International Journal of Innovative
Computing, Information and Control
Volume **9**, Number **7**, July **2013**

ICIC International ©2013 ISSN 1349-4198
pp. 2839–2860

ADAPTIVE ROBUST DYNAMIC SURFACE CONTROL OF PURE-FEEDBACK SYSTEMS USING SELF-CONSTRUCTING NEURAL NETWORKS

Peng Li[1,2], Jie Chen[1,2], Tao Cai[1,2] and Guanghui Wang[1,2]

[1]School of Automation
Beijing Institute of Technology
No. 5, Zhongguancun South Street, Haidian District, Beijing 100081, P. R. China
dolimenster@gmail.com

[2]Key Laboratory of Complex System Intelligent Control and Decision
Ministry of Education
Beijing 100081, P. R. China

Received May 2012; revised September 2012

Abstract. *This paper proposes an adaptive robust dynamic surface control (ARDSC) method integrated a novel self-constructing neural network (SCNN) for a class of complete non-affine pure-feedback systems with disturbances. By employing the mean-value theorem and implicit function theorem, the adaptive robust control (ARC) method is extended to pure-feedback systems, and improves the robustness and transient performance of the closed-loop system. The "explosion of complexity" in backstepping scheme is avoided via dynamic surface control (DSC) technique. Moreover, the controller complexity is further reduced by introducing an SCNN based on a novel pruning strategy and a width adjustment strategy. Input-to-state stability and small-gain theorem are utilized to analyze the stability of the closed-loop system. At the end, simulation results demonstrate effectiveness and advantages of the proposed control method.*

Keywords: Adaptive robust control, Non-affine nonlinearity, Pure-feedback systems, Self-constructing neural networks, Dynamic surface control, Input-to-state stability, Small-gain theorem

1. Introduction.

In the past decades, with the help of systematic backstepping technique, strict or semi-strict feedback systems with parameter uncertainties or nonlinear uncertainties have received much attention, and various outstanding achievements have been obtained [1-9]. However, due to non-affine appearance of the control input, a relatively small number of results are available for control of complete non-affine pure-feedback systems which generally can be described as follows [6]

$$\begin{cases} \dot{x}_i = f_i(\bar{x}_i, x_{i+1}) + p_i(\bar{x}_n, t) \quad i = 1, \cdots, n-1 \\ \dot{x}_n = f_n(\bar{x}_n, u) + p_n(\bar{x}_n, t) \\ y = x_1, \end{cases} \tag{1}$$

where $\bar{x}_i = [x_1, x_2, \cdots, x_i]^T \in R^i$ with $i = 1, \cdots, n$ are state vectors, $y \in R$ and $u \in R$ are system output and input respectively, $f_i(\bar{x}_i, x_{i+1})$ and $f_n(\bar{x}_n, u)$ are unknown smooth nonlinear functions, and $p_i(\bar{x}_n, t)$ $i = 1, \cdots, n$ are unknown disturbances.

A mass of control systems with lower-triangular nonlinear structure can be represented or transformed into the complete pure-feedback form, including strict and semi-strict feedback systems as special cases. Practical instances can be found in aircraft flight control systems, mechanical systems and biochemical processes [10]. Air delivery subsystem for

PEM (proton exchange membrane) fuel cell is a recent practical example of non-affine pure-feedback systems [11].

When dealing with complete non-affine pure-feedback systems, a tremendous challenge arises from circular construction of the controller, caused by non-affine nonlinearity of the system with respect to virtual controls and practical control signals [10,15]. In order to deal with this problem, by employing the implicit function theorem and mean-value theorem, Wang et al. explicitly constructed desired virtual and practical controls without including themselves as the component [10]. Then, the input-to-state stability (ISS) analysis and small-gain theorem, were employed to show the stability of the closed-loop system. Based on the same methodology, a class of perturbed pure-feedback systems with unknown dead-zone were investigated in [16]. In [17], non-affine functions were expanded at a fixed point by exploiting the mean-value theorem, and then novel Lyapunov-Krasovskii functions were employed to perform the stability analysis. In [18], adaptive neural control for the pure-feedback system with generalized hysteresis input was investigated, and the unknown virtual control direction problem was solved using the Nussbaum function. Robust stabilization problem for a class of non-affine pure-feedback systems with unknown time-delay functions and perturbed uncertainties was investigated in [19], and novel continuous packaged functions were introduced in advance to remove unknown nonlinear terms deduced from perturbed uncertainties and unknown time-delay functions.

For the complete non-affine pure-feedback systems, the following problems should be further taken into account, which are the motivations of this paper:

1) For pure-feedback systems, the transient performance and robustness have received less attention than strict or semi-strict feedback systems. In [20], reconstruction errors were suppressed by a smooth robust term. Zhao and Lin utilized an initialization technique to improve $\mathcal{L}_\infty$ tracking performance [21]. However, how to guarantee transient performance and system stability under reconstruction errors, parameter adaptation errors and disturbance needs further consideration.
2) The "explosion of complexity" is a significant drawback of the backstepping design. The complexity mainly comes from the derivative of virtual controls and the large scale of neural networks at each backstepping design step. Therefore, how to simplify controller design is another motivation of this paper.

Pertaining to the issue of improving robustness, an adaptive robust control algorithm proposed in [22-24] can deal with reconstruction errors and parameter adaptation errors simultaneously by combining deterministic robust control with adaptive control effectively, and meanwhile, endow the closed-loop system with prescribed transient performance. By employing a discontinuous projection operator, the ARC guarantees the boundedness of parameter estimation and adaptive control term, as well as the existence of a robust compensator. This control method has been widely used in practical systems, and various improved ARCs have also been proposed [25-29]. With the goal of improving the robustness of the closed-loop system, this paper extended the ARC algorithm to the control of non-affine pure-feedback systems.

To reduce the complexity caused by derivative of virtual controls, the DSC technique, which introduces a first-order filter to replace the differential operation, was proposed for a class of strict-feedback nonlinear systems [25,29-31]. Zhang and Ge developed an adaptive DSC for a class of simple pure-feedback nonlinear systems with practical control signal in affine form [14]. In [15,21], DSC design was performed by constructing an affine state variable. However, the condition of implicit function theorem has not been rigorously verified. In [47], DSC technique was also adopted for control of pure-feedback nonlinear systems and just one adaptive parameter was updated online, which reduced computation load at the expense of sacrificing transient performance.

The complexity of neural network based controller is determined by the number of neurons to some extent. For the sake of further reducing the complexity without deteriorating control performance, different self-constructing algorithms have been proposed to shrink the scale of neural networks. In literature, various self-constructing approximators have been developed based on different generating and pruning strategies [32-39]. However, self-constructing approximators proposed in [32-39] either employed estimation-based parameter adaptive laws which did not guarantee system's stability [33-36], or applied extremely complex generating/pruning strategies which consumed lots of hardware resources [34,35,37,38]. To solve these problems, pragmatic self-organizing fuzzy neural networks were investigated [39-42], where a simple distance-based criterion and a "weight variation" based criterion were utilized as the generating strategy and the pruning strategy respectively. However, in [40-42], Taylor expansion was used to deduce an adaptive law for the nonlinear-in-the-parameter approximator, which increased approximation complexity and computation load.

In this paper, we consider the disturbed complete non-affine pure-feedback system, and try to give a solution for the above two problems. The main contributions are as follows.

1) The adaptive robust control (ARC) algorithm is extended to the control of non-affine pure-feedback system. ARC can endow the closed-loop system with guaranteed system stability and output transient performance, even switching off the adaptive term. This feature implies that the closed-loop system has robust output tracking performance.
2) The complexity of the controller for complete pure-feedback systems is significantly reduced by the DSC technique and SCNN. Unlike [15,21,47], we utilize DSC technique directly for control of pure-feedback nonlinear systems. At the same time, a novel SCNN is proposed to simplify controller design and decrease the scale of neural networks. Relying on these two techniques, the controller is simplified, and transient performance is improved as much as possible.
3) A novel pruning strategy and a width adjustment strategy are proposed for SCNN structure learning. The pruning strategy prevents neurons from being pruned blindly by taking past behavior of the closed-loop system into account, which is different from [39-42]. Furthermore, based on tracking performance, the width adjustment strategy not only regulates the resolution of neural networks automatically with less knowledge of the plant, but also avoids Taylor expansion.

The remainder of this paper is organized as follows. In Section 2, the problem and some preliminaries are stated. A novel self-constructing neural network (SCNN) and its parameter adaptive law are introduced in Section 3. How SCNN can serve as an approximator in control system is also explained in Section 3. Section 4 shows the design procedure of the controller in detail. The analysis of stability and transient performance of the closed-loop system are given in Section 5. Section 6 shows simulations. Conclusions are made in Section 7.

2. **Problem Formulation and Preliminaries.** Throughout this paper, $\hat{*}$ denotes the estimate of $*$, and $\tilde{*} = * - \hat{*}$. $\| * \|$ represents the Euclidean norm of $*$, I_n denotes the identity matrix of size n, $\varnothing$ denotes the empty set, and $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ denote the smallest and largest eigenvalues of a square matrix, respectively.

2.1. **Problem formulation.** The objective is to design a controller u for system (1), such that all the signals in the closed-loop system are bounded and the output y tracks the desired output trajectory y_r as closely as possible.

To facilitate the controller design, the following assumptions and lemmas are needed.

Assumption 2.1. *The desired reference signal y_r and its ith ($i = 1, 2$) derivatives are continuous and available, and $[y_r, \dot{y}_r, \ddot{y}_r]^T \in \Omega_r$, where Ω_r is a known compact set and defined as $\Omega_r = \left\{[y_r, \dot{y}_r, \ddot{y}_r]^T | y_r^2 + \dot{y}_r^2 + \ddot{y}_r^2 \le B_r\right\} \subset R^3$, whose size B_r is a known positive constant.*

Lemma 2.1 (Implicit Function Theorem [12])**.** *If $\forall(x, y) \in R^n \times R$, $f(x, y) : R^n \times R \to R$ is continuously differentiable, and $\forall(x, y) \in R^n \times R$, $\exists\beta$ s.t. $\partial f(x, y)/\partial y \ge \beta > 0$, then there exists a continuous function $y^* = y(x)$ such that $f(x, y^*) = r$, $r \in R$.*

To simplify the representation, let x_{n+1} denote u. According to the mean-value theorem [43], $f_i(\bar{x}_i, x_{i+1})$ can be expressed as:

$$f_i(\bar{x}_i, x_{i+1}) = f_i(\bar{x}_i, x_{i+1}^0) + \frac{\partial f_i(\bar{x}_i, x_{i+1})}{\partial x_{i+1}}\Big|_{x_{i+1}=x_{i+1}^{\rho_i}} \times (x_{i+1} - x_{i+1}^0),\ i = 1, \cdots, n \tag{2}$$

where $x_{i+1}^{\rho_i} = x_{i+1}^0 + \rho_i(x_{i+1} - x_{i+1}^0)$ with $0 < \rho_i < 1$. Define

$$h_i(\bar{x}_i, x_{i+1}) = \frac{\partial f_i(\bar{x}_i, x_{i+1})}{\partial x_{i+1}},\ i = 1, \cdots, n \tag{3}$$

where $h_i(\bar{x}_i, x_{i+1})$ $i = 1, \cdots, n$ are unknown nonlinear functions.

Assumption 2.2. *The signs of $h_i(\cdot, \cdot)$, $(i = 1, \cdots, n)$ are known and fixed. In addition, there exist constants $0 < h_i^l < h_i^u < \infty$ such that $|h_i(\bar{x}_i, x_{i+1})| > h_i^l$, $\forall(\bar{x}_i, x_{i+1}) \in R^i \times R$ and $|h_i(\bar{x}_i, x_{i+1})| < h_i^u$, $\forall(\bar{x}_i, x_{i+1}) \in \Omega_i$, where $\Omega_i \subset R^{i+1}$ are compact sets.*

Without loss of generality, assume that $h_i(\bar{x}_i, x_{i+1}) > 0$.

Assumption 2.3. *The unknown uncertain disturbances $p_i(\bar{x}_n, t)$ are assumed to be bounded by*

$$|p_i(\bar{x}_n, t)| \le p_i^*,\ \forall(\bar{x}_n, t) \in R^n \times R_+ \ \ i = 1, \cdots, n \tag{4}$$

where p_i^* are known positive constants.

2.2. **Discontinuous projection operator.** To prevent parameter estimates from drifting unboundedly, the projection method is introduced herein to retain parameter estimates within a pre-specified compact and convex set Ω_Θ. In order to simplify the calculation, the discontinuous projection operator [24,29] is used

$$\mathrm{Proj}_{\hat{\Theta}}(\bullet) = \begin{cases} 0 & \text{if } \hat{\theta}_i = \theta_{i(\min)} \text{ and } \bullet_i < 0 \\ & \text{or } \hat{\theta}_i = \theta_{i(\max)} \text{ and } \bullet_i > 0 \\ \bullet_i & \text{otherwise} \end{cases} \tag{5}$$

where $\hat{\theta}_i$ denotes the ith element of parameter estimates $\hat{\Theta}$, $\theta_{i(\min)}$ and $\theta_{i(\max)}$ represent the minimum and maximum values of $\hat{\theta}_i$, and $\bullet$ represents any reasonable adaptation functions such as $\Gamma\Upsilon$, where Γ is the gain of the adaptive law and Υ is the adaptive function.

It is notable that the discontinuous projection operator has the following properties, making it suitable for designing an adaptive robust controller.

Property 2.1 The projection operator guarantees:

$$(i)\ \hat{\Theta} \in \Omega_\Theta = \{\Theta : \Theta_{\min} \le \Theta \le \Theta_{\max}\} \tag{6}$$

$$(ii)\ \forall\Upsilon\ \ \tilde{\Theta}^T[\Gamma^{-1}\mathrm{Proj}_{\hat{\Theta}}(\Gamma\Upsilon) - \Upsilon] \ge 0 \tag{7}$$

3. **Self-constructing Neural Networks.**

This section introduces SCNN and a novel structure learning algorithm, and gives an SCNN-based approximator for controller design.

3.1. **Radial basis function neural networks.**

The standard radial basis function neural network (RBFNN) is briefly described, and some assumptions are made.

According to the universal approximation theorem, RBFNN is able to approximate any continuous function on a compact set as accurately as possible with a sufficient number of neurons. On a compact set Ω_P, any continuous function $f(P)$ can be estimated as

$$f(P) = g_N(P, \Theta^*) + \epsilon(P), \ \forall P \in \Omega_P$$

where $\epsilon(P)$ is the residual approximation error, and

$$g_N(P, \Theta^*) = \Theta^{*T}\Psi(P) \tag{8}$$

represents an RBFNN approximator with l $(l > 1)$ neurons. $P \in \Omega_P \subset R^m$ is the input vector, and $\Psi(P) = [\psi_1(P), \cdots, \psi_l(P)]^T$ is the vector of radial basis function. In this paper, we use Gaussian function as the radial basis function

$$\psi_i(P) = \exp\left(\frac{-(P - c_i)^T(P - c_i)}{\sigma_i^2}\right), \ \ i = 1, \cdots, l$$

where c_i and σ_i are the center and the width of neuron i, respectively. $\Theta^* = [\theta_1^*, \cdots, \theta_l^*]^T$ is the optimal weight vector which satisfies

$$\Theta^* = \arg\min_{\Theta}\left(\sup_{P\in\Omega_P} |g_N(P, \Theta) - f(P)|\right).$$

Assumption 3.1. *On the compact set Ω_P, the bounded optimal weight vector Θ^* exists and makes $|\epsilon(P)| \le \epsilon^*$ for all $P \in \Omega_P$, where ϵ^* is a positive constant. In addition, $\Theta^* \in \Omega_\Theta$ with $\Omega_\Theta = \left\{\Theta \in R^l | \Theta_{\min} \le \Theta \le \Theta_{\max}\right\}$, where $\Theta_{\min}$ and $\Theta_{\max}$ are known vectors.*

Usually, Θ^* is estimated by the adaptive law. Therefore, for the unknown function $f(P)$, we have

$$f(P) = \hat{\Theta}^T\Psi(P) + \hat{\epsilon}(P) \tag{9}$$

where $\hat{\Theta}$ is the estimate of Θ^* and $\hat{\epsilon}(P) = \tilde{\Theta}^T\Psi(P) + \epsilon(P)$.

If we know the bound of the function $f(P)$, it is easy to obtain the compact set Ω_Θ. Usually, in practical application, a sufficiently large Ω_Θ can be adopted to make sure that Θ^* is included in Ω_Θ. Moreover, in terms of Assumption 3.1 and Property 2.1, Θ^*, $\hat{\Theta}$ and $\epsilon(P)$ are bounded; hence, $\hat{\epsilon}(P)$ is also bounded. Note that the boundedness of $\hat{\epsilon}(P)$ does not require that Θ^* is included in Ω_Θ.

Lemma 3.1 ([44]). *Consider the Gaussian RBF network (8) and define $c_{\min} := \frac{1}{2}\min_{i\neq j}$ $\|c_i - c_j\|$. Then, there is an upper bound of $\|\Psi(P)\|$:*

$$\|\Psi(P)\| \le \sum_{k=0}^{\infty} 3m(k+2)^{m-1}e^{-2c_{\min}^2 k^2/\sigma_i^2} := \psi^*$$

where m denotes the dimension of input P, σ_i denotes the minimal width of the Gaussian function, and ψ^* is a positive constant.

Remark 3.1. *The inequality can be proven through the ratio test theorem, and it is notable that provided $c_{\min} \neq 0$, ψ^* always exists.*

3.2. **Structure learning algorithm.** Compared with the standard RBFNN, SCNN can adjust their structure and parameters simultaneously. For the structure learning phase, the number of neurons varies over time, and neurons will be generated or pruned based on new observations. For the parameter learning phase, weight vectors are updated by the adaptive law derived from the Lyapunov function.

Through neuron generating, neuron pruning and width adjustment, the structure learning algorithm can automatically determine the number and the locations of neurons based on the information contained in sequently received observations. To describe this algorithm in detail, we define the normalized distance between input vector x and neuron center c as

$$D_{x,c} = \|x - c\| = \left(\frac{(x-c)^T(x-c)}{\sigma_c^2} \right)^{\frac{1}{2}} \tag{10}$$

where σ_c is the width of the Gaussian function associated with center c. s_i represents the output tracking error of the ith state in each backstepping step.

Neuron Generating: We firstly consider the general situation in which there already exist l ($l \geq 1$) neurons in SCNN.

Condition 3.1 *When any new input P is received, if D_{P,c_i} satisfies*

$$D_{P,c_i} \geq d_g \sigma_i, \ \forall 1 \leq i \leq l$$

where $d_g > 0$ is the preset neuron generating coefficient and σ_i is the width of the ith existing neuron, then a new neuron should be generated and added into the network.

The generated neuron is chosen as:

$$c_{l+1} = P, \ \sigma_{l+1} = \sigma(t), \ \theta_{l+1} = 0 \tag{11}$$

where c_{l+1} is the center of the new neuron, σ_{l+1} is the width of the Gaussian function, $\sigma(t)$ is determined by the width adjustment strategy, and θ_{l+1} is the weight associated with the new neuron.

When the first input P_1 is received, the first neuron is just located at P_1 with width σ^0. σ^0 represents the preset initial width of the Gaussian function which is also the maximum width.

Note that, due to the Gaussian function is the local supported function, Condition 3.1 means that the previous network cannot cover the new input P sufficiently. Therefore, when Condition 3.1 is satisfied, a new neuron located at P should be added into the network.

Neuron Pruning: Large-scale neural network will result in a very complex controller and large computation load. Therefore, neuron pruning strategy should be adopted to insure the scale of the neural network acceptable. Inspired by [40], a novel neuron pruning strategy is introduced herein. Firstly of all, we introduce some indexes used in the pruning criterion. For any new observation P, significance index SI_i and significance duration index DI_i of the ith existing neuron are denoted as

$$SI_i = \frac{|\theta_i \psi_i(P)|}{\sum\limits_{m=1}^{l} |\theta_m \psi_m(P)|}, \ DI_i : \begin{cases} DI_i(t) = 1, & t_{u_j} \leq t \leq t_{d_j} \\ DI_i(t) = \exp(-\tau_p(t - t_{d_j})), & t_{d_j} < t < t_{u_{j+1}} \end{cases} \tag{12}$$

where $\tau_p > 0$ is an adjustable parameter, and $0 < t_{u_j} < t_{d_j} < t_{u_{j+1}}$, $j = 0, 1, 2, \ldots$, t_{u_0} is the moment of the ith neuron created. Moreover, $D_{P,c_i} \leq d_p \sigma_i$ in $[t_{u_j}, t_{d_j}]$, and $D_{P,c_i} > d_p \sigma_i$ in $(t_{d_j}, t_{u_{j+1}})$, where d_p represents the preset neuron pruning coefficient.

Condition 3.2 *Based on above definitions, neurons can be pruned when the following* *conditions are satisfied:*

$$\begin{array}{ll} (i) & |s_i| \le \epsilon_p \\ (ii) & J := \{j \in L | D_{P,c_j} > d_p\sigma_j, DI_j < DI_p, SI_j < S_p\} \neq \varnothing \\ & L = \{1, 2, \cdots, l\} \end{array}$$

where DI_p and S_p are adjustable parameters.

With Condition 3.2, the pruning operation can be described as: if tracking error $s_i \le \epsilon_p$ and $J \neq \varnothing$, then neurons included in J can be removed from the network.

It is notable that the significance duration index DI_i decays over time when the input vector is far away from the center of the ith neuron. However, before the ith neuron is removed from the network, if the input vector returns to the ith neuron's effective range $d_p\sigma_j$, DI_i will be reset to 1. The effect of SI_i prevents the pruning operation from making significant variation of the output of SCNN.

Width Adjustment: As one of the parameters of the Gaussian function, width σ_i can be updated using adaptive law [40-42]. However, estimation of σ_i is a nonlinear optimization problem, and some linearization technique may be used, which makes adaptive law complex and less efficient. Therefore, in this paper, a simple width adjustment strategy is introduced to improve the performance of the SCNN without requiring complex computation.

It is well known that, the smaller width of the Gaussian function is, the higher resolution the neural network has. However, a smaller width will increase the curvature of the Gaussian function, and consequently, leads to a large-scale neural network. To maintain the neural network in a suitable scale, the width should be decreased just when the performance of the approximator is unacceptable.

To facilitate the statement, define time index ξ and accumulation error index e^{ξ} as:

$$\xi_i : \begin{cases} \xi_i(t_r) = 1, \ t_r, \ r = 0, 1, 2, \ldots \\ \xi_i(t) = \exp(-\tau_{\xi}(t - t_r)), \ t_r \le t < t_{r+1}, \ r = 0, 1, 2, \ldots \end{cases} \tag{13}$$

$$e_i^{\xi}(t) = \int_{t_r}^{t} |s_i(t)| dt, \ t_r < t \le t_{r+1}, \ r = 0, 1, 2, \ldots \tag{14}$$

where t_r is the moment such that $\xi_i(t) = \xi_w$ or a new neuron is added into the network. ξ_w is a positive constant.

Condition 3.3 *The conditions of decreasing the width of the new generated Gaussian function are*

$$\text{(i) } e_i^{\xi}(t_{r+1}) \ge \epsilon_{rw} \ \ \text{(ii) } e_i^{\xi}(t_r) - e_i^{\xi}(t_{r+1}) \le e_w \ \ \text{(iii) } \forall i, \ \ 1 \le i \le l, D_{P,c_i} < d_g\sigma_i,$$

where e_{rw} and e_w are positive constants.

The current width $\sigma(t)$ can be obtained by:

$$\sigma(t) = \frac{\sigma_0}{2^w} \tag{15}$$

where $w = 0, 1, 2, \ldots$ and σ_0 denotes the preset initial width. At the beginning $w = 0$, if Condition 3.3 is satisfied, w should be increased and $w_{new} = w_{old} + 1$.

On the contrary, the following condition indicates when w should be decreased.

Condition 3.4 *The condition of increasing the width of the new generated Gaussian function is:*

$$\begin{array}{l} \exists w', 0 \le w' < w, \\ \forall i \in \{n \in L | \sigma_n = \frac{\sigma_0}{2^{w'}}\}, \text{ such that } D_{P,c_i} \ge d_g\sigma_i, \end{array}$$

Consequently, based on Condition 3.4, if w' exists, we choose $w^{new} = \min\{w'\}$.

Remark 3.2. *On one hand, Condition 3.3 indicates that, between time interval* $[t_r, t_{r+1})$, *no neuron is generated although the output tracking error is still large and the performance of the neural network is ineffective. This situation indicates that the current resolution is too low to distinguish the variation of the unknown function. Therefore, a higher resolution is needed and the width should be decreased. To prevent neuron number from growing infinitely, an upper bound of* w *can be introduced. On the other hand, Condition 3.4 describes another situation in which the input vector* P *moves into the field which has not been explored sufficiently, so a lower resolution is suitable and the width should be increased.*

Remark 3.3. *Generally, how to choose a suitable value of the width of the Gaussian function needs extensive experience in controller design and sufficient information of the system. Therefore, the width adjustment strategy can assign suitable width automatically based on system behavior and the performance of neural networks.*

Remark 3.4. *Among the parameters of SCNN,* d_g, d_p, DI_p, ξ_w *and* e_w *have greater impact on neuron generating and pruning operations. The smaller* d_g *is, the more neurons will be generated. If large occurrence probability of pruning operation is expected, large* d_p *and small* DI_p *are needed. Moreover, large* e_w *and small* ξ_w *make the width tend to be smaller, which will increase the number of neurons. As a result, tradeoff between the scale of SCNN and the controller performance must be made.*

3.3. **Parameter adaptive law.** Besides the structure learning phase, the parameter learning phase is also necessary for approximating the unknown function. In this paper, the Lyapunov-based technique is used for the design of weight adaptive law, which guarantees the boundedness of weights and the stability of the closed-loop system. Using the discontinuous projection operator, the adaptive law for $\hat{\Theta}_i$ is chosen as

$$\dot{\hat{\Theta}}_i = \mathrm{Proj}[-\Gamma_i(\Psi_i(P_i)s_i + \eta_i\hat{\Theta}_i)],\ \ i = 1, \cdots, n \tag{16}$$

where $\Gamma_i = \Gamma_i^T > 0$ is the adaptive gain matrix, and $\eta_i > 0$ is a positive constant.

3.4. **Approximator based on self-constructing neural networks.** Let l_b denote the acceptable maximum number of neurons. Then, the neurons can be classified into two parts: the first part includes the neurons added into the neural network (activated neurons), and the other part contains the neurons not yet included in the neural network (unactivated neurons). Let $l(t)$ denote the number of neurons in the network at t, for an unknown continuous function $f(P)$, we have

$$f(P) = \Theta_a^{*T}\Psi_a(P) + \Theta_p^{*T}\Psi_p(P) + \epsilon(P)\ \ \forall P \in \Omega_P,$$

where $\Psi_a(P) = [\psi_1(P), \cdots, \psi_{l(t)}(P)]^T$ is the vector of activated neurons, $\Psi_p(P) = [\psi_{l(t)+1}(P), \cdots, \psi_{l_b}(P)]^T$ is the vector of unactivated neurons, $\Theta_a^* \in R^{l(t)\times 1}$ and $\Theta_p^* \in R^{(l_b-l(t))\times 1}$ are the weight vectors of activated neurons and unactivated neurons, respectively. Define

$$\begin{aligned} &\hat{\Theta}_c = [\hat{\Theta}_a^T, \mathbf{0}_{1\times(l_b-l(t))}]^T \\ &\Psi(P) = [\Psi_a^T(P), \Psi_p^T(P)]^T \\ &\Psi(P)_c = [\Psi_a^T(P), \mathbf{0}_{1\times(l_b-m(t))}] \end{aligned} \tag{17}$$

then

$$f(P) = \hat{\Theta}_c^T\Psi_c(P) + \tilde{\Theta}_c^T\Psi(P) + \epsilon(P)\ \ \forall P \in \Omega_P, \tag{18}$$

where $\tilde{\Theta}_c = \Theta^* - \hat{\Theta}_c$. Under Assumption 3.1, $\epsilon_c(P) = \tilde{\Theta}_c^T\Psi(P) + \epsilon(P)$ is bounded. Therefore, the approximator using SCNN is represented as

$$\hat{f}(P) = \hat{\Theta}_c^T\Psi_c(P), \tag{19}$$

and used in adaptive robust controller in the next section.

4. **Design of Adaptive Robust Controller with Dynamic Surface Control.** In this section, the design procedure of SCNN based ARDSC for the complete non-affine pure-feedback system is explained. The DSC technique is employed to overcome the "explosion of complexity" issue. In addition, SCNN is utilized as the adaptive term to further reduce controller complexity. In each step, the implicit function theorem and the mean-value theorem are used to transform the non-affine function into an affine form with respect to virtual controls and practical control. The detailed design procedure is shown as follows.

Step 1: Consider the first equation of system (1):

$$\dot{x}_1 = f_1(\bar{x}_1, x_2) + p_1(\bar{x}_n, t), \tag{20}$$

and define the first error surface $s_1 = x_1 - y_r$, let p_1 denote $p_1(\bar{x}_n, t)$, then the time derivative of s_1 is

$$\dot{s}_1 = f_1(\bar{x}_1, x_2) + p_1 - \dot{y}_r \tag{21}$$

From Assumption 2.2, it is known that $h_1(\bar{x}_1, x_2) > h_1^l > 0, \forall(\bar{x}_1, x_2) \in R^2$. In addition, since $\dot{y}_r$ is not a function of x_2, it is clear that $\partial[f_1(\bar{x}_1, x_2) - \dot{y}_r]/\partial x_2 > h_1^l > 0$. According to Lemma 2.1, there exists a continuous function $x_2^* = \alpha_1^0(\bar{x}_1, \dot{y}_r)$ such that $f_1(\bar{x}_1, x_2^*) - \dot{y}_r = 0$. Using (2), we have

$$\begin{aligned} \dot{s}_1 &= h_1(\bar{x}_1, x_2^{\rho_1})(x_2 - \alpha_1^0) + f_1(\bar{x}_1, \alpha_1^0) + p_1 - \dot{y}_r \\ &= h_1(\bar{x}_1, x_2^{\rho_1})(x_2 - \alpha_1^0) + p_1 \end{aligned} \tag{22}$$

where $x_2^{\rho_1} = \alpha_1^0 + \rho_1(x_2 - \alpha_1^0)$, $0 < \rho < 1$. An SCNN is utilized to approximate $\alpha_1^0(\bar{x}_1, \dot{y}_r)$ with $P_1 = [\bar{x}_1, \dot{y}_r] \in \Omega_{P_1} \subset R^2$, so $\alpha_1^0(\bar{x}_1, \dot{y}_r)$ can be rewritten as

$$\alpha_1^0(P_1) = \hat{\Theta}_{1c}^T \Psi_{1c}(P_1) + \tilde{\Theta}_{1c}^T \Psi_1(P_1) + \epsilon_1(P_1)$$

where $\hat{\Theta}_{1c} \in \Omega_{\Theta_{1c}} \subset R^{l_1 \times 1}$, $\tilde{\Theta}_{1c} = \Theta_1^* - \hat{\Theta}_{1c}$, $\Psi_{1c}(P_1) \in R^{l_1 \times 1}$ and l_1 is the maximum neuron number. According to Assumption 3.1, $|\epsilon_1(P_1)| \le \epsilon_1^*$ with positive constant ϵ_1^*. In order to synthesize a virtual control function α_1 for x_2 such that x_1 tracks the desired trajectory y_r as closely as possible, α_1 is designed as:

$$\begin{aligned} &\alpha_1 = \alpha_{1a} + \alpha_{1s}, \ \alpha_{1a} = \hat{\Theta}_{1c}^T \Psi_{1c}(P_1) \\ &\alpha_{1s} = \alpha_{1s1} + \alpha_{1s2}, \ \alpha_{1s1} = -k_1 s_1 \end{aligned} \tag{23}$$

where α_{1a} denotes the adaptive term to compensate α_1^0; α_{1s} is the robust control law and constructed by two parts: α_{1s1} is a proportional feedback to stabilize the nominal system; α_{1s2} is a robust term to compensate the modeling error and model uncertainties, and satisfies the following conditions:

$$\begin{aligned} &\text{i.} \quad s_1 h_1(\bar{x}_1, x_2^{\rho_1}) \left(\alpha_{1s2} - \tilde{\Theta}_{1c}^T \Psi_1(P_1) - \epsilon_1(P_1) + p_1^h \right) \le \delta_1 \\ &\text{ii.} \quad s_1 h_1(\bar{x}_1, x_2^{\rho_1}) \alpha_{1s2} \le 0 \end{aligned} \tag{24}$$

where δ_1 is a positive constant, and $p_1^h = \frac{p_1}{h_1(\bar{x}_1, x_2^{\rho_1})}$. α_{1s2} can be designed as

$$\alpha_{1s2} = -\frac{\pi_1 h_1^u}{2\delta_1} s_1, \quad \pi_1 \ge \|\Theta_{1\max} - \Theta_{1\min}\|_2^2 \|\Psi_1(P_1)\|_2^2 + \epsilon_1^{*2} + \frac{p_1^*}{h_1^l} \tag{25}$$

To avoid the requirement of $\dot{\alpha}_1$, the desired trajectory x_{2r} is produced from a first-order low-pass filter with α_1 being the input:

$$\nu_1 \dot{x}_{2r} + x_{2r} = \alpha_1, \quad x_{2r}(0) = \alpha_1(0). \tag{26}$$

From (22) and (26), it is clear that DSC technique is used directly for controller design by neither constructing affine state variables nor reducing adaptive parameters.

Define the second error surface as $s_2 = x_2 - x_{2r}$, and let $z_2 = x_{2r} - \alpha_1$. Then, the first error dynamic system can be represented as

$$E_1 : \dot{\lambda}_1 = A_1\lambda_1 + B_1\mu_1 + d_1 \tag{27}$$

where $\lambda_1 = [s_1, z_2]^T$, $\mu_1 = [s_2, \dot{\alpha}_1]^T$ and

$$A_1 = \begin{bmatrix} -k_1 h_1(\bar{x}_1, x_2^{\rho_1}) & h_1(\bar{x}_1, x_2^{\rho_1}) \\ 0 & -\frac{1}{\nu_1} \end{bmatrix}, \quad B_1 = \begin{bmatrix} h_1(\bar{x}_1, x_2^{\rho_1}) & 0 \\ 0 & -1 \end{bmatrix}$$

$$d_1 = \begin{bmatrix} h_1(\bar{x}_1, x_2^{\rho_1})(\alpha_{1s2} - \tilde{\Theta}_{1c}^T\Psi_1(P_1) - \epsilon_1(P_1) + p_1^h) \\ 0 \end{bmatrix}.$$

Choose the Lyapunov function candidate $V_1 = \frac{1}{2}\lambda_1^T\lambda_1$. Taking the derivative of V_1 along (27) yields

$$\begin{aligned} \dot{V}_1 &= \lambda^T(A_1\lambda_1 + B_1\mu_1 + d_1) \\ &= -k_1 h_1(\bar{x}_1, x_2^{\rho_1})s_1^2 + s_1 h_1(\bar{x}_1, x_2^{\rho_1})(s_2 + z_2) + z_2(-\frac{1}{\nu_1}z_2 - \dot{\alpha}_1) \\ &\quad + s_1 h_1(\bar{x}_1, x_2^{\rho_1})(\alpha_{1s2} - \tilde{\Theta}_{1c}^T\Psi_1(P_1) - \epsilon_1(P_1) + p_1^h) \end{aligned} \tag{28}$$

According to the Young's inequality and (24), we have

$$\begin{aligned} \dot{V}_1 \le &- k_1 h_1(\bar{x}_1, x_2^{\rho_1})s_1^2 + 2h_1(\bar{x}_1, x_2^{\rho_1})s_1^2 + \frac{1}{4}h_1(\bar{x}_1, x_2^{\rho_1})s_2^2 \\ &+ \left(-\frac{1}{\nu_1} + 1 + \frac{1}{4}h_1(\bar{x}_1, x_2^{\rho_1})\right) z_2^2 + \frac{1}{4}|\dot{\alpha}_1|^2 + \delta_1 \\ \le &- (k_1 - 2)h_1(\bar{x}_1, x_2^{\rho_1})s_1^2 + \frac{1}{4}h_1(\bar{x}_1, x_2^{\rho_1})s_2^2 + \delta_1 \\ &- \left(\frac{1}{\nu_1} - 1 - \frac{1}{4}h_1(\bar{x}_1, x_2^{\rho_1})\right) z_2^2 + \frac{1}{4}|\dot{\alpha}_1|^2. \end{aligned} \tag{29}$$

From Assumption 2.2, $h_1(\bar{x}_1, x_2^{\rho_1})$ is bounded on a compact set. Therefore, there exist k_1 and ν_1 such that

$$\dot{V}_1 \le -\gamma_1 V_1 + \beta_1\|\mu_1\|^2 + \delta_1$$

Step i ($2 \le i \le (n-1)$): Consider the ith equation of system (1), define ith error surface as $s_i = x_i - x_{ir}$, and let p_i denote $p_i(\bar{x}_n, t)$, then we have

$$\dot{s}_i = f_i(\bar{x}_i, x_{i+1}) + p_i - \dot{x}_{ir}. \tag{30}$$

From Assumption 2.2, it is known that $h_i(\bar{x}_i, x_{i+1}) > h_i^l > 0, \forall(\bar{x}_i, x_{i+1}) \in R^{i+1}$. In addition, since $\dot{x}_{ir}$ is not a function of x_{i+1}, it is clear that $\partial[f_i(\bar{x}_i, x_{i+1}) - \dot{x}_{ir}]/\partial x_{i+1} > h_i^l > 0$. According to Lemma 2.1, there exists a continuous function $x_{i+1}^* = \alpha_i^0(\bar{x}_i, \dot{x}_{ir})$ such that $f_i(\bar{x}_i, x_{i+1}^*) - \dot{x}_{ir} = 0$. Employing an SCNN to approximate x_{i+1}^* with $P_i = [\bar{x}_i, \dot{x}_{ir}] \in \Omega_{P_i} \subset R^{i+1}$, and using (2), we can rewrote (30) as

$$\dot{s}_i = h_i(\bar{x}_i, x_{i+1}^{\rho_i})(x_{i+1} - \hat{\Theta}_{ic}^T\Psi_{ic}(P_i) - \tilde{\Theta}_{ic}^T\Psi_i(P_i) - \epsilon_i(P_i)) + p_i \tag{31}$$

where $x_{i+1}^{\rho_i} = \alpha_i^0 + \rho_i(x_{i+1} - \alpha_i^0)$, $\hat{\Theta}_{ic} \in \Omega_{\Theta_{ic}} \subset R^{l_i\times 1}$, $\tilde{\Theta}_{ic} = \Theta_i^* - \hat{\Theta}_{ic}$, $\Psi_{ic}(P_i) \in R^{l_i\times 1}$ and l_i is the maximum number of neurons. Moreover, $|\epsilon_i(P_i)| \le \epsilon_i^*$ with the positive constant

ϵ_i^*. A virtual control function α_i for x_{i+1} is synthesized such that x_i tracks the desired trajectory x_{ir} as closely as possible. Therefore, α_i is chosen as:

$$\begin{aligned} &\alpha_i = \alpha_{ia} + \alpha_{is}, \alpha_{ia} = \hat{\Theta}_{ic}^T \Psi_{ic}(P_i) \\ &\alpha_{is} = \alpha_{is1} + \alpha_{is2}, \alpha_{is1} = -k_i s_i \end{aligned} \tag{32}$$

where α_{ia}, α_{is}, α_{is1} and α_{is2} have similar definitions as (23). Choose a positive constant δ_i, and let α_{is2} satisfy:

$$\begin{aligned} &\text{i.} \quad s_i h_i(\bar{x}_i, x_{i+1}^{\rho_i})(\alpha_{is2} - \tilde{\Theta}_{ic}^T \Psi_i(P_i) - \epsilon_i(P_i) + p_i^h) \le \delta_i \\ &\text{ii.} \quad s_i h_i(\bar{x}_i, x_{i+1}^{\rho_i}) \alpha_{is2} \le 0 \end{aligned} \tag{33}$$

where $p_i^h = \frac{p_i}{h_i(\bar{x}_i, x_{i+1}^{\rho_i})}$. Then, let α_{is2} be of the following form

$$\alpha_{is2} = -\frac{\pi_i h_i^u}{2\delta_i} s_i, \quad \pi_i \ge \|\Theta_{i\max} - \Theta_{i\min}\|_2^2 \|\Psi_i(P_i)\|_2^2 + \epsilon_i^{*2} + \frac{p_i^*}{h_i^l}. \tag{34}$$

Construct the ith first-order low-pass filter as:

$$\nu_i \dot{x}_{(i+1)r} + x_{(i+1)r} = \alpha_i, \quad x_{(i+1)r}(0) = \alpha_i(0), \tag{35}$$

define the $(i+1)$th error surface as $s_{i+1} = x_{i+1} - x_{(i+1)r}$, and let $z_{i+1} = x_{(i+1)r} - \alpha_i$. Then, the ith error dynamic system is described as

$$E_i : \dot{\lambda}_i = A_i \lambda_i + B_i \mu_i + d_i \tag{36}$$

where $\lambda_i = [s_i, z_{i+1}]^T$, $\mu_i = [s_{i+1}, \dot{\alpha}_i]^T$ and

$$\begin{aligned} A_i &= \begin{bmatrix} -k_i h_i(\bar{x}_i, x_{i+1}^{\rho_i}) & h_i(\bar{x}_i, x_{i+1}^{\rho_i}) \\ 0 & -\frac{1}{\nu_i} \end{bmatrix}, \quad B_i = \begin{bmatrix} h_i(\bar{x}_i, x_{i+1}^{\rho_i}) & 0 \\ 0 & -1 \end{bmatrix}, \\ d_i &= \begin{bmatrix} h_i(\bar{x}_i, x_{i+1}^{\rho_i})(\alpha_{is2} - \tilde{\Theta}_{ic}^T \Psi_i(P_i) - \epsilon_i(P_i) + p_i^h) \\ 0 \end{bmatrix}. \end{aligned}$$

Choose the Lyapunov function candidate $V_i = \frac{1}{2}\lambda_i^T \lambda_i$. In terms of Young's inequality and (33), the time derivative of V_i along (36) satisfies:

$$\begin{aligned} \dot{V}_i \le &- (k_i - 2) h_i(\bar{x}_i, x_{i+1}^{\rho_i}) s_i^2 + \frac{1}{4} h_i(\bar{x}_i, x_{i+1}^{\rho_i}) s_{i+1}^2 + \delta_i \\ &- \left(\frac{1}{\nu_i} - 1 - \frac{1}{4} h_i(\bar{x}_i, x_{i+1}^{\rho_i}) \right) z_{i+1}^2 + \frac{1}{4} |\dot{\alpha}_i|^2. \end{aligned} \tag{37}$$

From Assumption 2.2, $h_i(\bar{x}_i, x_{i+1}^{\rho_i})$ is bounded on a compact set, then there exist k_i and ν_i such that

$$\dot{V}_i \le -\gamma_i V_i + \beta_i \|\mu_i\|^2 + \delta_i$$

Step n: Consider the nth equation of system (1), define the nth error surface as $s_n = x_n - x_{nr}$, and let $p_n = p_n(\bar{x}_n, t)$ we can derive

$$\dot{s}_n = f_n(\bar{x}_n, u) + p_n - \dot{x}_{nr} \tag{38}$$

Because $\dot{x}_{nr}$ is not a function of u and $h_n(\bar{x}_n, u) > h_n^l > 0, \forall (\bar{x}_n, u) \in R^{n+1}$, it is clear that $\partial[f_n(\bar{x}_n, u) - \dot{x}_{nr}]/\partial u > h_n^l > 0$. Therefore, there exists a continuous function $u^* = \alpha_n^0(\bar{x}_n, \dot{x}_{nr})$ such that $f_n(\bar{x}_n, u^*) - \dot{x}_{nr} = 0$. Employing an SCNN to approximate u^* with $P_n = [\bar{x}_n, \dot{x}_{nr}] \in \Omega_{P_n} \subset R^{n+1}$, we can rewrite (38) as

$$\dot{s}_n = h_n(\bar{x}_n, u^{\rho_n}) \left(u - \hat{\Theta}_{nc}^T \Psi_{nc}(P_n) - \tilde{\Theta}_{nc}^T \Psi_n(P_n) - \epsilon_n(P_n) \right) + p_n \tag{39}$$

where u^{ρ_n}, $\hat{\Theta}_{nc}$, $\tilde{\Theta}_{nc}$, $\Psi_{nc}(P_n)$ and l_n are similar to those defined in step i, and $|\epsilon_n(P_n)| \leq \epsilon_n^*$ with a positive constant ϵ_n^*. A practical control law $u = \alpha_n$ is synthesized such that x_n tracks the desired trajectory x_{nr} as closely as possible. Therefore, α_n is designed as:

$$\begin{aligned} &\alpha_n = \alpha_{na} + \alpha_{ns}, \ \alpha_{na} = \hat{\Theta}_{nc}^T \Psi_{nc}(P_n) \\ &\alpha_{ns} = \alpha_{ns1} + \alpha_{ns2}, \ \alpha_{ns1} = -k_n s_n \end{aligned} \tag{40}$$

and α_{ns2} has the following form

$$\alpha_{ns2} = -\frac{\pi_n h_n^u}{2\delta_n} s_n, \quad \pi_n \geq \|\Theta_{n\max} - \Theta_{n\min}\|_2^2 \|\Psi_n(P_n)\|_2^2 + \epsilon_n^{*2} + \frac{p_n^*}{h_n^l}, \tag{41}$$

which satisfies

$$\begin{aligned} &\text{i.} \ \ s_n h_n(\bar{x}_n, u^{\rho_n}) \left(\alpha_{ns2} - \tilde{\Theta}_{nc}^T \Psi_n(P_n) - \epsilon_n(P_n) + p_n^h\right) \leq \delta_n \\ &\text{ii.} \ \ s_n h_n(\bar{x}_n, u^{\rho_n}) \alpha_{ns2} \leq 0. \end{aligned} \tag{42}$$

Defining $\lambda_n = s_n$, we have

$$E_n : \dot{\lambda}_n = -k_n h_n(\bar{x}_n, u^{\rho_n}) \lambda_n + d_n \tag{43}$$

where $d_n = h_n(\bar{x}_n, u^{\rho_n})(\alpha_{ns2} - \tilde{\Theta}_{nc}^T \Psi_n(P_n) - \epsilon_n(P_n)) + p_n$.

The time derivative of $V_n = \frac{1}{2}\lambda_n^T \lambda_n$ along (43) satisfies

$$\begin{aligned} \dot{V}_n &= -k_n h_n(\bar{x}_n, u^{\rho_n}) s_n^2 + s_n h_n(\bar{x}_n, u^{\rho_n})(\alpha_{ns2} - \tilde{\Theta}_{nc}^T \Psi_n(P_n) - \epsilon_n(P_n) + p_n^h) \\ &\leq -k_n h_n(\bar{x}_n, u^{\rho_n}) s_n^2 + \delta_n \end{aligned} \tag{44}$$

Because $h_n(\bar{x}_n, u^{\rho_n})$ is bounded, there exists k_n such that

$$\dot{V}_n \leq -\gamma_n V_n + \delta_n$$

Remark 4.1. *From the above design procedure, by replacing the derivative of virtual control $\dot{\alpha}_i$ with $\dot{x}_{(i+1)r}$, the dimension of neural networks' input vectors is significantly reduced; therefore, the controller is simpler than those proposed in [10,16]. Because of introducing α_i, the boundedness of $\dot{\alpha}_i$ should be guaranteed in the stability analysis. This issue will be explained in detail in the next section.*

Remark 4.2. *Because of utilizing the SCNN, structure parameters of neural networks, such as the location of centers and the number of neurons, are not necessary to be designed in advance, which is very useful especially when rare information about the plant is available. At the same time, with the structure learning algorithm, the size of SCNN can be maintained in a smaller scale, while the traditional design method may produce a large-scale RBFNN which contains many useless neurons. The simulation results will further illustrate the effectiveness of SCNN.*

Remark 4.3. *In the above design procedure, by virtue of the robust term α_{is2}, ARDSC guarantees the transient performance and the boundedness of output tracking error even without SCNN and adaptive law. Therefore, ARDSC provides an improved robustness performance for the closed-loop system.*

5. **Stability Analysis.** This section provides the stability and transient performance analysis of the closed-loop system in detail. Firstly, the semi-global stability of the closed-loop system is guaranteed by the proposed control method without considering SCNN and parameter adaptive law. Then, all signals in the closed-loop system are shown to be uniformly ultimately bounded when SCNN and adaptive law are applied, and the steady state performance is improved by virtue of the adaptive component. The results are summarized into the following two theorems.

Theorem 5.1. *Consider the closed-loop system consisting of plant (1) and controller (40). If all the Assumptions 2.1-3.1 are satisfied, and SCNN and the adaptive law (16) maintain switched off, then, for any bounded initial conditions* $\sum_{i=1}^{n} s_i^2 + \sum_{n=2}^{n} z_i^2 \le 2r_c$*, there exist* k_i *and* ν_i*, such that all the signals in the closed-loop system remain bounded, and the output tracking error is uniformly ultimately bounded by a neighborhood around zero.*

Proof: Firstly, $|\dot{\alpha}|$ is proved to be bounded by a continuous function. Then we give the stability analysis of the closed-loop system.

The time derivative of virtual control α_i can be exactly expanded as

$$\dot{\alpha}_i = \dot{\hat{\Theta}}_{ic}^T \Psi_{ic}(P_i) + \hat{\Theta}_{ic}^T \begin{bmatrix} \frac{\partial \Psi_{ic}(P_i)}{\partial P_{i1}} \dot{P}_{i1} \\ \vdots \\ \frac{\partial \Psi_{ic}(P_i)}{\partial P_{il_i}} \dot{P}_{il_i} \end{bmatrix} - k_i \dot{s}_i + \dot{\alpha}_{is2},\ i = 1, \cdots, n.$$

Note that x_1 depends on y_r and s_1, so α_1 is a function of $\hat{\Theta}_{1c}$, y_r, $\dot{y}_r$, s_1. Because $x_2 = s_2 + \alpha_1 + z_2$, x_2 relies on s_1, s_2, $\hat{\Theta}_{1c}$, y_r, $\dot{y}_r$ and z_2. According to (32), α_2 is a function of s_1, s_2, $\hat{\Theta}_{1c}$, $\hat{\Theta}_{2c}$, z_2, y_r and $\dot{y}_r$. Similarly, since $x_i = s_i + \alpha_{i-1} + z_i$, x_i relies on $s_1, \cdots, s_i$, $\hat{\Theta}_{ic}, \cdots, \hat{\Theta}_{(i-1)c}$, $z_2, \cdots, z_i$, y_r and $\dot{y}_r$. Therefore, α_i is a function of $\bar{s}_i = [s_1, \cdots, s_i]^T$, $\bar{\hat{\Theta}}_{ic} = [\hat{\Theta}_{1c}, \cdots, \hat{\Theta}_{ic}]^T$, $\bar{z}_i = [z_2, \cdots, z_i]^T$, y_r and $\dot{y}_r$. Consequently, the closed-loop system can be completely described by $\bar{s}_i$, $\bar{\hat{\Theta}}_{ic}$, $\bar{z}_i$, , y_r, $\dot{y}_r$.

Due to the fact that $\dot{\hat{\Theta}}_{ic}^T$, $\frac{\partial \Psi_{ic}(P_i)}{\partial P_{ij}} \dot{P}_{ij}$, $\dot{s}_i$ and $\dot{\alpha}_{is2}$ can be bounded by continuous functions, $|\dot{\alpha}_i|$ satisfies

$$|\dot{\alpha}_i| \le C_i(\bar{\Theta}_i, \bar{s}_i, \bar{z}_i, y_r, \dot{y}_r)$$

where C_i is a positive-definite continuous function. On the compact set $\Pi := \left\{ \bar{s}_n, \bar{z}_n \middle| \sum_{i=1}^{n} s_i^2 + \sum_{n=2}^{n} z_i^2 \le 2r_c \right\}$, the continuous function C_i has an upper bound

$$C_i(\bar{\Theta}_{ic}, \bar{s}_i, \bar{z}_i, y_r, \dot{y}_r) \le M_i,\ \ i = 1, \cdots, n-1$$

so

$$|\dot{\alpha}_i| \le M_i,\ \ i = 1, \cdots, n-1 \tag{45}$$

on the compact set Π.

Consider (29), (37) and (44), and choose the Lyapunov function $V_e = \sum_{i=1}^{n} V_i$, then its time derivative is

$$\begin{aligned} \dot{V}_e \le & \sum_{i=1}^{n-1} [-k_i h_i(\bar{x}_i, x_{i+1}^{\rho_i}) s_i^2 + \delta_i] - k_n h_n(\bar{x}_n, u^{\rho_n}) s_n^2 + \delta_n \sum_{i=1}^{n-1} h_i(\bar{x}_i, x_{i+1}^{\rho_i}) \left[2s_i^2 + \frac{1}{4} s_{i+1}^2 \right] \\ & + \sum_{i=1}^{n-1} \left[\left(-\frac{1}{\nu_i} + 1 + \frac{1}{4} h_i(\bar{x}_i, x_{i+1}^{\rho_i}) \right) z_{i+1}^2 + \frac{1}{4} |\dot{\alpha}_i|^2 \right] \end{aligned} \tag{46}$$

Note that $h_i(\bar{x}_i, x_{i+1}^{\rho_i}) > 0$. Clearly, there exist k_i and ν_i:

$$\begin{cases} k_1 \geq 2 + \zeta; \\ k_i \geq 2 + \dfrac{1}{4} + \zeta, \ i = 1, \cdots, n-1; \\ k_n \geq \dfrac{1}{4} + \zeta; \\ \nu_i \leq \dfrac{1}{\frac{1}{4}h_i^u + 1 + \zeta}, \ i = 2, \cdots, n; \end{cases} \tag{47}$$

such that

$$\dot{V}_e \leq -2\zeta V + b \tag{48}$$

where ζ is a positive constant, and $b = \sum\limits_{i=1}^{n} \delta_i + \sum\limits_{i=1}^{n-1} \frac{1}{4}M^2$. If we let $\zeta \geq \frac{b}{2r_c}$, then $\dot{V}_e \leq 0$ when $V_e(t) \geq r_c$. Therefore, $V_e(t) \leq r_c$ is an invariant set on $t \in [0, \infty)$.

Integrating the both sides of (48), we have

$$0 \leq V_e(t) \leq V_e(0)e^{-2\zeta t} + \frac{b}{2\zeta}(1 - e^{-2\zeta t}), \tag{49}$$

which indicates that s_i and z_j ($i = 1, \cdots, n$, $j = 2, \cdots, n$) are uniformly ultimately bounded. Because the adaptive law is switched off, $\hat{\Theta}_{ic}$ is bounded. Thus, α_i ($i = 1, \cdots, n$) is bounded as well. Moreover, in terms of Assumption 2.1 and the boundedness of s_i and z_j ($i = 1, \cdots, n, j = 2, \cdots, n$), x_i ($i = 1, \cdots, n$) are also bounded. Finally, we can conclude that all signals in the closed-loop system are bounded. In addition, the error bound can be pre-specified by choosing a proper value of ζ.

Remark 5.1. *From Theorem 5.1, it is clear that the ARDSC guarantees the pre-specified exponential convergence rate 2ζ and steady state performance (uniformly ultimately bound) by choosing suitable controller parameters k_i and ν_i, even without the adaptive compensator. However, the better performance is expected, the larger k_i and the smaller ν_i are required, and this finally results in a larger control input. Since it is impractical to require an arbitrarily large control signal, an adaptive compensation term can be introduced to further reduce the tracking error. Theorem* 5.2 *gives the stability analysis of the closed-loop system with the SCNN and adaptive being law activated.*

In the following, the stability of the closed-loop system with SCNN and adaptive law being applied is analyzed by employing the ISS concept and small-gain theorem.

Theorem 5.2. *Consider the closed-loop system consisting of plant (1) and the controller (40). If all the Assumptions 2.1-3.1 are satisfied, and the SCNN is activated and the adaptive law (16) is applied, then, for any bounded initial conditions $\sum\limits_{i=1}^{n} s_i^2 + \sum\limits_{n=2}^{n} z_i^2 \leq 2r_c$, there exist k_i and ν_i, such that all the signals in the closed-loop system still remain bounded, and the output tracking error is also uniformly ultimately bounded by a neighborhood around zero.*

Proof: Firstly, the error dynamic subsystem with equations (27), (36) and (43) can be regarded as a system with the inputs $\tilde{\Theta}_a = [\tilde{\Theta}_{1a}^T, \cdots, \tilde{\Theta}_{na}^T]^T$ and $\epsilon = [\epsilon_1, \cdots, \epsilon_n]^T$. As in the proof of Theorem 5.1, choose $V_e = \sum\limits_{i=1}^{n} V_i$ as the Lyapunov function, and for the sake of simplifying the representation, let h_i stand for $h_i(\bar{x}_i, x_{i+1}^{\rho_i})$, then we have

$$\begin{aligned}\dot{V}_e =& -\sum_{i=1}^{n} k_i h_i s_i^2 + \sum_{i=1}^{n} s_i h_i [\alpha_{is2} - \tilde{\Theta}_{ic}^T \Psi_i(P_i) - \epsilon_i(P_i)] \\ &+ \sum_{i=1}^{n-1} h_i(s_i s_{i+1} + s_i z_{i+1}) + \sum_{i=2}^{n} \left(-\frac{1}{\nu_{i-1}} z_i^2 - z_i \dot{\alpha}_{i-1} \right) + \sum_{i=1}^{n} s_i p_i.\end{aligned} \tag{50}$$

Note that $s_i h_i \alpha_{is2} \le 0$, and from (17)

$$\begin{aligned}\dot{V}_e \le& -\sum_{i=1}^{n} k_i h_i s_i^2 + \sum_{i=1}^{n} h_i \|\tilde{\Theta}_{ia}\| \|\Psi_{ia}(P_i)\| |s_i| + \sum_{i=1}^{n} s_i p_i \\ &- \sum_{i=1}^{n} s_i h_i \epsilon_i(P_i) + \sum_{i=1}^{n-1} h_i(s_i s_{i+1} + s_i z_{i+1}) + \sum_{i=2}^{n} \left(-\frac{1}{\nu_{i-1}} z_i^2 - z_i \dot{\alpha}_{i-1} \right) \\ \le& -\sum_{i=1}^{n-1} \left(k_i - 4 - \frac{1}{h_i} \right) h_i s_i^2 + \sum_{i=2}^{n} \frac{1}{4} h_i s_i^2 - \left(k_n - 2 - \frac{1}{h_n} \right) h_n s_n^2 + \sum_{i=1}^{n-1} \frac{1}{4} |\dot{\alpha}_i|^2 \\ &- \sum_{i=2}^{n} \left(\frac{1}{\nu_{i-1}} - 1 - \frac{1}{4} h_i \right) z_i^2 + \sum_{i=1}^{n} \frac{1}{4} h_i \psi_i^{*2} \|\tilde{\Theta}_{ia}\|^2 + \sum_{i=1}^{n} \frac{1}{4} h_i \epsilon_i^2(P_i) + \sum_{i=1}^{n} \frac{1}{4} p_i^{*2}\end{aligned}$$

Due to $h_i > h_i^l > 0$, let $z_1 = 0$, $\psi^* = \max\limits_{1 \le i \le n} \psi_i^*$ and $\dot{\alpha}_n = 0$, it is available to choose k_i and ν_i as:

$$\begin{cases} k_1 \ge 4 + \zeta + \dfrac{1}{h_1^l}; \\ k_i \ge 4 + \dfrac{1}{4} + \dfrac{1}{h_i^l} + \zeta, \ i = 2, \cdots, n-1; \\ k_n \ge 2 + \dfrac{1}{4} + \dfrac{1}{h_n^l} + \zeta; \\ \nu_i \le \dfrac{1}{\frac{1}{4} h_i^u + 1 + \zeta h_i^u}, \ i = 2, \cdots, n; \end{cases} \tag{51}$$

such that

$$\dot{V}_e \le \sum_{i=1}^{n} h_i \left[-\zeta s_i^2 - \zeta z_i^2 + \frac{1}{4} \psi^{*2} \|\tilde{\Theta}_{ia}\|^2 + \frac{1}{4} \epsilon_i^2(P_i) + \frac{|\dot{\alpha}_i|^2}{4h_i} + \frac{p_i^{*2}}{4h_i} \right]$$

where ζ is a positive constant. Consider that $|h_i| > h_i^l > 0$ and $|\dot{\alpha}_i| \le M_i$ on the compact set $\Pi := \left\{ \bar{s}_n, \bar{z}_n \middle| \sum\limits_{i=1}^{n} s_i^2 + \sum\limits_{n=2}^{n} z_i^2 \le 2r_c \right\}$, so

$$\dot{V}_e \le \sum_{i=1}^{n} h_i \left[-\zeta(1-b)(s_i^2 + z_i^2) - b\zeta(s_i^2 + z_i^2) + \frac{1}{4} \psi^{*2} \|\tilde{\Theta}_{ia}\|^2 + \frac{1}{4} \epsilon_i^2(P_i) + \delta_{is} \right]. \tag{52}$$

where $\delta_{is} = \frac{|M_i|^2 + p_i^{*2}}{4h_i^l}$, $0 < b < 1$. Denote $Z = [0, z_2, \cdots, z_n]^T$, $S = [s_1, \cdots, s_n]^T$ and $\Lambda = [S^T, Z^T]^T$. From (52), we can see that the error dynamic subsystem is input-to-state practical stable with Lyapunov gain [45]:

$$\gamma_{\tilde{\Theta}_a}^{\Lambda} = \left(\frac{\psi^{*2}}{4b\zeta} \right)^{\frac{1}{2}}, \ \gamma_{\epsilon}^{\Lambda} = \left(\frac{1}{4b\zeta} \right)^{\frac{1}{2}}. \tag{53}$$

Therefore, there exists a function β_Λ of class KL such that

$$\|\Lambda\| \leq \beta_\Lambda(\|\Lambda(0)\|, t) + \gamma^\Lambda_{\tilde{\Theta}_a}(\|\tilde{\Theta}_a\|_\infty) + \gamma^\Lambda_\epsilon(\|\epsilon\|_\infty) + \left(\frac{\delta_s}{b\zeta}\right)^{\frac{1}{2}} \tag{54}$$

where $\delta_s = \sum_{i=1}^{n} \frac{|M_i|^2 + p_i^{*2}}{4h_i^l}$.

Secondly, consider the parameter adaptive dynamics (16) as another subsystem which has the state $\tilde{\Theta}_a$ and the inputs Θ^* and Λ. Select the Lyapunov function $V_{\tilde{\Theta}} = \frac{1}{2}\tilde{\Theta}_a^T \Gamma^{-1}\tilde{\Theta}_a$, and in terms of $\tilde{\Theta}_a = \Theta_a^* - \hat{\Theta}_a$, it is clear that

$$\dot{V}_{\tilde{\Theta}} = -\tilde{\Theta}_a^T \Gamma^{-1} \mathrm{Proj}[-\Gamma(\Psi_a(P)S + \eta\hat{\Theta}_a)] \tag{55}$$

where $\Gamma = \mathrm{diag}\{\Gamma_1, \cdots, \Gamma_n\}^T$, $\Psi_a(P) = \mathrm{diag}\{\Psi_{ia}(P_i), \cdots, \Psi_{an}(P_n)\}$ and $\eta = \mathrm{diag}\{\eta_1 \boldsymbol{I}, \cdots, \eta_n \boldsymbol{I}\}$. Taking into account Property 2.1.ii and $\|S\| \leq \|\Lambda\|$, it is clear that

$$\begin{aligned}
\dot{V}_{\tilde{\Theta}} &\leq \tilde{\Theta}_a^T[\Psi_a(P)S + \eta(\Theta_a^* - \tilde{\Theta}_a)] \\
&\leq -(1-p)\tilde{\Theta}_a^T\tilde{\Theta}_a - p\tilde{\Theta}_a^T\tilde{\Theta}_a + \eta\tilde{\Theta}_a^T\Theta_a^* + \tilde{\Theta}_a^T\Psi_a(P)S \\
&\leq -(1-p)\|\tilde{\Theta}_a\|^2 - p\|\tilde{\Theta}_a\|^2 + \|\tilde{\Theta}_a\|\|\Theta_a^*\| + \psi^*\|\tilde{\Theta}_a\|\|\Lambda\|
\end{aligned}$$

where $0 < p < 1$. Similarly, the parameter adaptive subsystem is also input-to-state stable with Lyapunov gain:

$$\gamma_\Lambda^{\tilde{\Theta}_a} = \frac{\psi^*}{p}, \ \gamma_{\Theta_a^*}^{\tilde{\Theta}_a} = \frac{1}{p}. \tag{56}$$

and there exists a function $\beta_{\tilde{\Theta}_a}$ of class KL such that

$$\|\tilde{\Theta}_a\| \leq \beta_{\tilde{\Theta}_a}(\|\tilde{\Theta}_a(0)\|, t) + \gamma_\Lambda^{\tilde{\Theta}_a}(\|\Lambda\|_\infty) + \gamma_{\Theta_a^*}^{\tilde{\Theta}_a}(\|\Theta_a^*\|_\infty) \tag{57}$$

Based on the small-gain theorem [46], the interconnected system with (27), (36), (43) and (16) is input-to-state practical stable with the state $x = (\Lambda, \tilde{\Theta}_a)$ and the input $u = (\epsilon, \Theta_a^*)$, if

$$\lambda_\Lambda^{\tilde{\Theta}_a} \circ \lambda^\Lambda_{\tilde{\Theta}_a}(s) < s. \tag{58}$$

Therefore, we can choose

$$\zeta > \frac{\Psi^{*4}}{4p^2 b}. \tag{59}$$

Since $\epsilon < \epsilon^*$, ϵ_a^* and Θ^* are assumed to be constants, the boundedness of Λ and $\tilde{\Theta}_a$ can be established. As a result, x, $\hat{\Theta}_a$ and u are bounded. Therefore, all the signals in the closed-loop system remain bounded. In addition, we must show the compact set Π is an invariant set to guarantee the boundedness of $\dot{\alpha}_i$. Considering (52), Assumption 3.1 and Property 2.1, we have

$$\begin{aligned}
\dot{V}_e &\leq \sum_{i=1}^n h_i\left[-\zeta(s_i^2 + z_i^2) + \frac{1}{4}\psi^{*2}Q_i^2 + \frac{1}{4}\epsilon_i^{*2} + \delta_{is}\right] \\
&\leq \sum_{i=1}^n h_i\left[-\zeta\|\Lambda_i\|^2 + \Delta_{is}\right]
\end{aligned} \tag{60}$$

where $\Lambda_i = [s_i, z_i]^T$ and

$$Q_i = \|\Theta_{i(\max)} - \Theta_{i(\min)}\|, \quad \Delta_{is} = \frac{1}{4}\psi^{*2}Q_i^2 + \frac{1}{4}\epsilon_i^{*2} + \delta_{is}.$$

Therefore, if we let

$$\zeta > \frac{\Delta_s}{2r_c}, \tag{61}$$

where $\Delta_s = \max\{\Delta_{is}\}$, $i = 1, \cdots, n$, then $\dot{V}_e \leq 0$ when $V_e(t) > r_c$. Therefore, $V_e(t) \leq r_c$ is an invariant set on $t \in [0, \infty)$, and the assumption $|\dot{\alpha}_i| \leq M_i$ is reasonable. Finally, if

$$\zeta = \max\left\{\frac{\Psi^{*4}}{4p^2 b}, \frac{\Delta_s}{2r_c}\right\}, \tag{62}$$

then the closed-loop system is input-to-state practical stable, and the compact set $\Pi := \left\{\bar{s}_n, \bar{z}_n | \sum_{i=1}^{n} s_i^2 + \sum_{n=2}^{n} z_i^2 \leq 2r_c\right\}$ is an invariant set. Because ϵ^* and Θ^* are assumed to be constants, all the signals in the closed-loop system remain bounded, and the output tracking error is also uniformly ultimately bounded by a neighborhood around zero.

Remark 5.2. *From Theorem 5.2, by choosing suitable control parameters, the compact set* $\sum_{i=1}^{n} s_i^2 + \sum_{n=2}^{n} z_i^2 \leq 2r_c$ *is proved to be an invariant set on* $t \in [0, \infty)$, *which guarantees the boundedness of* $\dot{\alpha}$. *Consequently, it is reasonable that the error dynamic subsystem is input-to-state practical stable.*

Remark 5.3. *Theorem 5.1 and Theorem 5.2 show that, with the proposed ARDSC, the bounded initial conditions result in the boundedness of all the signals in the closed-loop system, which guarantees that it is possible to construct SCNNs with bounded approximation error* ϵ_i *on a sufficiently large compact set* Ω_{P_i} *such that* $P_i = [\bar{x}_i, \dot{x}_{ir}] \in \Omega_{P_i} \subset R^{i+1}$.

6. Simulation Results.

To verify the effectiveness and advantages of the SCNN based ARDSC, numerical simulations are given in this section.

6.1. Example 1.

Consider the following second-order complete non-affine pure-feedback system [10]:

$$\begin{cases} \dot{x}_1 = x_1 + x_2 + \dfrac{x_2^3}{5} \\ \dot{x}_2 = x_1 x_2 + u + \dfrac{u^3}{7} \\ y = x_1 \end{cases} \tag{63}$$

The famous van der Pol oscillator is considered as the reference model [10]:

$$\begin{cases} \dot{x}_{d1} = x_{d2} \\ \dot{x}_{d2} = -x_{d1} + \beta(1 - x_{d1}^2)x_{d2} \\ y = x_{d1} \end{cases} \tag{64}$$

where $\beta = 0.2$ in this simulation and the initial states are $[x_{d1}(0)\ x_{d2}(0)]^T = [1.5\ 0.8]^T$. The objective is to design a control law u such that the output of system (63) tracks the desired signal x_{d1} as closely as possible, and meanwhile, all the signals in the closed-loop system are bounded.

There are two SCNNs used in this controller design, and the bounds of θ_{ij} ($i = 1, 2, j = 1, \cdots, l_i$) are set as $\theta_{ij\,\min} = -100$ and $\theta_{ij\,\max} = 100$. The parameters of the ARDSC and SCNNs are chosen as follows: $k_1 = k_2 = 20$, $\pi_1 = 20$, $\pi_2 = 30$, $\delta_1 = \delta_2 = 5$, $\nu_1 = 0.001$, $\Gamma_1 = 50I_{l_1(t)}$, $\Gamma_2 = 50I_{l_2(t)}$, $\eta_1 = \eta_2 = 0.2$, $d_{g1} = d_{g2} = 0.5$, $d_{p1} = d_{p2} = 1$, $S_{p1} = S_{p2} = 0.05$, $DI_{p1} = DI_{p2} = 0.1$, $\tau_{p1} = \tau_{p2} = 0.1$, $\epsilon_{p1} = \epsilon_{p2} = 0.001$, $\epsilon_{rw1} = \epsilon_{rw2} = 0.25$, $\xi_{w1} = \xi_{w2} = 0.1$, $\tau_{\xi 1} = \tau_{\xi 2} = 0.1$, $e_{w1} = e_{w2} = 0.1$, $\sigma_{01} = \sigma_{02} = 2$.

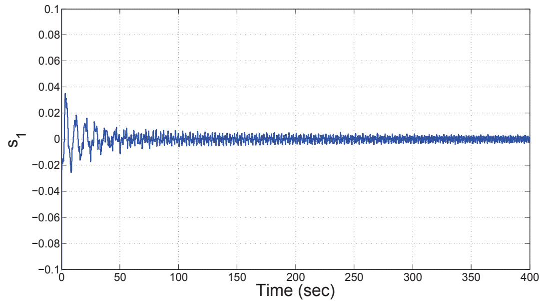

FIGURE 1. Output tracking error using SCNN based ARDSC in example 1

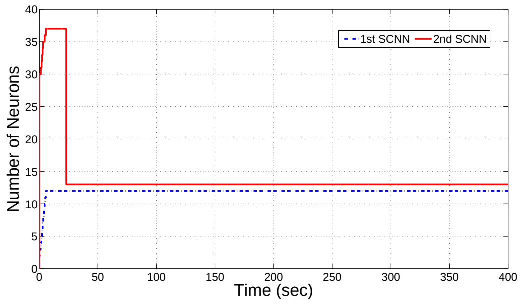

FIGURE 2. The number of neurons in example 1

The simulation results are shown in Figure 1 and Figure 2. From Figure 1, we see that, using the proposed control algorithm, the tracking error gradually converges to a small neighborhood around zero. Compared with the result in [10], the proposed method has a simple controller form as shown by the mathematical expression in Section 4. In addition, Figure 2 shows that the number of neurons is significantly reduced to less than 15 by the proposed structure learning algorithm, while there were 160 neurons in the controller designed in [10] and 180 neurons in [21]. Consequently, we can confirm that the proposed control method is simpler and requires less hardware resources.

6.2. **Example 2.** The following third-order system is considered in this case:

$$\begin{cases} \dot{x}_1 = x_1^2 \sin(x_1) + x_2^3 + p_1 \\ \dot{x}_2 = x_1 \cos(x_2) + x_1 x_2 + (2 + 0.7 \sin(x_1 x_2)) x_3 + p_2 \\ \dot{x}_3 = x_1 x_3 + x_3 \sin(x_2) + (1 + 0.1 \cos(x_2 x_3)) u + u^3 + p_3 \\ y = x_1 \end{cases} \tag{65}$$

where $p_1 = 0.1\cos(x_1 x_2 x_3 t)$, $p_2 = 0.1\sin(x_2 x_3 t)$ and $p_3 = 0.05\sin(x_1 t)$. The initial states are $[x_1\ x_2\ x_3]^T = [0.5\ \ -0.5\ \ 0]^T$. The control objective is to design a controller for the system (65) such that the output y tracks the reference signal $y_d = \sin(2\pi t)$ as closely as possible.

In the controller design, three SCNNs are used, and the bounds of θ_{ij} $(i = 1, 2, 3, j = 1, \cdots, l_i)$ are set as $\theta_{ij\,\min} = -100$ and $\theta_{ij\,\max} = 100$. The parameters of the ARDSC and SCNNs are chosen as follows: $k_1 = 20$, $k_2 = 30$, $k_3 = 40$, $\pi_1 = 20$, $\pi_2 = 30$, $\pi_3 = 40$, $\delta_1 = \delta_2 = \delta_3 = 5$, $\nu_1 = \nu_2 = 0.001$, $\Gamma_1 = 50I_{l_1(t)}$, $\Gamma_2 = 50I_{l_2(t)}$, $\Gamma_3 = 50I_{l_3(t)}$, $\eta_1 = \eta_2 = \eta_3 = 0.2$, $d_{g1} = d_{g2} = d_{g3} = 1$, $d_{p1} = d_{p2} = d_{p3} = 2$, $S_{p1} = S_{p2} = S_{p3} = 0.05$, $DI_{p1} = DI_{p2} = DI_{p3} = 0.8$, $\tau_{p1} = \tau_{p2} = \tau_{p3} = 0.1$, $\epsilon_{p1} = \epsilon_{p2} = \epsilon_{p3} = 0.001$, $\epsilon_{rw1} = \epsilon_{rw2} = \epsilon_{rw3} = 0.25$, $\xi_{w1} = \xi_{w2} = \xi_{w2} = 0.6$, $\tau_{\xi 1} = \tau_{\xi 2} = \tau_{\xi 3} = 0.1$, $e_{w1} = 0.05$, $e_{w2} = e_{w3} = 0.1$, $\sigma_{01} = 0.5$, $\sigma_{02} = 1$, $\sigma_{03} = 4$.

6.2.1. *ARDSC without SCNN and adaptive law.* In this simulation, SCNN and adaptive law are disabled and the output tracking error is shown in Figure 3. From Figure 3, it is clear that the output tracking error converges to the ultimate bound within no more than one second, which confirms that the ARDSC guarantees transient performance and system stability even without the adaptive compensator. Actually, when some knowledge of the plant is known a priori, transient performance and ultimate bound of output tracking error can be specified in advance by controller parameter design.

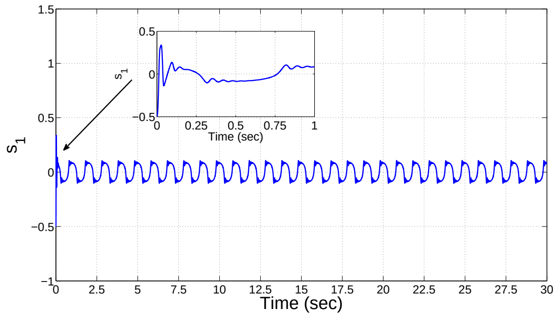

FIGURE 3. Output tracking error using ARDSC without SCNN in example 2

6.2.2. *ARDSC with SCNN and adaptive law.* In this simulation, SCNN and adaptive law are activated, and simulation results are shown in Figures 4-6.

Figure 4 shows the output tracking error using SCNN based ARDSC, and it can be seen that tracking error converges into a smaller region, as compared with the result shown in Figure 3. This phenomenon indicates that steady state tracking performance is improved due to the parameter learning of SCNN.

Figure 5 presents the changes of neuron number over time. Figure 6 demonstrates how the width of new generated neurons is regulated by the structure leaning strategy. At the beginning (0s $\sim$ 3s), since system states cover a wide region, large width is chosen and lots of neurons are created to capture system dynamics. In the period from 3s to 10s, useless neurons are pruned when system states converge into a small region. During the

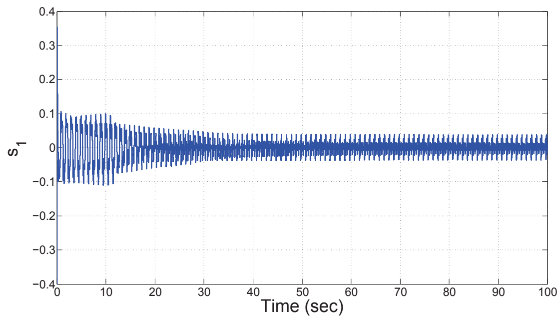

FIGURE 4. Output tracking error using SCNN based ARDSC in example 2

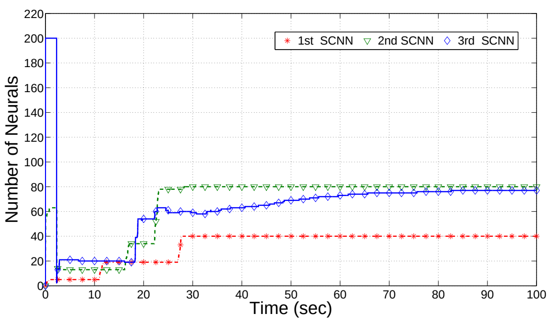

FIGURE 5. The number of neurons in example 2

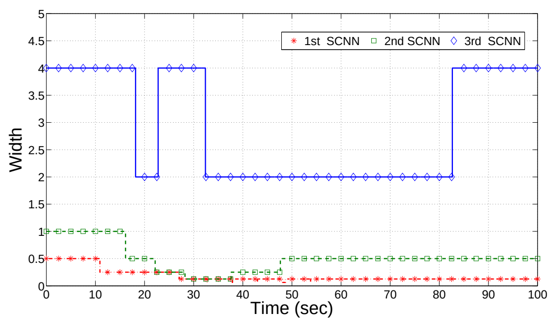

FIGURE 6. The width of the new generated neurons in example 2

final stage (10s $\sim$ 100s), based on output tracking error and system dynamic behavior, the width adjustment strategy automatically regulates resolution of SCNN by changing neuron width to improve the tracking performance.

From Figures 5 and 6, the following three conclusions can be made:

(i) For high-order pure-feedback nonlinear systems, the proposed control algorithm is much easier to be implemented, which can be verified from the mathematical expression of the controller, the dimension of the input vector and the number of neurons shown in Figure 5.

(ii) Due to lacking of knowledge about system (65), relative large widths are chosen as the initial width for SCNNs. From Figure 6, we can find that, when the resolution of the radial basis function is not competent for the control system, the radii of the generated neurons will be decreased to obtain a high resolution.

(iii) The proposed SCNN is useful for reducing the scale of neural networks. Moreover, it does not require constructing neural networks manually at the initial phase of controller design, and this feature is especially useful for unknown nonlinear systems.

7. **Conclusion.** This paper proposed an SCNN based ARDSC for nonlinear systems in complete non-affine pure-feedback form. The ARDSC guarantees the boundedness of all signals in the closed-loop system even in the absence of adaptive components. Meanwhile, the complexity of the controller is significantly reduced via DSC technique and SCNN without deteriorating control performance. Stability analysis of the closed-loop system is provided by employing ISS analysis and small-gain theorem. Finally, two simulations are performed to verify the effectiveness of the proposed method.

Acknowledgment. This work is supported by Projects of Major International (Regional) Joint Research Program NSFC (61120106010), and National Science Fund for Distinguished Young Scholars, No. 60925011.

REFERENCES

[1] I. Kanellakopoulos, P. V. Kokotović and A. S. Morse, Systematic design of adaptive controller for feedback linearizable systems, *IEEE Trans. Automatic Control*, vol.36, no.11, pp.1241-1253, 1991.

[2] M. M. Polycarpou, Stable adaptive neural shceme for nonlinear systems, *IEEE Trans. Automatic Control*, vol.41, no.3, pp.447-451, 1996.

[3] T. P. Zhang, S. S. Ge and C. C. Hang, Adaptive neural network control for strict-feedback nonlinear systems using backstepping design, *Automatica*, vol.36, no.12, pp.1835-1846, 2000.

[4] Q. Zhou, P. Shi, J. Lu and S. Xu, Adaptive output feedback fuzzy tracking control for a class of nonlinear systems, *IEEE Trans. on Fuzzy Systems*, vol.19, no.5, pp.972-982, 2011.

[5] J. Yu, Y. Ma, B. Chen and H. Yu, Adaptive fuzzy tracking control for induction motors via backstepping, *ICIC Express Letters*, vol.5, no.2, pp.425-431, 2011.

[6] M. Krstić, I. Kanellakopoulos and P. Kokotović, *Nonlinear and Adaptive Control Design*, Wiley, New York, 1995.

[7] C. Kwan and F. L. Lewis, Robust backstepping control of nonlinear systems using neural networks, *IEEE Trans. Systems, Man, and Cybernetics – Part A: Systems and Humans*, vol.30, no.6, pp.753-766, 2000.

[8] C. Liu, S. Tong and Y. Li, Adaptive fuzzy backstepping output feedback control for nonlinear systems with unknown sign of high-frequency gain, *ICIC Express Letters*, vol.4, no.5(A), pp.1689-1694, 2010.

[9] Z. Wu, M. Cui and P. Shi, Backstepping control in vector form for stochastic Hamiltonian systems, *SIAM Journal on Control and Optimization*, vol.50, no.2, pp.925-942, 2012.

[10] C. Wang, D. J. Hill, S. S. Ge and G. Chen, An ISS-modular approach for adaptive neural control of pure-feedback systems, *Automatica*, vol.42, no.5, pp.723-731, 2006

[11] P. Li, J. Chen, T. Cai and B. Zhang, Adaptive control of air delivery system for PEM fuel cell using backstepping, *Proc. of the 8th Asian Control Conference*, Kaohsiunga, pp.1282-1287, 2011.

[12] S. S. Ge and C. Wang, Adaptive NN control of uncertain nonlinear pure-feedback systems, *Automatica*, vol.38, no.4, pp.671-682, 2002.

[13] D. Wang and J. Huang, Adaptive neural network control for a class of uncertain nonlinear systems in pure-feedback form, *Automatica*, vol.38, no.8, pp.1365-1372, 2002.

[14] T. P. Zhang and S. S. Ge, Adaptive dynamic surface control of nonlinear systems with unknown dead zone in pure feedback form, *Automatica*, vol.44, no.7, pp.1895-1903, 2008.

[15] D. Wang, Neural network-based adaptive dynamic surface control of uncertain nonlinear pure-feedback systems, *International Journal of Robust and Nonlinear Control*, vol.21, no.5, pp.527-541, 2011.

[16] T. P. Zhang, H. Wen and Q. Zhu, Adaptive fuzzy control of nonlinear systems in pure-feedback form based on input-to-state stability, *IEEE Trans. Fuzzy Systems*, vol.18, no.1, pp.80-93, 2010.

[17] M. Wang, S. S. Ge and K. Hong, Approximation-based adaptive tracking control of pure-feedback nonlinear systems with multiple unknown time-varying delays, *IEEE Trans. Neural Networks*, vol.21, no.11, pp.1804-1815, 2010.

[18] B. B. Ren, S. S. Ge, C. Y. Su and T. H. Lee, Adaptive neural control for a class of uncertain nonlinear systems in pure-feedback form with hysteresis input, *IEEE Trans. Systems, Man, and Cybernetics – Part B: Cybernetics*, vol.39, no.2, pp.431-443, 2009.

[19] M. Wang, X. Liu and P. Shi, Adaptive neural control of pure-feedback nonlinear time-delay systems via dynamic surface technique, *IEEE Trans. on Systems, Man and Cybernetics, Part B: Cybernetics*, vol.41, no.6, pp.1681-1692, 2011.

[20] A. M. Zou, Z. G. Hou and M. Tan, Adaptive control of a class of nonlinear pure-feedback systems using fuzzy backstepping approach, *IEEE Trans. Fuzzy Systems*, vol.16, no.4, pp.886-897, 2008.

[21] Q. Zhao and Y. Lin, Adaptive dynamic surface control for pure-feedback systems, *International Journal of Robust and Nonlinear Control*, vol.22, no.14, pp.1647-1660, 2011.

[22] B. Yao and M. Tomizuka, Adaptive robust control of SISO nonlinear systems in a semi-strict feedback form, *Automatica*, vol.33, no.5, pp.893-900, 1997.

[23] B. Yao and M. Tomizuka, Adaptive robust control of MIMO nonlinear systems in semi-strict feedback forms, *Automatica*, vol.37, no.9, pp.1305-1321, 2001.

[24] X. B. Liu, H. Y. Su, B. Yao and J. Chu, Adaptive robust control of nonlinear systems with dynamic uncertainties, *International Journal of Adaptive Control and Signal Processing*, vol.23, no.4, pp.353-377, 2009.

[25] G. Z. Zhang, J. Chen and Z. P. Li, Adaptive robust control for servo mechanisms with parially unknown states via dynamic surface, *IEEE Trans. Control Systems Technology*, vol.18, no.3, pp.723-731, 2010.

[26] Y. Hone and B. Yao, A globally stable saturated desired compensation adaptive robust control for linear motor systems with comparative experiments, *Automatica*, vol.43, no.10, pp.1840-1848, 2007.

[27] Z. J. Yang, K. Kunitoshi, S. Kanae and K. Wada, Adaptive robust output-feedback control of a magnetic levitation system by K-filter approach, *IEEE Trans. Industrial Electronics*, vol.55, no.1, pp.390-399, 2008.

[28] G. Z. Zhang, J. Chen and Z. P. Li, Identifier based adaptive robust control for servo mechanisms with improved transient performance, *IEEE Trans. Industrial Electronics*, vol.57, no.7, pp.2536-2547, 2010.

[29] J. Chen, Z. Li, G. Zhang and M. Gan, Adaptive robust dynamic surface control with composite adaptation laws, *International Journal of Adaptive Control and Signal Processing*, vol.24, no.12, pp.1036-1050, 2010.

[30] D. Swaroop, J. K. Hedrick, P. P. Yip and J. C. Gerdes, Dynamic surface control for a class of nonlinear systems, *IEEE Trans. Automatic Control*, vol.45, no.10, pp.1893-1899, 2000.

[31] S. Tong, Y. Li and T. Wang, Adaptive fuzzy backstepping fault-tolerent control for uncertain nonlinear systems based on dynamic surface, *International Journal of Innovative Computing, Information and Control*, vol.5, no.10(A), pp.3249-3261, 2009.

[32] Y. Yu, Y. Bai, T. Zhang and X. Wang, The provenance analysis based on self-organizing neural network improved by immune algorithm, *ICIC Express Letters, Part B: Applications*, vol.2, no.5, pp.1057-1062, 2011.

[33] J. Platt, A resource-allocationg network for function interpolation, *Neural Computation*, vol.3, no.2, pp.213-225, 1991.

[34] S. Q. Wu and M. J. Er, Dynamic fuzzy neural networks – A novel approach to function approximation, *IEEE Trans. Systems, Man, and Cybernetics – Part B: Cybernetics*, vol.30, no.2, pp.358-364, 2000.

[35] S. Q. Wu, M. J. Er and Y. Gao, A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks, *IEEE Trans. Fuzzy Systems*, vol.9, no.4, pp.578-594, 2001.

[36] F. J. Lin and C. H. Lin, A permanent-magnet synchronous motor servo drive using self-constructing fuzzy neural network controller, *IEEE Trans. Energy Conversion*, vol.19, no.1, pp.66-72, 2004.
[37] Y. Gao and M. J. Er, Online adaptive fuzzy neural identification and control of a class of MIMO nonlinear systems, *IEEE Trans. Fuzzy Systems*, vol.11, no.4, pp.462-477, 2003.
[38] G. P. Liu, V. Kadirkamanathan and S. A. Billings, Variable neural networks for adaptive control of nonlinear systems, *IEEE Trans. Systems, Man, and Cybernetics – Part C: Applications and Reviews*, vol.29, no.1, pp.34-43, 1999.
[39] J. H. Park, S. H. Huh, S. H. Kim, S. J. Seo and G. T. Park, Direct adaptive controller for nonaffine nonlinear systems using self-structuring neural networks, *IEEE Trans. Neural Networks*, vol.16, no.2, pp.414-422, 2005.
[40] C. F. Hsu, Self-organizing adaptive fuzzy neural control for a class of nonlinear systems, *IEEE Trans. Neural Networks*, vol.18, no.4, pp.1232-1241, 2007.
[41] C. F. Hsu, Intelligent position tracking control for LCM drive using table online self-constructing recurrent neural network controller with bound architecture, *Control Engineering Practice*, vol.17, no.6, pp.714-722, 2009.
[42] H. C. Lu, M. H. Chang and C. H. Tsai, Adaptive self-constructing fuzzy neural network controller for hardware implementation of an inverted pendulum system, *Applied Soft Computing*, vol.11, no.5, pp.3962-3975, 2011.
[43] H. K. Khalil, *Nonlinear Systems*, 3rd Edition, Englewood Cliffs, Prentice Hall, NJ, 2002.
[44] A. J. Kurdila, F. J. Narcowich and J. D. Ward, Persistency of excitation in identification using radial basis function approximants, *SIAM Journal of Control and Optimization*, vol.33, no.2, pp.625-642, 1995.
[45] P. D. Christofides and A. R. Teel, Singular perturbations and input-to-state stability, *IEEE Trans. Automatic Control*, vol.41, no.11, pp.1645-1650, 1996.
[46] Z. P. Jiang, A. R. Teel and L. Praly, Small-gain theorem for ISS systems and applications, *Mathematics of Control Signals and Systems*, vol.7, no.2, pp.95-120, 1994.
[47] T. P. Zhang, Q. Zhu and Y. Q. Yang, Adaptive neural control of non-affine pure-feedback nonlinear systems with input nonlinearity and perturbed uncertainties, *International Journal of Systems Science*, vol.43, no.4, pp.691-706, 2012.