}_1 + k_1 z_1. \tag{13}$$

Since $G(s) = \frac{z_1(s)}{z_2(s)} = \frac{1}{s+k_1}$ is a stable transfer function, z_1 can converge to zero when z_2 converges to zero. Now our main task is to make z_2 converge to zero. By taking the derivative of z_2 combining with Eq. (9), we have

$$\dot{z}_2 = \dot{x}_2 - \dot{x}_{2\mathrm{eq}} = \theta_1 u - \theta_2 x_2 - \Delta' - \tau - \dot{x}_{2\mathrm{eq}}. \tag{14}$$

Step 2: Design the actual control input u

Noting (14), a feedback linearization controller can be designed as

$$u = (\theta_{2n} x_2 + \Delta' + \tau + \dot{x}_{2\mathrm{eq}} - k_2 z_2)/\theta_{1n}, \tag{15}$$

where $k_2 > 0$ is a feedback gain, θ_{1n} and θ_{2n} are the nominal values of θ_1 and θ_2.

If the exact system model is known, i.e., $\theta_{1n} = \theta_1$, $\theta_{2n} = \theta_2$, $\Delta' = 0$ and $\tau = 0$, we could define a Lyapunov function as follows:

$$V_2 = \frac{1}{2} z_1^2 + \frac{1}{2} z_2^2 > 0, \tag{16}$$

then

$$\begin{aligned} \dot{V}_2 &= z_1(z_2 - k_1 z_1) + z_2(-k_2 z_2) \\ &= -k_1(z_1 - \frac{1}{2k_1} z_2)^2 - (k_2 - \frac{1}{4k_1}) z_2^2 < 0, \end{aligned} \tag{17}$$

provided $\left(k_2 - \frac{1}{4k_1}\right) \geq 0$.

Thus the designed feedback linearization controller Eq. (15) can obtain asymptotic tracking performance. Unfortunately, θ_{1n} and θ_{2n} are uncertain parameters which are not equal to θ_1 and θ_2. $\hat{\Delta}$ is the model inversion error which would not be zero and τ is a lumped disturbance which would not be zero either. Thus, we need to improve our controller. An improved controller is designed as follows:

$$u = (\hat{\theta}_2 x_2 + \hat{\Delta}' + \dot{x}_{2\mathrm{eq}} - k_2 z_2 + u_{s2})/\hat{\theta}_1, \tag{18}$$

where $\hat{\theta}_1$, $\hat{\theta}_2$ and $\hat{\Delta}'$ represent the estimation of θ_1, θ_2 and Δ' respectively. u_{s2} is a nonlinear robust feedback item used to compensate the error of parameter estimation, approximation error of dead-zone model inversion as well as the external disturbance.

Substituting Eq. (18) into Eq. (14), we could obtain the error dynamic equation of z_2 as follows:

$$\dot{z}_2 = -\tilde{\theta}^{\mathrm{T}}\varphi - \tilde{\Delta}' - k_2 z_2 + u_{s2} - \tau, \tag{19}$$

where $\tilde{\theta} = [\tilde{\theta}_1, \tilde{\theta}_2]^{\mathrm{T}} = [\hat{\theta}_1 - \theta_1, \hat{\theta}_2 - \theta_2]^{\mathrm{T}}$, $\varphi = [u, -x_2]^{\mathrm{T}}$, $\tilde{\Delta}' = \Delta' - \hat{\Delta}'$.

Step 3: Design the parameter adaptation law

In this step, our goal is to design a parameter adaption law to estimate the unknown parameter $\theta = [\theta_1, \theta_2]$.

A discontinuous projection is given as follows:

$$\mathrm{Proj}_{\hat{\theta}}(i) \begin{cases} 0, & \text{if } \hat{\theta} = \theta_{\max} \text{ and } i > 0 \\ 0, & \text{if } \hat{\theta} = \theta_{\min} \text{ and } i < 0 \\ i, & \text{otherwise,} \end{cases} \tag{20}$$

where $\theta_{\max}$ and $\theta_{\min}$ represent the upper and lower bounds of θ respectively. The purpose of using discontinuous projection (20) is to make the estimation of system parameters bounded, so that the estimation values of parameters could be within a range near their true values and then the controller could be stable.

Then, the parameter adaptation law of θ is given by

$$\dot{\hat{\theta}} = \mathrm{Proj}_{\hat{\theta}}(\Gamma_1 \chi), \tag{21}$$

where $\Gamma_1 > 0$ is the slope of adaptation law, and χ is the adaptation function to be designed later. For the function χ, there is an inequality guaranteed as below:

$$\tilde{\theta}^{\mathrm{T}} \left[\mathrm{Proj}_{\hat{\theta}}(\Gamma_1 \chi) - \Gamma_1 \chi \right] \leq 0, \tag{22}$$

χ is chosen as follows:

$$\chi = \mathbf{Z}^{\mathrm{T}} P b \varphi, \tag{23}$$

where $\mathbf{Z} = [z_1, \dot{z}_1]^{\mathrm{T}}$, $b = [0, 1]^{\mathrm{T}}$, and P is a (2×2) symmetric positive definite matrix satisfying

$$PA + \mathbf{A}^{\mathrm{T}} P = -Q, \tag{24}$$

where $\mathbf{A}$ is a Hurwitz matrix, and

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -k_1 k_2 & -(k_1 + k_2) \end{bmatrix}. \tag{25}$$

Step 4: Design a neural network based observer

In this step, we will design an approximator to approximate $\hat{\Delta}$ which represents the output error of the actuator. As is known to all, neural network is capable of approximating any nonlinear function within arbitrary accuracy when given a

sufficient number of hidden layer neurons together with essential input information. A single hidden layer neural network is used here since it has a simple structure and it is trained online to approximate $\hat{\Delta}$.

The input-output mapping of the single layer neural network can be given as follows:

$$f(\mathbf{x}) = \mathbf{W}^{*\mathrm{T}} h(x) + \varepsilon_{\mathrm{app}} = \Delta', \tag{26}$$

$$h_j = \exp\left(\frac{\|x - c_j\|^2}{2b_j^2}\right), \tag{27}$$

where $\mathbf{x} = [x_1, x_2, u]^{\mathrm{T}}$ is the input vector of the network, j is the j-th node of the hidden layer, $\mathbf{W}^*$ is the ideal weight value of the network, $\mathbf{h} = [h_1, h_2, \ldots, h_j]^{\mathrm{T}}$ is the output of network's Guass radial function, $\varepsilon_{\mathrm{app}} < \varepsilon_N$ is the approximation error of the network.

The actual output of the network is

$$\hat{f}(\mathbf{x}) = \hat{\mathbf{W}}^{\mathrm{T}} h(x) = \hat{\Delta}', \tag{28}$$

where $\hat{\mathbf{W}}$ is the estimated weights.

Design the weight adaptation law as follows:

$$\dot{\hat{\mathbf{W}}} = -\Gamma_2 \Psi, \tag{29}$$

where Γ_2 is the weights adaptation velocity matrix, and ψ is the adaptation function to be designed. Then we could obtain the error dynamic equation of z_2 as follows:

$$\dot{z}_2 = -k_2 z_2 - \tilde{\theta}^{\mathrm{T}} \varphi + \tilde{\mathbf{W}}^{\mathrm{T}} h(x) + u_{s2} - \varepsilon_{\mathrm{app}} - \tau, \tag{30}$$

where $\tilde{\mathbf{W}} = \hat{\mathbf{W}} - \mathbf{W}^*$ is the estimation error between estimated and ideal weight.

ψ is chosen as

$$\psi = \mathbf{Z}^{\mathrm{T}} \mathrm{Pb} h(x). \tag{31}$$

Step 5: Design the sliding mode robust feedback term

In this step, a sliding mode robust feedback term u_{s2} will be designed to overcome the influence of parameter estimation error, approximation error of dead-zone effect and the external disturbance so as to guarantee the system stability.

Design $u_{s2} = -\mathrm{sgn}(\mathbf{z}^{\mathrm{T}} \mathrm{Pb}) h$, where h represents the upper bound of the sum of $|\varepsilon_{\mathrm{app}}|$ and $|\tau|$. Then, there is a property given as follows:

$$\mathbf{z}^{\mathrm{T}} \mathrm{Pb} \left[u_{s2} - \varepsilon_{\mathrm{app}} - \tau \right] \leq 0, \tag{32}$$

where $\delta_1 \geq |\tau|, \delta_2 \geq |\varepsilon_{\mathrm{app}}|, h \geq \delta_1 + \delta_2$.

Theorem 1. By designing the discontinuous projection-type parameter adaptation law Eqs. (21), (23) and weight adaptation law Eqs. (29), (31), the control input Eq. (18) can guarantee asymptotic tracking performance of the system, in other words, $\mathbf{z} \triangleq [z_1, z_2] \to 0$ as $t \to \infty$.

Remark 1. Theoretically speaking, sliding mode robust feedback term u_s can help to acquire system asymptotic stability, but sliding mode control achieve this goal by constantly switching control directions, which may lead to unacceptable chattering around the zero in actual systems. If the chattering is too big to put up with, we can transform it to a high frequency feedback robust control [32], sacrificing part control accuracy by set an eclectic or comparative proper band width of the error so as to reduce chattering, and then we can achieve bounded stability.

Remark 2. In terms of computational burden, the proposed controller has certain complexity. Using this controller, the first step is to calculate the control value u, which contains a term of the estimation of model inversion error $\hat{\Delta}'$ and it is calculated by a neural network. As we all know, the calculation of neural network is large, however it mainly depends on the number of neurons. As long as we choose a proper number of neurons, we can take into account both the calculation accuracy and the calculation amount. It needs some experience in engineering. Then the second step is to use the given Eq. (3) to find the inverse of u to output the final control value $\hat{v}$. Eq. (3) seems a little bit complex since it contains some exponential operations. However, the expansion of Taylor series can be used to approximate the exponential operation so as to reduce the amount of calculation.

Remark 3. Here some guidelines about how to select the proposed controller's parameters values and the effect of each parameter are given. k_1 and k_2 are feedback gains. The larger they are, the faster the convergence rate of the controlled system is, but large feedback gain is easy to cause oscillation and instability. Thus it is necessary to select a suitable value, which is often determined according to the debugging effect in practice. Γ_1, Γ_2 are positive constants, which determine the parameter adaptation rate and neural network weights adaptation rate. Usually we choose an appropriate value for them according to experiment results. h represents the upper bound of the sum of neural network approximation error and external disturbance. Sometimes it is difficult to determine its real value, but we just choose a relative large value for it.

Proof. See Appendix. □

4. Comparative simulation results

In this section, to verify the effectiveness of proposed ARCNN controller in this paper, we will compare the tracking performance of another four kinds of common controllers with that of the ARCNN controller under two working conditions, that is, high and low frequency tracking modes.

The total five different controllers are listed below.

1) PID: This is the well known traditional three-loop proportional-integral-derivative controller. Basing on position-loop, we choose $k_p = -900$, $k_i = -6000$, $k_d = 0$ in simulation, which represent proportional gain, integral gain and derivative gain respectively.

2) FBL: This is the feedback linearization controller. The control parameters are chosen as $k_1 = 0.05$, $k_2 = 0.005$.

3) FBLNN: This is the feedback linearization controller with neural network, in which the network is also used to compensate the dead-zone error. We choose the control parameters as $k_1 = 0.05$, $k_2 = 0.005$, $\Gamma_2 = \text{diag}\{0.1, 0.1, 0.1, 0.1, 0.1\}$.

4) ARC: This is the adaptive robust controller with adaptive law Eq. (21). And the nonlinear robust feedback term u_s is designed to not only overcome the external disturbance but also attenuate the whole effect caused by the dead-zone. Hence, a larger constant h is to be chosen. The control parameters are choose as $\Gamma_1 = \text{diag}\{0.009, 0.009\}$, $k_1 = 25$, $k_2 = 1$, $h = 0.05$. The initial estimated value of $\hat{\theta}$ is chosen as $[0.001, 0.001]$. The bounds of uncertain parameters are set at $[0.0005, 0.004]$.

5) ARCNN: This is the adaptive robust controller with neural network and adaptive law Eq. (21), which is proposed in this paper and discussed in the previous sections. This control parameters are chosen as $\Gamma_1 = \text{diag}\{0.009, 0.009\}$, $\Gamma_2 = \text{diag}\{0.1, 0.1, 0.1, 0.1, 0.1\}$, $k_1 = 13$, $k_2 = 4.95$, $h = 0.01$. The initial estimated value of $\hat{\theta}$ is chosen as $[0.001, 0.001]$. The bounds of uncertain parameters are set at $[0.0005, 0.004]$.

Case 1. High frequency tracking mode
In this case, we set the motion trajectory as $x_c = (1 - \exp(-0.1 \times t)) \times \sin(1.0 \times t)$.

Case 2. Low frequency tracking mode
In this case, we set the motion trajectory as $x_c = (1 - \exp(-0.2 \times t)) \times \sin(0.2 \times t)$.

The simulation time in the high frequency tracking mode is chosen as 50s, since the frequency of the desired motion trajectory is high and this time is enough to show the tracking performance of all the controllers, while the simulation time in the low frequency tracking mode is chosen as 100s, since the frequency of the desired motion trajectory is low and we need to extend simulation time to show the tracking performance of all the controllers adequately.

The designed motion trajectory and corresponding tracking performance of the five controllers in high frequency and low frequency tracking modes are shown in Fig. 5 and Fig. 6.
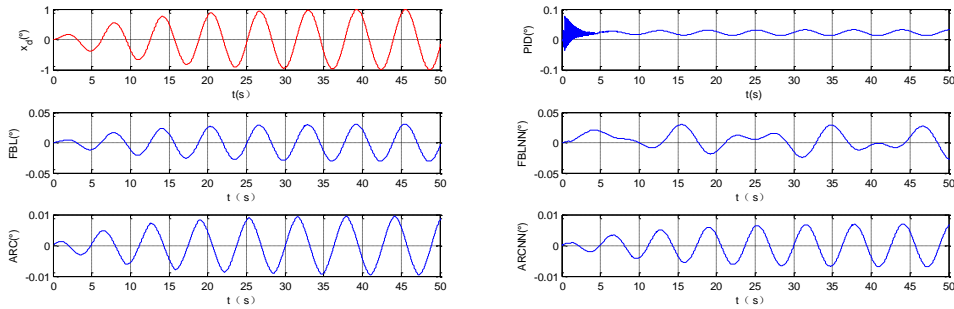

Fig. 5 *Tracking errors of the five controllers in the high frequency tracking mode.*
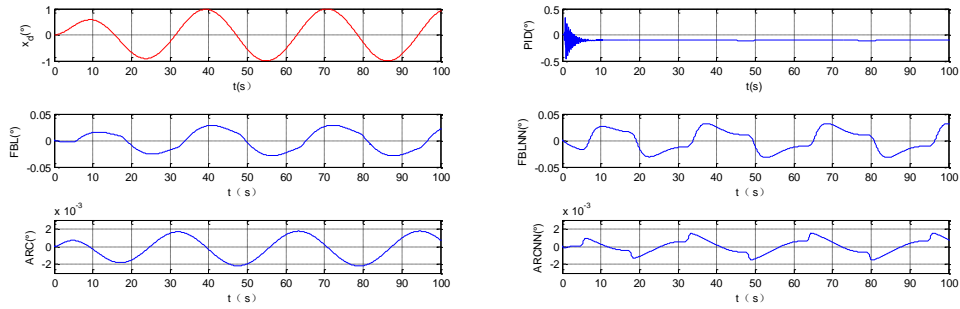
37

Fig. 6 *Tracking errors of the five controllers in the low frequency tracking mode.*

As we can see from Fig. 5 and Fig. 6, in both cases the ARCNN controller achieves the best tracking performance from the view of tracking error. The PID control strategy is not based on system model, in another word, it has no model compensation. Thus its tracking performance is quite undesirable with large stable-state error and fierce chatter in transient state. The ARC controller has a better tracking performance than that of the PID controller and the FBL controller, due to its learning capability brought by parameter estimation as well as the existence of the nonlinear robust feedback term designed to attenuate the effect of the disturbance. Obviously, the ARCNN controller has a good robustness since it inherits this advantage from the ARC controller. Furthermore, ARCNN controller estimates the dead-zone and compensates it effectively. Thus it has the best control performance. In addition, the system switches more frequently in both sides of the dead-zone in the high frequency tracking mode, which is the reason why the ARCNN has a better performance in low frequency tracking mode. The specific tracking indexes and the values are listed in Tab. I (see Appendix) which can help verify the effectiveness of the ARCNN controller more precisely.

Fig. 7 and Fig. 8 show the designed velocity trajectory and velocity tracking errors of the five controllers in the two modes. The performance of the ARCNN controller is also the most satisfactory with the minimal tracking error. In the low frequency tracking mode, we can see distinctly that the time the ARCNN controller
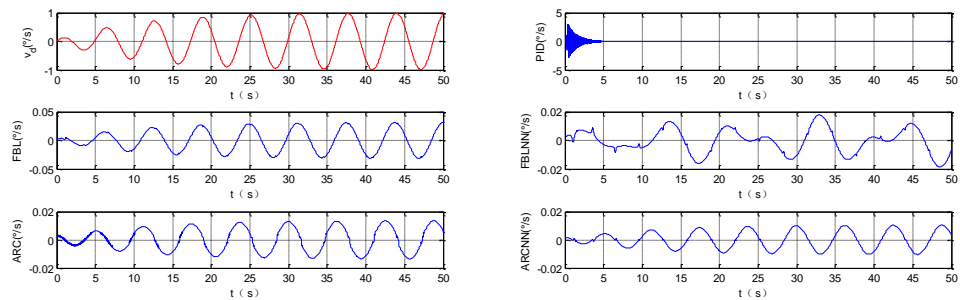

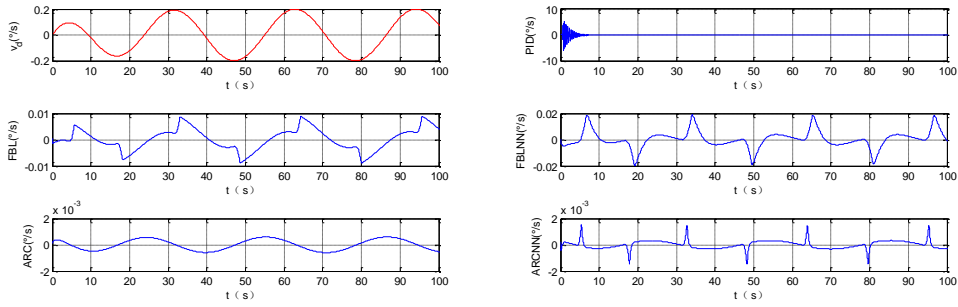**Fig. 7** *Velocity tracking errors of the five controllers in the high frequency tracking mode.*

Fig. 8 *Velocity tracking errors of the five controllers in the low frequency tracking mode.*

takes to cross the dead-zone is much less that of all the other controllers, which indicate that the ARCNN controller can compensate the effects of the dead-zone effectively.

Fig. 9 and Fig. 10 show the parameter estimation of ARC and ARCNN controllers in the two modes. It is easy and distinct to see that the adaptive law designed in ARCNN controller can make the estimated parameter reach its truth
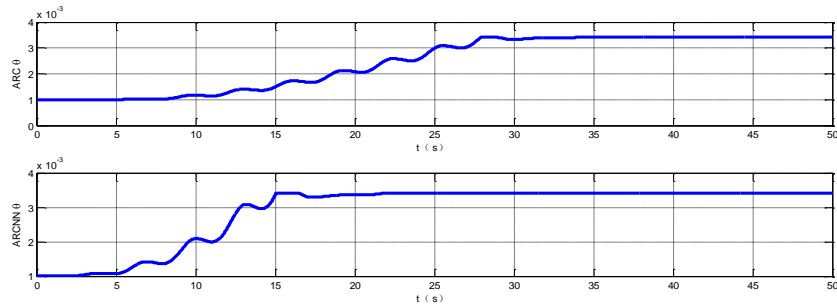

Fig. 9 *Parameter estimation of ARC and ARCNN controllers in the high frequency tracking mode.*
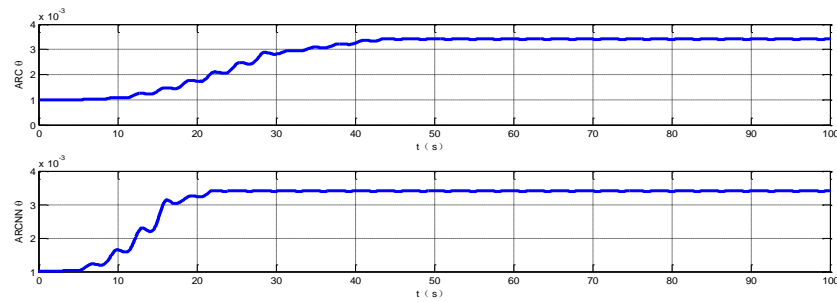

Fig. 10 *Parameter estimation of ARC and ARCNN controllers in the low frequency tracking mode.*

value more quickly. In addition, for the same controller, the estimation process is shorter in low frequency tracking mode than that in high frequency tracking mode.

The network tracking performance of FBLNN and ARCNN controllers in both modes is shown in Fig. 11 and Fig. 12. From the two figures, we can draw a conclusion that the network integrated with ARC controller estimates the dead-zone error in a relatively better accuracy compared with that integrated with FBL controller. In addition, in the high frequency tracking mode, the tracking performance is much worse than that in another mode due to the fact that the motion system crosses the dead-zone more frequently, making the dead-zone error be a heartbeat-like spike curve, which leads to a harder estimation process for the network.
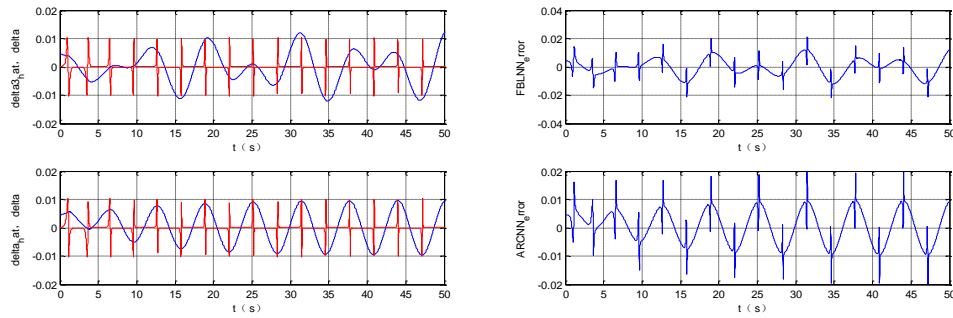

Fig. 11 *Network tracking performance of FBLNN and ARCNN controllers in the high frequency tracking mode.*
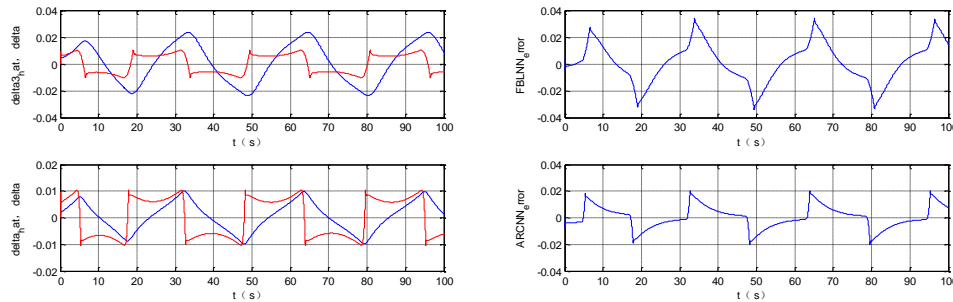

Fig. 12 *Network tracking performance of FBLNN and ARCNN controllers in the low frequency tracking mode.*

5. Conclusion

In this paper, a composite control strategy has been proposed for a motor servo system subjected to dead-zone nonlinearity, the parametric uncertainty and unknown bounded disturbance. A smooth dead-zone inverse is designed to approximately substitute the actual dead-zone inverse so as to guarantee the continuity of actual control input and then overcome fierce system chattering. Moreover a

single-hidden-layer neural network is applied to compensate the control input error resulted from the approximate dead-zone inverse. Furthermore the parametric uncertainty is solved by designing an adaptation law via Lyapunov method. At last a sliding mode robust feedback term is designed to overcome the integrated disturbances including the external disturbance and error of network estimation. The proposed controller can guarantee the motion system obtain an asymptotic stability. The effectiveness of the proposed controller has been verified by comparing the simulation result of the proposed controller with those of another four different controllers.

Acknowledgement

References

[1] HU C. YAO B., WANG Q. Adaptive robust precision motion control of systems with unknown input dead-zones: A case study with comparative experiments, IEEE Trans. Ind. Electron., Jun. 2011, 58(6), pp. 2454–2464.

[2] YAO J., JIAO Z., MA D. RISE-based precision motion control of DC motors with continuous friction compensation, IEEE Trans. Ind. Electron. 2014, 61(12) pp. 7067–7075.

[3] KOSTARIGKA A.K., DOULGERI Z. ROVITHAKIS G.A. Prescribed performance tracking for flexible joint robots with unknown dynamics and variable elacticity, Automatica 49 (5) (2013) 1137–1147.

[4] HUANG W.-S., LIU C.-W., HSU P.-L., YEH S.-S. Precision control and compensation of servomotors and machine tools via the disturbance observer, *IEEE Trans. Ind. Electron.*, 2010, 57(1), pp. 420–429.

[5] LU L., YAO B., WANG Q., CHEN Z. Adaptive robust control of linear motors with dynamic friction compensation using modified LuGre model, *Automatica*, 2009, 45(12), pp. 2890–2896.

[6] YAO J., JIAO Z., YAO B. Robust control for static loading of electrohydraulic load simulator with friction compensation, *Chin. J. Aeronaut.*, 2012, 25(6), pp. 954–962.

[7] YAO J., JIAO Z., MA D., YAN L. High-accuracy tracking control of hydraulic rotary actuators with modeling uncertainties, *IEEE/ASME Trans. Mechatronics.* doi: 10.1109/TMECH.2013.2252360. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber$=$6490408

[8] YAO J., JIAO Z., YAO B., SHANG Y., DONG W. Nonlinear adaptive robust force control of hydraulic load simulator, *Chin. J. Aeronaut.*, 2012, 25(5), pp. 766–775.

[9] YAO B., BU F., REEDY J., CHIU G.T.C. Adaptive robust motion control of single-rod hydraulic actuators: Theory and experiments, *IEEE/ASME Trans. Mechatronics*, 2000, 5(1), pp. 79–91.

[10] VILLWOCK S., PACAS M. Time-domain identification method for detecting mechanical backlash in electrical drives, *IEEE Trans. Ind. Electron.*, 2009, 56(2), pp. 568–573.

[11] CAMPOS J., LEWIS F.L., Selmic R. Backlash compensation with filtered prediciton in discrete time nonlinear systems by dynamic inversion using neural networks, *submitted to the Asian Journal of Control*, November, 1999.

[12] DENG W., YAON J., MA D. Robust adaptive asymptotic tracking control of a class of nonlinear systems with unknown input dead-zone, Journal of the Franklin Institute, 2015, 352, pp. 5686–5707.

[13] TAO G., LEWIS F.L. *Adaptive Control of Nonsmooth Dynamic Systems.* New York, NY, USA: Springer-Verlag, 2001.

[14] Kanellakopoulos I., Kokotovic P.V., Morse A.S. Systematic design of adaptive controllers for feedback linearizable systems, IEEE Trans. Autom. Control, 1991, 36(11), pp. 1241–1253.

[15] KRISTIC M., KOKOTOVIC P.V., KANELLAKOPOULOS I. Nonlinear and Adaptive Control Design, John Wiley, New York,1995.

[16] LI S., LIU Z. Adaptive speed control for permanent-magnet synchronous motor system with variations of load inertia, *IEEE Trans. Ind. Electron.*, 2009, 56(8), pp. 3050–3059.

[17] ABOOTORABI ZARCHI H., SOLTANI J., ARAB MARKADEH G. Adaptive input–output feedback-linearization-based torque control of synchronous reluctance motor without mechanical sensor, *IEEE Trans. Ind. Electron.*, 2010, 57(1), pp. 375–384.

[18] GE S.S., LI Z. Robust adaptive control for a class of MIMO nonlinear systems by state and output feedback, IEEE Trans. Autom. Control, 2014, 59(6), pp. 1624–1629.

[19] YAO B., TOMIZUKA M. Smooth robust adaptive sliding mode control of robot manipulators with guaranteed transient performance, *Trans. ASME, J. Dyn. Syst., Meas. Control*, 1996, 118(4), pp. 764–775.

[20] LIU S., SU C.-Y., LI Z. Robust adaptive inverse control of a class of nonlinear systems with Prandtl–Ishlinskii hysteresis model, IEEE Trans. Autom. Control, 2014, 59(8), pp. 2170–2175.

[21] YAO B., Tomizuka M. Adaptive robust control of SISO nonlinear systems in a semi-strict feedback form, Automatica, 1997, 33(5), pp. 893–900.

[22] SUN W., GAO H., YAO B. Adaptive robust vibration control of full-car active suspensions with electrohydraulic actuators, IEEE Trans. Control Syst. Technol., 2013, 21(6), pp. 2417–2422.

[23] YAO J., JIAO Z., MA D. High dynamic adaptive robust control of load emulator with output feedback signal, J.Frankl. Inst. – Eng. Appl. Math., 2014, 351(8), pp. 4415–4433.

[24] YAO J., JIAO Z., HAN S. Friction compensation for low velocity control of hydraulic flight motion simulator: a simple adaptive robust approach, Chin. J. Aeronaut., 2013, 26(3), pp. 814–822.

[25] YAO J., JIAO Z., MA D. Adaptive robust control of dc motors with extended state observer, IEEE Trans. Ind.Electron. 2014, 61(7), pp. 3630–3637.

[26] IBRIR S., XIE W.-F., SU C.-Y. Adaptive tracking control of nonlinear systems with non-symmetric dead-zone input, Automatica, 2007, 43(3), pp. 522–530.

[27] TAO G., KOKOTOVIC P.V. Adaptive control of plants with unknown dead-zones, IEEE Trans. Autom. Control, 1994, 39(1), pp. 59–68.

[28] ZHOU J., WEN C., ZHANG Y. Adaptive output control of nonlinear systems with uncertain dead-zone nonlinearity, IEEE Trans. Autom. Control, 2006, 51(3), pp. 504–511.

[29] ZHANG Z., XU S., ZHANG B. Asymptotic tracking control of uncertain nonlinear systems with unknown actuator nonlinearity, IEEE Trans. Autom. Control, 2014, 59(5), pp. 1336–1341.

[30] MA H.-J., YANG G.-H. Adaptive output control of uncertain nonlinear systems with non-symmetric dead-zone input, Automatica, 2010, 46(2), pp. 413–420.

[31] HUA C., WANG Q., GUAN X. Adaptive tracking controller design of nonlinear systems with time delays and unknown dead-zone input, IEEE Trans. Autom. Control, 2008, 53(7), pp. 1753–1759.

[32] YONG-CHUN F., GUI-ZHANG L. Nonlinear system theory [M]. Tsinghua University Press, 2009, pp. 81–90.

[33] GINOYA D., SHENDGE P.D., PHADKE S.B. Sliding mode control for mismatched uncertain systems using an extended disturbance observer, IEEE Trans. Ind. Electron., 2014, 61(4), pp. 1983–1992.

42

[34] XIAN B., DAWSON D.M., DE QUEIROZ M.S., CHEN J. A continuous asymptotic tracking control strategy for uncertain nonlinear systems, IEEE Trans. Autom. Control, 2004, 49(7), pp. 1206–1211.

[35] HUA C., WANG Q., GUAN X. Adaptive tracking controller design of nonlinear systems with time delays and unknown dead-zone input, IEEE Trans. Autom. Control, 2008, 53(7), pp. 1753–1759.

[36] GINOYA D., SHENDGE P.D., PHADKE S.B. Sliding mode control for mismatched uncertain systems using an extended disturbance observer, IEEE Trans. Ind. Electron., 2014, 61(4), pp. 1983–1992.

Appendix

	High frequency tracking mode		Low frequency tracking mode	
	mean	standard deviation	mean	standard deviation
PID	0.0221	0.0075	−0.1040	0.0270
FBL	−4.9308e − 004	0.0196	−8.1583e − 004	0.0197
FBLNN	0.0027	0.0142	−7.2267e − 004	0.0201
ARC	−6.8101e − 005	0.0059	−2.3971e − 004	0.0013
ARCNN	−4.3668e − 005	0.0044	−2.0023e − 005	7.4499e − 004

Tab. I *The means and standard deviations of tracking results of the five controllers working in two modes.*

Proof of theorem 1 Defining $\mathbf{Z} = [z_1, \dot{z}_1]^{\mathrm{T}}$, we have

$$
\begin{aligned}
\dot{\mathbf{Z}} &= \begin{bmatrix} \dot{z}_1 \\ \ddot{z}_1 \end{bmatrix} = \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 - k_1 \dot{z}_1 \end{bmatrix} \\
&= \begin{bmatrix} \dot{z}_1 \\ -\tilde{\theta}^{\mathrm{T}}\varphi + \tilde{W}^{\mathrm{T}}h(x) + u_{s2} - \varepsilon_{\mathrm{app}} - \tau - k_2(\dot{z}_1 + k_1 z_1) - k_1 \dot{z}_1 \end{bmatrix} \\
&= \begin{bmatrix} 0 & 1 \\ -k_1 k_2 & -(k_1 + k_2) \end{bmatrix} \begin{bmatrix} z_1 \\ \dot{z}_1 \end{bmatrix} + \\
&\quad + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \left[-\tilde{\theta}^{\mathrm{T}}\varphi + \tilde{W}^{\mathrm{T}}h(x) + u_{s2} - \varepsilon_{\mathrm{app}} - \tau \right] \\
&= AZ + b\left[-\tilde{\theta}^{\mathrm{T}}\varphi + \tilde{W}^{\mathrm{T}}h(x) + u_{s2} - \varepsilon_{\mathrm{app}} - \tau \right].
\end{aligned}
\tag{33}
$$

Since $\mathbf{A}$ is a Hurwitz matrix, we have

$$
\mathbf{PA} + \mathbf{A}^{\mathrm{T}}\mathbf{P} = -\mathbf{Q},
\tag{34}
$$

where P represents a symmetric positive definite matrix.

Consider the Lyapunov function candidate

$$
V = \frac{1}{2}Z^T P Z + \frac{1}{2}\Gamma_1^{-1}\tilde{\theta}^T\tilde{\theta} + \frac{1}{2}\Gamma_2^{-1}\tilde{W}^T\tilde{W}.
\tag{35}
$$

Thus, the time derivative of V is

$$
\begin{aligned}
\dot{V} &= \mathbf{Z}T\mathbf{P}\dot{\mathbf{Z}} + \Gamma_1^{-1}\tilde{\theta}^{\mathrm{T}}\dot{\hat{\theta}} + \Gamma_2^{-1}\tilde{\mathbf{W}}^{\mathrm{T}}\dot{\hat{\mathbf{W}}} \\
&= \mathbf{Z}^{\mathrm{T}}\mathbf{P}\{\mathbf{A}\mathbf{Z} + b\left[-\tilde{\theta}^{\mathrm{T}}\varphi + \tilde{\mathbf{W}}^{\mathrm{T}}h(x) + u_{s2} - \varepsilon_{\mathrm{app}} - \tau\right]\} + \Gamma_1^{-1}\tilde{\theta}^{\mathrm{T}}\dot{\hat{\theta}} + \\
&\quad + \Gamma_2^{-1}\tilde{\mathbf{W}}^{\mathrm{T}}\dot{\hat{\mathbf{W}}} \\
&= \frac{1}{2}(\mathbf{Z}^{\mathrm{T}}\mathbf{PAZ} + \mathbf{Z}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{PZ}) + \mathbf{Z}^{\mathrm{T}}\mathbf{P}b * \left[-\tilde{\theta}^{\mathrm{T}}\varphi + \tilde{\mathbf{W}}^{\mathrm{T}}h(x) + u_{s2} - \varepsilon_{\mathrm{app}} - \tau\right] + \\
&\quad + \Gamma_1^{-1}\tilde{\theta}^{\mathrm{T}}\dot{\hat{\theta}} + \Gamma_2^{-1}\tilde{\mathbf{W}}^{\mathrm{T}}\dot{\hat{\mathbf{W}}} \\
&= \frac{1}{2}\mathbf{Z}^{\mathrm{T}}(\mathbf{PA} + \mathbf{A}^{\mathrm{T}}\mathbf{P})\mathbf{Z} + \mathbf{Z}^{\mathrm{T}}\mathbf{P}b * \left[-\tilde{\theta}^{\mathrm{T}}\varphi + \tilde{\mathbf{W}}^{\mathrm{T}}h(x) + u_{s2} - \varepsilon_{\mathrm{app}} - \tau\right] + \\
&\quad + \Gamma_1^{-1}\tilde{\theta}^{\mathrm{T}}\dot{\hat{\theta}} + \Gamma_2^{-1}\tilde{\mathbf{W}}^{\mathrm{T}}\dot{\hat{\mathbf{W}}} \\
&= -\frac{1}{2}\mathbf{Z}^{\mathrm{T}}Q\mathbf{Z} + \mathbf{Z}^{\mathrm{T}}\mathbf{P}b\left[-\tilde{\theta}^{\mathrm{T}}\varphi + \tilde{\mathbf{W}}^{\mathrm{T}}h(x) + u_{s2} - \varepsilon_{\mathrm{app}} - \tau\right] + \\
&\quad + \Gamma_1^{-1}\tilde{\theta}^{\mathrm{T}}\dot{\hat{\theta}} + \Gamma_2^{-1}\tilde{\mathbf{W}}^{\mathrm{T}}\dot{\hat{\mathbf{W}}} \\
&= -\frac{1}{2}\mathbf{Z}^{\mathrm{T}}Q\mathbf{Z} + \mathbf{Z}^{\mathrm{T}}\mathbf{P}b\left[u_{s2} - \tau - \varepsilon_{\mathrm{app}}\right] + \\
&\quad + \Gamma_1^{-1}\tilde{\theta}^{\mathrm{T}}(\dot{\hat{\theta}} - \Gamma_1\mathbf{Z}^{\mathrm{T}}\mathbf{P}b\varphi) + \Gamma_2^{-1}\tilde{\mathbf{W}}^{\mathrm{T}}\left[\dot{\hat{\mathbf{W}}} + \Gamma_2\mathbf{Z}^{\mathrm{T}}\mathbf{P}bh(x)\right].
\end{aligned} \tag{36}
$$

Choose the sliding mode robust feedback term, the parameter adaptation law and the weight adaptation law as

$$
\begin{cases}
u_{s2} = -\mathrm{sgn}(\mathbf{Z}^{\mathrm{T}}Pb)h(x) \\
\dot{\hat{\theta}} = \mathrm{Proj}_{\hat{\theta}}(\Gamma_1\mathbf{Z}^{\mathrm{T}}Pb\varphi) \\
\dot{\hat{\mathbf{W}}} = -\Gamma_2\mathbf{Z}^{\mathrm{T}}\mathrm{P}bh(x)
\end{cases} . \tag{37}
$$

Substituting Eq. (37) in Eq. (36), we have

$$
\begin{aligned}
\dot{\mathbf{V}} &= -\frac{1}{2}\mathbf{Z}^{\mathrm{T}}\mathbf{Q}\mathbf{Z} + \mathbf{Z}^{T}Pb\left[u_{s2} - \tau - \varepsilon_{\mathrm{app}}\right] + \Gamma_1^{-1}\tilde{\theta}^{T} \times \\
&\quad \left[\mathrm{Proj}_{\hat{\theta}}(\Gamma_1\mathbf{Z}^{\mathrm{T}}Pb\varphi) - \Gamma_1\mathbf{Z}^{\mathrm{T}}Pb\varphi\right] + \Gamma_2^{-1}\tilde{\mathbf{W}}^{\mathrm{T}} \times \\
&\quad \left[-\Gamma_2\mathbf{Z}^{\mathrm{T}}Pbh(x) + \Gamma_2\mathbf{Z}^{\mathrm{T}}Pbh(x)\right] \leq 0
\end{aligned} \tag{38}
$$

Thus, employing the proposed ARCNN controller, the motion system considered in this paper can obtain asymptotic stability theoretically, namely, $z_1 \to 0$ as $t \to \infty t$.