01 Jan 2006

Neuro Control of Nonlinear Discrete Time Systems with Deadzone and Input Constraints

Pingan He

Wenzhi Gao

Jagannathan Sarangapani
Missouri University of Science and Technology, sarangap@mst.edu

Recommended Citation

P. He et al., "Neuro Control of Nonlinear Discrete Time Systems with Deadzone and Input Constraints," *Proceedings of the 2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control*, Institute of Electrical and Electronics Engineers (IEEE), Jan 2006.
The definitive version is available at https://doi.org/10.1109/CACSD-CCA-ISIC.2006.4777088

Proceedings of the 2006 IEEE
International Conference on Control Applications
Munich, Germany, October 4-6, 2006

Neuro Control of Nonlinear Discrete Time Systems with Deadzone and Input Constraints

Pingan He, Wenzhi Gao, and S. Jagannathan

Abstract— **A neural network (NN) controller in discrete time is designed to deliver a desired tracking performance for a class of uncertain nonlinear systems with unknown deadzones and magnitude constraints on the input. The NN controller consists of two NNs: the first NN for compensating the unknown deadzones; and the second NN for compensating the uncertain nonlinear system dynamics. The magnitude constraints on the input are modeled as saturation nonlinearities and they are dealt with in the Lyapunov-based controller design. The uniformly ultimate boundedness (UUB) of the closed-loop tracking errors and the neural network weights estimation errors is demonstrated via Lyapunov stability analysis.**

I. Introduction

UNKNOWN actuator deadzone compensation in continuous time is treated in the seminal work of [1] for a known nonlinear system. By contrast, in [2], compensation of non-symmetric deadzones in discrete time is considered for linear systems whereas in [3], a fuzzy logic compensator is proposed for constant non-symmetric input deadzones in discrete time. On the other hand, in [4], the effect of the deadzone nonlinearity is overcome by using a neural network (NN) compensator in continuous time.

In the above works [1-4], the unknown deadzone is compensated without using any constraints on the input magnitude. In fact, physical limitations dictate that hard limits be imposed on the input magnitude to avoid damage to or deterioration of the system. The actuator magnitude constraints manifest themselves as saturation nonlinearities. In this paper, we show how to consider both an unknown deadzone and the input constraints for an uncertain nonlinear system via a novel NN controller. The main contributions of the proposed work are:

1) The saturation nonlinearity is introduced in our controller design besides compensating for the input deadzone so that the magnitude constraints of the actuators can be modeled. This makes the Lyapunov analysis quite involved;
2) The general case of non-symmetric time-varying deadzones in discrete time is treated here compared to the case of compensating a constant deadzone nonlinearity in discrete time [2, 3] and in continuous time [1, 4]. The unknown deadzone is compensated using a NN;

In Section II, we present background information on a class of nonlinear systems and neural network approximation property. In Section III, an inverse NN deadzone compensation scheme is proposed and the throughput error is shown to approach zero when the NN reconstruction error and weights estimation errors converge to zero. Input saturation nonlinearity is also discussed in that section. Section IV presents the NN controller design by considering uncertain nonlinear system dynamics, the unknown input deadzones, and the input saturation. Section VI provides the conclusions.

II. Background

A. Nonlinear System Description

Consider the following nonlinear system, to be controlled, given in the following form

$$\begin{aligned} x_1(k+1) &= x_2(k),\\ x_2(k+1) &= x_3(k),\\ &\vdots\\ x_n(k+1) &= f(x(k)) + u(k) + d'(k), \end{aligned} \tag{1}$$

where $x(k)=[x_1^T(k),x_2^T(k),\ldots,x_n^T(k)]^T \in R^{nm}$ with each $x_i(k)\in R^m$, $i=1,\ldots,n$, $x_i(k)$ is the state at time instant k, $f(x(k))\in R^m$ is the unknown nonlinear dynamics of the system, $u(k)\in R^m$ is the control input and $d'(k)\in R^m$ is the unknown but bounded disturbance, whose bound is given by $\|d'(k)\|\le d'_m$.

Definition 1 (Tracking Errors): Given a desired trajectory, $x_d(k)\in R^m$, and its past values $x_d(k+i-n)\in R^m$, $i=1,\ldots,n$, the tracking errors are defined as

$$e_i(k) = x_i(k) - x_d(k+i-n)\,, \tag{2}$$

with each error $e_i(k)\in R^m$. Combining (1) and (2), the error system is given by

$$\begin{aligned} e_1(k+1) &= e_2(k),\\ e_2(k+1) &= e_3(k),\\ &\vdots\\ e_n(k+1) &= f(x(k)) - x_d(k+1) + u(k) + d'(k) \end{aligned} \tag{3}$$

P. He is with the MotoTron Corp., 505 Marion Rd, Oshkosh, WI 54901 USA (e-mail: evertuga@yahoo.com).

W. Gao is with the Dept. of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA 70803 USA (e-mail: wenzhigao@yahoo.com).

S. Jagannathan is with the Dept. of electrical and Computer Engineering, University of Missouri, Rolla, MO 65409 USA (e-mail: sarangap@umr.edu).

The control objective is to show that $e_i(k) \in R^m$, i=1,..,n, is bounded.

B. Approximation Property

For a suitable approximation of unknown nonlinear functions, several neural network architectures are available. In [6], it is shown that a continuous function $f(x(k)) \in C(S)$, within a compact subset, S of R^n, can be approximated using a single-layer NN

$$f(x(k)) = w^T \phi(v^T x(k)) + \varepsilon(x(k)), \tag{4}$$

where w and v are target weights of the hidden to the output and input to the hidden layers respectively, $\phi(v^T x(k))$ denotes the vector of activation functions at the instant k, and $\varepsilon(x(k))$ is the NN functional reconstruction error vector. The neural network output is defined as

$$\hat{f}(x(k)) = \hat{w}^T(k)\phi(v^T x(k)), \tag{5}$$

where $\hat{w}(k)$ is the actual weight matrix. For simplicity, $\phi(\hat{v}^T x(k))$ is denoted as $\phi(x(k))$.

The input to the hidden layer weights, v, are selected at random initially and will not be tuned. The output layer weights $\hat{w}(k)$ are tunable. It is demonstrated in [6] that, if the number of hidden layer nodes is sufficiently large, the norm of reconstruction error $\|\varepsilon(x(k))\|$ can be made arbitrarily small on the compact set so that the bound $\|\varepsilon(x(k))\| \le \varepsilon_m$ holds for all $x(k) \in S$.

III. Deadzone and Nonlinearities

A. Deadzone Nonlinearity

The time-varying deadzone nonlinearity is displayed in Fig. 1. If $\tau(k)$ and $q(k)$ are scalars, the time-varying deadzone nonlinearity is given by

$$q(k) = h(\tau(k)) = \begin{cases} f_1(\tau(k)) & \tau(k) > b_+(k) \\ 0 & -b_-(k) \le \tau(k) \le b_+(k) \\ f_2(\tau(k)) & \tau(k) \le -b_-(k) \end{cases}, \tag{6}$$

where $\tau(k)$ and $q(k)$ are the input and output of deadzone function, $b_+(k)$ and $b_-(k)$ are positive time-varying scalars, and $f_1(\tau(k))$, $f_2(\tau(k))$ are nonlinear functions.

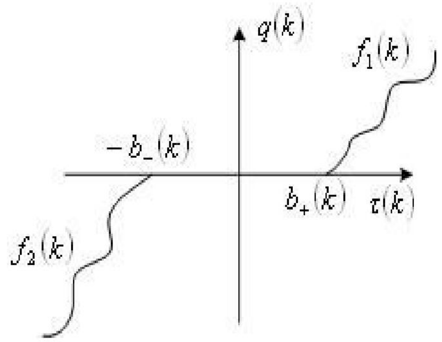

Fig. 1. Time-varying deadzone nonlinearity.

To compensate the deadzone nonlinearity, its inverse is required. Therefore, following assumption is needed for the inverse to exist.

Assumption 1: Both $f_1(\tau(k))$ and $f_2(\tau(k))$ are smooth, continuous and invertible functions.

Note: The above assumption implies that $f_1(\tau(k))$ and $f_2(\tau(k))$ are either increasing or decreasing nonlinear functions. In other words, $h(\tau(\mathrm{k}))$ is either non-decreasing or non-increasing function.

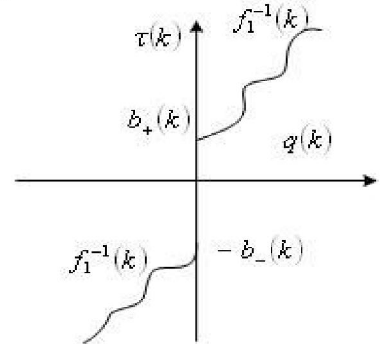

Fig. 2. Inverse of a deadzone nonlinearity.

With Assumption 1, the inverse time-varying deadzone function, $h^{-1}(q(k))$, is now given by

$$\tau(k) = h^{-1}(q(k)) = \begin{cases} f_1^{-1}(q(k)) & q(k) > 0 \\ 0 & q(k) = 0 \\ f_2^{-1}(q(k)) & q(k) < 0 \end{cases}. \tag{7}$$

where $f_1^{-1}(\cdot)$ and $f_2^{-1}(\cdot)$ are the inverse functions of $f_1(\cdot)$ and $f_2(\cdot)$ respectively. The inverse time-varying deadzone function is shown in Fig. 2.

B. Compensation of Deadzone Nonlinearity

To offset the deleterious effects of deadzones, a pre-compensator displayed in Fig. 3 is proposed [3]. The desired objective of the pre-compensator is to make the throughput from $p(k)$ to $q(k)$ equal to unity. Here, $p(k) \in R^m$, $\tau(k) \in R^m$ and $q(k) \in R^m$ are vectors.

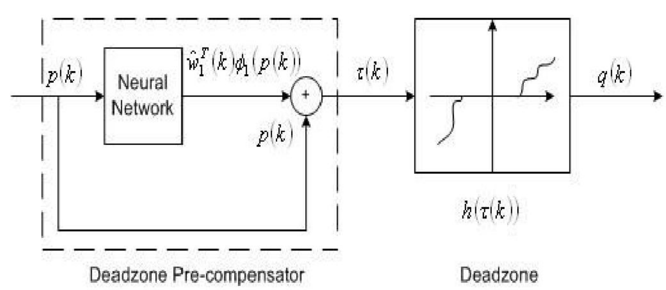

Fig. 3. Deadzone pre-compensator plus deadzone nonlinearity.

The pre-compensator consists of two parts [2]: a linear part, $p(k)$, designed to achieve the tracking of the reference signal, and a NN part, which is used to cancel the deadzone by approximating the nonlinear function $h^{-1}(p(k))-p(k)$. In other words,

$$\begin{aligned} & h^{-1}(p(k))-p(k) \\ & = w_1^T \phi_1(v_1^T p(k)) + \varepsilon_1(p(k)), \\ & = w_1^T \phi_1(p(k)) + \varepsilon_1(p(k)) \end{aligned} \tag{8}$$

where $w_1 \in R^{n_1 \times m}$ and $v_1 \in R^{m \times n_1}$ are the target weights, and $\varepsilon_1(k)$ is the NN reconstruction error, with n_1 the number of hidden layer nodes. For simplicity, hidden layer activation function $\phi_1(v_1^T p(k))$ is written as $\phi_1(p(k))$.

The actual NN output is defined as $\hat{w}_1^T(k)\phi_1(v_1^T p(k))$. For simplicity, it is expressed as $\hat{w}_1^T(k)\phi_1(p(k))$, with $\hat{w}_1(k) \in R^{n_1 \times m}$ being the actual output layer weights. A total of two single layer NNs will be used in the neural network controller design whereas the third one is only meant for the analysis of throughput error in Theorem 1 and it is not used in the NN controller design.

Definition 2: The weight estimation errors of all the NN are defined as

$$\tilde{w}_i(k) = \hat{w}_i(k) - w_i, \; i = 1, 2, 3. \tag{9}$$

Moreover, for convenience, define $\xi_i(k) \in R^m$ as

$$\xi_i(k) = \tilde{w}_i^T(k)\phi_i(k), \; i = 1, 2, 3. \tag{10}$$

where $\phi_i(k), i = 1,2,3$, are the hidden layer activation functions.

The deadzone function, $h(\cdot)$, defined in (6), is approximated by using a single layer NN as

$$h(k) = w_3^T \phi_3(v_3^T \tau(k)) + \varepsilon_3(\tau(k)), \tag{11}$$

where $w_3 \in R^{n_3 \times m}$ and $v_3 \in R^{m \times n_3}$ are the target weights, and $\varepsilon_3(\tau(k))$ is the NN reconstruction error with n_3 the number of hidden layer nodes. For simplicity, hidden layer activation function $\phi_3(v_3^T \tau(k))$ is written as $\phi_3(k)$.

To show that the effectiveness of the proposed deadzone pre-compensator, the following assumptions are required to proceed.

Assumption 2: The activation function $\phi_3(k)$ for the third NN is differentiable over a compact set S, and its derivative $\phi_3'(k)$ is bounded over the compact set S by $\|\phi_3'(k)\| \le \phi_{3m}'$, with $\phi_{3m}' \in R^+$.

Fact 1: The activation functions are bounded by known positive values so that

$$\|\phi_i(k)\| \le \phi_{im}, \quad i = 1,2,3, \tag{12}$$

where $\phi_{im} \in R^+, i = 1,2,3$ is the upper bound for $\phi_i(k), i = 1,2,3$.

Assumption 3 (Bounded Ideal Weights): The Frobenius norm [5] of the target weight matrix for all the NNs is bounded above by known positive values $w_{im} \in R^+, i = 1,2,3$ so that

$$\|w_i\| \le w_{im}, i = 1, 2, 3. \tag{13}$$

Assumption 4 (Bounded NN Reconstruction Errors): The NN reconstruction errors $\varepsilon_i(k), i = 1,2,3$ are bounded above over the compact set by $\varepsilon_{im} \in R^+, i = 1,2,3$ [5].

The next theorem shows that when the NN weight estimation and the reconstruction errors of the NN pre-compensator become zero, the throughput error, $q(k)-p(k)$ approaches zero.

Theorem 1 (Throughput Error): The throughput of the compensator plus the deadzone is given by

$$q(k) = p(k) + g(k)\xi_1(k) - g(k)\varepsilon_1(p(k)) + \varepsilon_3(\tau(k)) - \varepsilon_3(h^{-1}(p(k))), \tag{14}$$

with $\xi_1(k)$ defined in (10), $\varepsilon_1(p(k))$, $\varepsilon_3(\tau(k))$ and $\varepsilon_3(h^{-1}(p(k)))$ are the NN reconstruction errors, and the $g(k)$ is defined as

$$g(k) = w_3^T \phi_3'(\theta(k)), \tag{15}$$

where $\theta(k) \in R^m$ is a certain value between $\tau(k)$ and $h^{-1}(p(k))$. Given Assumptions (2) and (3), we can see that $g(k)$ bounded over the compact set S by $g_m \in R^+$

$$\|g(k)\| \le g_m, \tag{16}$$

and

$$g_m = w_{3m}\phi_{3m}', \tag{17}$$

where ϕ_{3m}' and w_{3m} are the upper bound for $\phi_3'(k)$ and w_3, and they are defined in Assumption 2 and 3 respectively.

From the Theorem 1, it can be concluded that when the NN weights estimation and the NN reconstruction error goes to zero, the throughput error, $q(k)-p(k)$, approaches to zero. This makes the deadzone pre-compensator plus the deadzone equal to unity. It also implies that the effect of deadzone is overcome by the proposed NN pre-compensator.

Proof: Proof is similar to that of Theorem 2. Due to space consideration, it is omitted. ■

C. Saturation Nonlinearity

The actuator limit can be modeled as saturation nonlinearity with limit defined as $u_{\max}$. Using Fig. 1, actuator constraint is expressed as

$$u(k) = \begin{cases} q(k) & \|u(k)\| \le u_{\max} \\ u_{\max}\,\mathrm{sgn}(q(k)) & \|u(k)\| > u_{\max} \end{cases}, \tag{18}$$

with $\mathrm{sgn}(\cdot)$ the sign function.

IV. NN Controller Design

The control objective is to make the system errors, $e_i(k)$, $i = 1,\dots n$, small with all the internal signals uniformly ultimately bounded (UUB). The proposed neural network controller consists of two NNs: one NN for deadzone compensating and the other for estimating unknown nonlinear dynamics plus tracking desired trajectory. The input constraints are considered in the Lyapunov-based controller design.

A. NN Controller Design

Case 1: $\|u(k)\| \le u_{\max}$.

Using (18), i.e., $u(k) = q(k)$ and combining with (3) and (14), we obtain

$$e_n(k+1) = f(x(k)) - x_d(k+1) + p(k) + g(k)(\xi_1(k) - \varepsilon_1(p(k))) + \varepsilon_3(\tau(k)) - \varepsilon_3(h^{-1}(p(k))) + d'(k) \quad (19)$$

A single-layer NN will be used to approximate the uncertain nonlinear dynamics, $f(x(k))$, as

$$f(x(k)) = w_2^T \phi_2(v_2^T x(k)) + \varepsilon_2(x(k)) = w_2^T \phi_2(k) + \varepsilon_2(x(k)), \quad (20)$$

and

$$\hat{f}(x(k)) = \hat{w}_2^T(k)\phi_2(v_2^T x(k)) = \hat{w}_2^T(k)\phi_2(k), \quad (21)$$

where $w_2 \in R^{n_2 \times m}$ and $v_2 \in R^{nm \times n_2}$ are the target weights, $\hat{w}_2(k) \in R^{n_2 \times m}$ is the actual weight matrix, with n_2 being the number of hidden layer nodes. For convenience, the hidden layer activation function $\phi_2(v_2^T x(k))$ is written as $\phi_2(k)$. Choose

$$p(k) = l e_n(k) - \hat{w}_2^T(k)\phi_2(k) + x_d(k+1), \quad (22)$$

with $l \in R^{m \times m}$ being the gain matrix. The error $e_n(k+1)$ is obtained using (19) – (22) as

$$e_n(k+1) = l e_n(k) + g(k)\xi_1(k) - \xi_2(k) + d_1(k), \quad (23)$$

where $\xi_2(k)$ is defined in (10), and $d_1(k)$ is defined as

$$d_1(k) = \varepsilon_2(x(k)) - g(k)\varepsilon_1(p(k)) + \varepsilon_4(\tau(k)) - \varepsilon_4(h^{-1}(p(k))) + d'(k). \quad (24)$$

Note in (24), $d_1(k)$ is bounded above by d_{1m} in the compact set S due to the fact that $\varepsilon_2(x(k))$, $g(k)$, $\varepsilon_4(\tau(k))$, $\varepsilon_4(h^{-1}(p(k)))$, and $d'(k)$ are bounded. The error system (3) for the Case I can be rewritten as

$$\begin{aligned} e_1(k+1) &= e_2(k) \\ e_2(k+1) &= e_3(k) \\ &\vdots \\ e_n(k+1) &= l e_n(k) + g(k)\xi_1(k) - \xi_2(k) + d_1(k) \end{aligned} \quad (25)$$

Case 2: $\|u(k)\| > u_{\max}$.

In this case, $p(k)$ is still defined as (22). Using (18), taking $u(k) = u_{\max}\operatorname{sgn}(q(k))$ into (3) gives

$$e_n(k+1) = f(x(k)) - x_d(k+1) + u_{\max}\operatorname{sgn}(q(k)) + d'(k). \quad (26)$$

Combining (20) and (26) gives

$$e_n(k+1) = w_2^T \phi_2(v_2^T x(k)) + \varepsilon_2(x(k)) - x_d(k+1) + u_{\max}\operatorname{sgn}(q(k)) + d'(k) \quad (27)$$

Let us denote

$$d_2(k) = \varepsilon_2(x(k)) - x_d(k+1) + u_{\max}\operatorname{sgn}(q(k)) + d'(k). \quad (28)$$

Equation (27) is simplified as

$$e_n(k+1) = w_2^T \phi_2(v_2^T x(k)) + d_2(k). \quad (29)$$

where the term $d_2(k)$ is bounded by d_{2m} over the compact set S given the fact the boundedness of $\varepsilon_2(x(k))$, $x_d(k+1)$, $u_{\max}$, $\operatorname{sgn}(q(k))$ and $d'(k)$. Using (29), the error system (3) becomes

$$\begin{aligned} e_1(k+1) &= e_2(k) \\ e_2(k+1) &= e_3(k) \\ &\vdots \\ e_n(k+1) &= w_2^T \varphi(v_2^T x(k)) + d_2(k) \end{aligned} \quad (30)$$

The structure of the proposed NN controller is depicted in Fig. 4 where the controller structure is naturally derived from the analysis presented above. Note that when the two NNs are removed from the inner loop, the outer-loop controller becomes a proportional controller.

B. Main Result.

It is required to show that the tracking errors (25) or (30), the NN weights $\hat{w}_1(k)$ and $\hat{w}_2(k)$ are bounded. In the following theorem, discrete-time weight tuning algorithms are given, which guarantee that both the tracking error and the NN weight estimates are bounded.

Theorem 2: Consider the system given in (1), the input deadzones (6) and the input constraints (18), and let the Assumptions 1 through 4 hold. Let the NN reconstruction errors, $\varepsilon_i(\cdot)$, $i = 1,2,3$, disturbances $d'(k)$, the desired trajectory, $x_d(k)$, and its past values be bounded. Let the first NN weight tuning be given by

$$\hat{w}_1(k+1) = \hat{w}_1(k) - \alpha_1\phi_1(k)\left(\hat{w}_1^T(k)\phi_1(k) + A l e_1(k)\right)^T, \quad (31)$$

with the second NN weights tuning be provided by

$$\hat{w}_2(k+1) = \hat{w}_2(k) - \alpha_2\phi_2(k)\left(\hat{w}_2^T(k)\phi_2(k) + B l e_1(k)\right)^T, \quad (32)$$

where $\alpha_1 \in R^+$, $\alpha_2 \in R^+$, $l \in R^{m \times m}$, $A \in R^{m \times m}$ and $B \in R^{m \times m}$ are design parameters. Consider the deadzone compensator $\tau(k) = p(k) + \hat{w}_1^T(k)\phi_1(k)$ with $p(k)$ given by (22). The tracking errors in (3), the NN weights $\hat{w}_1(k)$ and $\hat{w}_2(k)$, are UUB provided the design parameters are selected as:

$$(1)\ 0 < \alpha_i\|\phi_i(k)\|^2 < 1,\ i = 1,2, \quad (33)$$

$$(2)\ |l_{\max}| < \min\left(\frac{1}{2\sqrt{2}g_m},\ 1\right), \quad (34)$$

$$(3)\ \|A\|^2 + \|B\|^2 < \frac{1}{6}, \quad (35)$$

where $l_{\max}$ is the maximum eigenvalue of the gain matrix l.

Proof: See the appendix. ■

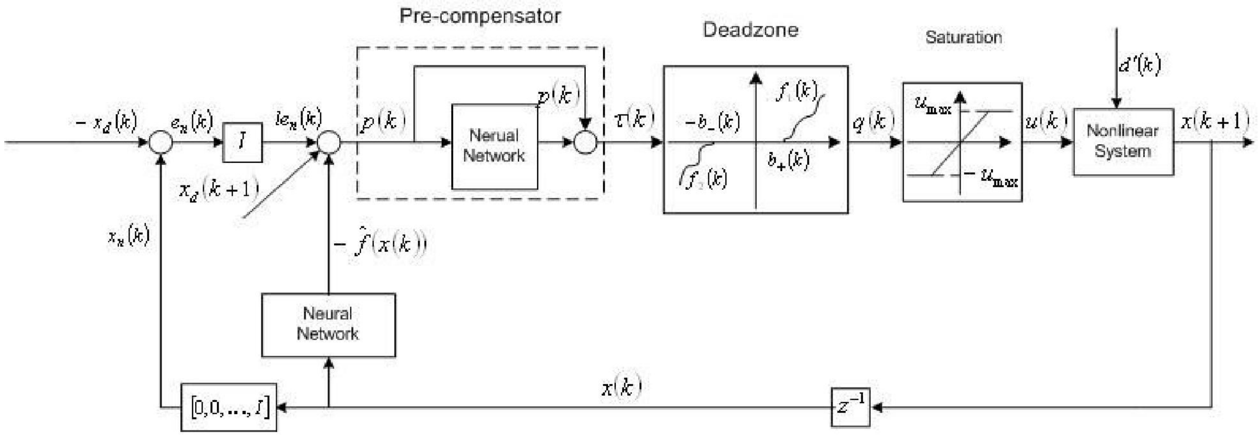

Fig. 4. NN controller with unknown deadzone and magnitude constraints.

V. Conclusion

Learning-based NN controller was developed for a class of uncertain discrete-time nonlinear systems with unknown deadzones with input magnitude constraints. The magnitude constraints are manifested in the controller design as saturation nonlinearities. The proposed neuro controller consisting of two NNs renders a satisfactory tracking performance. Lyapunov analysis ensures the boundedness of all the closed-loop signals in the presence of multiple nonlinearities.

Appendix

Proof of Theorem 2

Case 1: $\|u(k)\| \le u_{\max}$.

(1) $0 < g_m < 1$.

g_m is defined in Equations (17). Define the Lyapunov function candidate as

$$J(k) = J_1(k) + J_2(k) + J_3(k), \qquad \text{(A.1)}$$

where

$$J_1(k) = \frac{1}{16}\sum_{i=1}^{n} e_i^T(k)e_i(k) + \frac{1}{16}e_n^T(k)e_n(k), \qquad \text{(A.2)}$$

$$J_2(k) = \frac{1}{\alpha_1} tr(\widetilde{w}_1^T(k)\widetilde{w}_1(k)), \qquad \text{(A.3)}$$

$$J_3(k) = \frac{1}{\alpha_2} tr(\widetilde{w}_2^T(k)\widetilde{w}_2(k)), \qquad \text{(A.4)}$$

$\alpha_1 \in R^+$ and $\alpha_2 \in R^+$ are design parameters (see Theorem 2), $e_i(k), i = 1,\ldots,n$ are system tracking errors, $\widetilde{w}_i(k), i = 1,2$ are the weights estimation errors, which are defined in Equation (9), and $tr(\cdot)$ is the trace of a matrix.

The first difference of $J(k)$ is given by

$$\Delta J_1(k) \le -\frac{1}{16}\left(\|e_n(k)\|^2 + \|e_1(k)\|^2\right) + \frac{1}{2}\|le_n(k)\|^2 + \frac{1}{2}\left(\|g(k)\xi_1(k)\|^2 + \|\xi_2(k)\|^2 + \|d_1(k)\|^2\right). \qquad \text{(A.5)}$$

Since $\|g(k)\| \le g_m$ and $0 < g_m < 1$, we have $\|g(k)\| \le 1$. Combined with the fact that $l_{\max}$ is the maximum eigenvalue of the gain matrix l, (A.5) can be simplified as

$$\Delta J_1(k) \le \frac{1}{2}\left(l_{\max}^2 - \frac{1}{8}\right)\|e_n(k)\|^2 + \frac{1}{2}\|\xi_1(k)\|^2 + \frac{1}{2}\|\xi_2(k)\|^2 + \frac{1}{2}\|d_1(k)\|^2 - \frac{1}{16}\|e_1(k)\|^2, \qquad \text{(A.6)}$$

(2) $1 \le g_m$.

Define $J_1(k)$ as

$$J_1(k) = \frac{1}{16g_m^2}\sum_{i=1}^{n} e_i^T(k)e_i(k) + \frac{1}{16g_m^2}e_n^T(k)e_n(k), \qquad \text{(A.7)}$$

and the $J_2(k)$ and $J_3(k)$ the same as those of (A.3) and (A.4).

Similar to the procedure as the above, combining with (25), we have

$$\Delta J_1(k) \le \frac{1}{2}\left(l_{\max}^2 - \frac{1}{8g_m^2}\right)\|e_n(k)\|^2 + \frac{1}{2}\|\xi_1(k)\|^2 + \frac{1}{2}\|\xi_2(k)\|^2 + \frac{1}{2}\|d_1(k)\|^2 - \frac{1}{16g_m^2}\|e_1(k)\|^2. \qquad \text{(A.8)}$$

For $g_m \ge 1$, we have

$$-\frac{1}{16}\|e_n(k)\|^2 \le -\frac{1}{16g_m^2}\|e_n(k)\|^2, \qquad \text{(A.9)}$$

and

$$-\frac{1}{16}\|e_1(k)\|^2 \le -\frac{1}{16g_m^2}\|e_1(k)\|^2, \qquad \text{(A.10)}$$

Comparing (A.6) and (A.8), in both cases $0 < g_m \le 1$ and $1 \le g_m$, we have

$$\Delta J_1(k) \le \frac{1}{2}\left(l_{\max}^2 - \frac{1}{8g_m^2}\right)\|e_n(k)\|^2 + \frac{1}{2}\|\xi_1(k)\|^2 + \frac{1}{2}\|\xi_2(k)\|^2 + \frac{1}{2}\|d_1(k)\|^2 - \frac{1}{16g_m^2}\|e_1(k)\|^2 . \quad (A.11)$$

The next step is to get the first difference of $J_2(k)$ and $J_3(k)$. Based on the definition of (A.3), we have

$$\Delta J_2(k) = \frac{1}{\alpha_1} tr\left(\tilde{w}_1^T(k+1)w_1(k+1) - \tilde{w}_1^T(k)w_1(k)\right). \quad (A.12)$$

From the weights updating rule of (31), we obtain

$$\begin{aligned}\tilde{w}_1(k+1) &= \hat{w}_1(k+1) - w_1 \\ &= \tilde{w}_1(k) - \alpha_1\phi_1(k)\left(\tilde{w}_1^T(k)\phi_1(k) + w_1^T\phi_1(k) + Ale_1(k)\right)^T\end{aligned} . \quad (A.13)$$

Using $\xi_1(k) = \tilde{w}_1^T(k)\phi_1(k)$ in (10) to simplify (A.13):

$$\tilde{w}_1(k+1) = \tilde{w}_1(k) - \alpha_1\phi_1(k)\left(\tilde{w}_1^T(k)\phi_1(k) + w_1^T\phi_1(k) + Ale_1(k)\right)^T . \quad (A.14)$$

Simplifying (A.14) to get

$$\Delta J_2(k) \le -(1-\alpha_1\|\phi_1(k)\|^2)\left\|\xi_1(k) + (w_1^T\phi_1(k) + Ale_1(k))\right\|^2 + 3w_{1m}^2\phi_{1m}^2 - \|\xi_1(k)\|^2 + 3\|A\|^2 l_{\max}^2\|e_1(k)\|^2 , \quad (A.15)$$

where $\xi_1(k)$ is defined in (10), w_{1m} and ϕ_{1m} are upper bounds for ideal weight w_1 and activation function $\phi_1(k)$ respectively, and $l_{\max}$ is the maximum eigenvalue of the gain matrix l.

Similar to the above procedure, we can get $\Delta J_3(k)$ as

$$\Delta J_3(k) \le -(1-\alpha_2\|\phi_2(k)\|^2)\left\|\xi_2(k) + (w_2^T\phi_2(k) + Ble_1(k))\right\|^2 + 3w_{2m}^2\phi_{2m}^2 - \|\xi_2(k)\|^2 + 3\|B\|^2 l_{\max}^2\|e_1(k)\|^2 . \quad (A.16)$$

Combining (A.11), (A.15) with (A.16), the first difference of Lyapunov function $\Delta J(k)$ can be written as

$$\begin{aligned}\Delta J_1(k) \le\ & \frac{1}{2}\left(l_{\max}^2 - \frac{1}{8g_m^2}\right)\|e_n(k)\|^2 - \frac{1}{2}\|\xi_1(k)\|^2 - \frac{1}{2}\|\xi_2(k)\|^2 \\ & + \frac{1}{2}\|d_1(k)\|^2 + 3w_{1m}^2\phi_{1m}^2 + 3w_{2m}^2\phi_{2m}^2 + \\ & \left(3\|A\|^2 l_{\max}^2 + 3\|B\|^2 l_{\max}^2 - \frac{1}{16g_m^2}\right)\|e_1(k)\|^2 \\ & - (1-\alpha_1\|\phi_1(k)\|^2)\left\|\xi_1(k) + (w_1^T\phi_1(k) + Ale_1(k))\right\|^2 \\ & - (1-\alpha_2\|\phi_2(k)\|^2)\left\|\xi_2(k) + (w_2^T\phi_2(k) + Ble_1(k))\right\|^2 .\end{aligned} \quad (A.17)$$

This implies that $\Delta J(k) \le 0$ as long as (33) through (35) hold and

$$\|e_n(k)\| > \frac{\sqrt{2}D_{1M}}{\sqrt{\frac{1}{8g_m^2} - l_{\max}^2}} , \quad (A.18)$$

or

$$\|\xi_1(k)\| > \sqrt{2}D_{1M} , \text{ or } \|\xi_2(k)\| > \sqrt{2}D_{1M} , \quad (A.19)$$

where $D_{1M}^2 = \frac{1}{2}d_{1m}^2 + 3w_{1m}^2\phi_{1m}^2 + 3w_{2m}^2\phi_{2m}^2$.

Case 2: $\|u(k)\| > u_{\max}$.

The proof is similar to that of Case 1, and same conclusion can be derived that $\Delta J(k) \le 0$.

In both Case 1 and Case 2, $\Delta J(k) \le 0$ for all k greater than zero. According to the standard Lyapunov extension theorem [5], this demonstrates that $e_n(k)$ and the weight estimation errors are UUB. The boundedness of $e_n(k)$ implies that all the tracking errors are bounded from the error system (3). The boundedness of $\|\zeta_1(k)\|$ and $\|\zeta_2(k)\|$ implies that $\|\tilde{w}_1(k)\|$ and $\|\tilde{w}_2(k)\|$ are bounded, and this further implies that the weight estimates $\hat{w}_1(k)$ and $\hat{w}_2(k)$ are bounded. Therefore all the signals in the closed-loop system are bounded.

References

[1] D. A. Recker, P. V. Kokotovic, D. Rhode, and J. Winkelman, "Adaptive nonlinear controlof systems containing a deadzone," *Proc. IEEE Conf. Decis. Contr.*, pp. 2111-2115, 1991.

[2] G. Tao and P. V. Kokotovic, "Discrete-time adaptive control of systems with unknown deadzones," *Int. J. Contr.*, vol. 61, pp. 1-17, 1995.

[3] J. Campos, and F.L. Lewis, "Deadzone compensation in discrete-time using adaptive fuzzy logic", *IEEE Trans. Fuzzy Systems*, vol. 7, no. 6, pp. 697-707, 1999.

[4] Wenzhi Gao and R. Selmic, "Neural network control of a class of nonlinear systems with actuator saturation," *IEEE Trans. Neural Networks*, vol. 17, pp. 147 – 156, 2006.

[5] F. L. Lewis, S. Jagannathan, and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*, Taylor & Francis, PA, 1999.

[6] B. Igelnik and Y. H. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," *IEEE Trans. Neural Networks*, vol. 6, 1995.

[7] K. J. Astrom and B. Wittenmark, *Adaptive Control*, Addison Wesley, Reading, Massachusetts, 1995.