IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 7 February 2023, accepted 19 February 2023, date of publication 27 February 2023, date of current version 10 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3249346

APPLIED RESEARCH

A Study on the Design of Error-Based Adaptive Robust RBF Neural Network Back-Stepping Controller for 2-DOF Snake Robot's Head

SUNG-JAE KIM[1], MAOLIN JIN[2], (Senior Member, IEEE), AND JIN-HO SUH[1,3], (Senior Member, IEEE)

[1]Department of Intelligent Robot Engineering, Pukyong National University, Busan 48513, South Korea
[2]Human-Centered Robotics Research Center, Korea Institute of Robotics and Technology Convergence, Pohang 37553, South Korea
[3]Major of Mechanical System Engineering, Pukyong National University, Busan 48513, South Korea

Corresponding author: Jin-Ho Suh (suhgang@pknu.ac.kr)

This work was supported by the Technology Innovation Program (Development of a wireless teleoperable relief robot, for detecting, searching, and responding in narrow space) by the Ministry of Trade, Industry and Energy (MOTIE) in south Korea under Grant 20018110.

ABSTRACT In this paper, we will propose the controller of a 2-DOF head in a snake robot for effective image data reading when the snake robot is driving, and present an error-based adaptive robust radial base function neural network back-stepping controller. The snake robot head system is a nonlinear system, and there are unmeasurable disturbances that occur while driving. To solve this problem, we use back-stepping controller and radial basis function neural network to compensate for unmeasurable disturbances to improve the steady state. In order to further compensate for the network approximation error that occurs during training or large signal changes, we use adaptive coefficients and error functions to approximate the network approximation error and design adaptive robust terms. Compared to the previous controller, the proposed controller can actively compensate for large signal changes and has the advantage of not generating residual input in a steady state. The proposed controller is based on Lyapunov function candidate to design an adaptive law and prove the system stability. The stability of the control system is proven through Lyapunov analysis and bounded. The proposed controller compared and verified the controller performance for two inputs through simulation and presented the efficiency of the controller.

INDEX TERMS Intelligent control, back-stepping control, radial basis function neural network, robust term, snake robot.

I. INTRODUCTION

The snake robot is inspired by the locomotion of a biological snake that can move in a variety of environmental conditions, such as a challenging environment, a narrow space, and vertical movement [1], [2], [3], [4], [5], [6]. The snake robot has the advantage of being able to drive in a challenging environment using a hyper-redundant structure, so it is in the spotlight as a robot to be used in an environment that requires mobility [7], [8], [9], [10], [11], [12]. Through research, the snake robot can efficiently locomote in many challenging environments using various locomotion. To apply the various locomotion of the biological snake to the robot, a different sinusoidal signal is applied to the robot module [13], [14], [15], [16], [17], [18], [19], [20], [21]. However, this control method causes a large swing in the snake's robot head with a mounted camera. As shown in Figure. 1, the snake robot's head swing makes the video information difficult to read. Therefore, it is difficult for the operator to control the snake robot and obtain environmental information [22]. Therefore, camera stabilization must be considered essential for using a snake robot as a mobile robot. To improve this problem, it is necessary to control the snake robot's head like a gimbal. In Previous research cases, using the Kinematics base control method to change the gait [23], [24], [25], [26], [27], [28], [29], [30]. However, despite using several driving modules for

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaojie Su.

control, they do not have a rigorous analysis of driving performance. There is only one method that can be applied to only one locomotion. In addition, there are many unmeasurable disturbances in a rough environment, snake robot requires an adaptive control method rather than a kinematics-based control method. Several considerations exist to devise an adaptive control law. The snake robot's head is a non-linear system and the head swings due to locomotion, and at the same time, the control performance is affected by an unknown disturbance depending on the environment. Therefore, the snake robot head system is a non-linear Multi-Input Multi-Output system with unknown disturbances, and it must compensate for head swing due to locomotion in these conditions.

Nonlinearity, disturbance, and mathematical model uncertainty are one of the major difficulties in the controller design problem. In a robot system, it is difficult to control a robot due to a high degree of freedom, non-linearity, and disturbance. To solve this problem, many researchers have designed nonlinear controllers or disturbance observers or studied adaptive control. On the other hand, intelligent control is being studied in various systems due to the development of computer technology [30], [31], [32], [33], [34], [35], [36]. Among them, artificial neural network control is attracting a lot of attention. Neural network control can easily solve the effects of nonlinearities, disturbances, and parameter fluctuations without complex system dynamics. In addition, the self-learning ability can avoid complex mathematical interpretations such as adaptation laws, and can solve highly non-linear problems using multi-layer neural networks and non-linear activation functions [37]. In this paper, we propose to use RBF NN(Radial Basis Function Neural Network). RBF NN has a nonlinear activate function and simple structure that avoids unnecessary and lengthy calculations as compared to the multi-layer neural network. Due to these characteristics, it is useful for online control in combination with a mechanical system such as a robot system [37], [38]. However, only using neural network control is dangerous because it is difficult to prove the stability of the system, and the system may become unstable due to system characteristics and initial value problems. In this regard, many research and application methods have been studied and applied in combination with control theory. Among them, the BSC(Back-Stepping Control) method is one of the powerful methods that can stably control the nonlinear system by controlling the nonlinearity step by step using a virtual control input based on the Lyapunov theory. However, there is a limit to accurately controlling it due to uncertainty and unknown disturbances. To solve this problem, we estimated the disturbance including system uncertainty through an artificial neural network and achieved high control performance by combining BSC and neural network. This combined controller has been applied in various fields to prove its ability [39], [40], [41], [42], [43], [44], [45], [46], [47], [48]. Whereas, neural networks have problems in that convergence speed delays or instability occur due to discontinuous signals or parameter changes in the initial control stage. As a research case [34], [35], [38], [39], [40], [41], [42], [43], [44], [45], a control input called robust term was used. As a research case, an additional control input called a ''robust term'' was applied in the study. They could use the robust term to increase control performance. The robust term additionally applies a fixed control input according to the sign of the error. The stability of the control using the Robust term has been proven based on *Lyapunov's theory*, but since it uses a fixed constant value, the unnecessary control input is continuously supplied even in a steady state. In addition, since the control gain cannot be actively adjusted, there is no change in the control input according to the signal. So, it is difficult to respond to large signal changes.

Other recent research is [49], [50], [51], [52], and [53]. Reference [49] is a proposed adaptive funnel back-stepping control scheme for high-order nonlinear systems with uncertainty [50], [51], [52], [53] is a proposed prescribed performance control with a fuzzy-neural network for nonlinear systems with uncertainty and constraint conditions. Their research results proved effective through simulation, but it requires a complex design process, control gain, and strict performance function settings to implement the system.

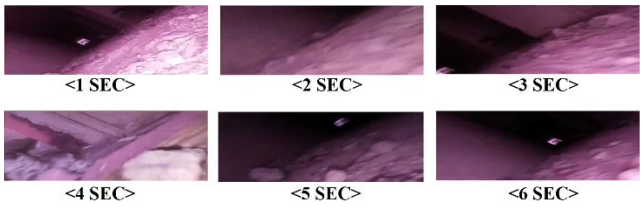


FIGURE 1. Image data when moving snake robot.

In this paper, we propose an RBF NN back-stepping control with an error-based robust adaptive input applied with a reverse saturation function as a snake head module control method for stabilizing the image data of a snake robot. The snake robot head designs the back-stepping controller using the 2-DOF nonlinear dynamic equation. Next, improve the performance of the back-stepping controller by estimating and compensating for the error generated from the influence of disturbance and uncertainty of the mathematical model through the RBF NN. The RBF NN consists of a minimum number of nodes to minimize computing power and approximates all disturbances by configuring the input as an error signal without using system parameters. An error-based adaptive robust term has an adaptive coefficient that can respond to large signal changes, which is a disadvantage of the previous robust term. In addition, the reverse saturation function is applied to the unnecessary input supply occurring in the steady state to remove unnecessary control input. RBF NN weight update rule and robust adaptive input coefficients are designed based on *Lyapunov's theory*. Next, the proposed control system uses *Lyapunov's theory* and investigates bounded to prove its stability. The proposed controller can be easily designed through a step-by-step design sequence. The performance of the proposed controller is verified by

comparing it with the previous controller through computer simulation.

This paper is structured as follows. Section II briefly introduces the snake robot and the snake robot's head system. In section III, Design BSC with RBF NN and apply the adaptive robust term. And prove the stability. In section IV, simulations were performed to compare the proposed controller and the previous controller. In section V, conclusions are based on simulation results.

II. SNAKE ROBOT SYSTEM

The snake robot to be dealt with is a 2-axis snake robot [13], [14], [15], [16], [17], [18], [19], [20], [21] with pitch and yaw rotation as shown in Figure. 2. The snake robot head system targets two modules adjacent to the robot head. When the snake robot is moving, the snake robot's head swings in the yaw and pitch directions by locomotion is to be compensated for the pitch and yaw rotation by using the head module.

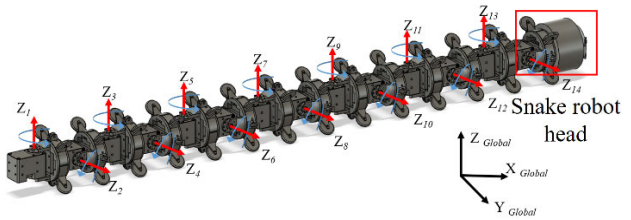


FIGURE 2. 2-axis snake robot.

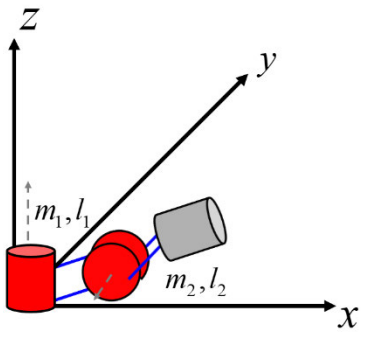


FIGURE 3. 2-axis snake robot head system.

The snake's robot head system is similar to the two-axis robot arm system as shown in Figure.3, so it is applied and used. This system is a MIMO(Multi-Input Multi-Output) system and it contains non-linearity. Using the *Lagrangian equation*, the system can be expressed as (1).

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) = \tau + \tau_{Disturbance} \tag{1}$$

$M(\theta)$ is 2×2 inertia matrix, $C(\theta, \dot{\theta})$ is 2×1 coriolis matrix, $G(\theta)$ is 2×1 gravity matrix, τ is 2×1 system input matrix, $\tau_{Disturbance}$ is unmeasurable disturbance. The several angles, angular velocity, and angular acceleration of module are $\theta = \begin{bmatrix}\theta_1 & \theta_2\end{bmatrix}^T$, $\dot{\theta} = \begin{bmatrix}\dot{\theta}_1 & \dot{\theta}_2\end{bmatrix}^T$, and $\ddot{\theta} = \begin{bmatrix}\ddot{\theta}_1 & \ddot{\theta}_2\end{bmatrix}^T$, respectively.

The system expressed in (1) is expressed as a strict feedback form as follows (2) $\sim$ (6).

$$x_1 = \theta \tag{2}$$
$$\dot{x}_1 = x_2 = \dot{\theta} \tag{3}$$
$$\ddot{x}_1 = \dot{x}_2 = \ddot{\theta} \tag{4}$$
$$\dot{x}_2 = -M^{-1}(C + G) + M^{-1}u + D \tag{5}$$
$$y = x_1 = \theta \tag{6}$$

The control input and output are $u = [\tau_1 \tau_2]^T$ and $y = \begin{bmatrix}\theta_1 & \theta_2\end{bmatrix}^T$, respectively. D is an unmeasurable disturbance.

III. CONTROLLER DESIGN

A. BACK-STEPPING CONTROLLER DESIGN

The main objective of this section is to design a BSC.

For the controller design, we define the error as the (7) and (8) [47], [48].

$$e_1 = x_1 - x_{1,d} \tag{7}$$
$$e_2 = x_2 - x_{2,d} \tag{8}$$
$$\dot{e}_1 = \dot{x}_1 - \dot{x}_{1,d} \tag{9}$$

$x_{1,d}$ and $x_{2,d}$ are desired angles of x_1, x_2, respectively. Equation (9) is the derivative (7). considering the Lyapunov function candidate positive definite

$$V_1 = e_1^2/2 \tag{10}$$

Its time derivative is

$$\dot{V}_1 = e_1\dot{e}_1 = e_1(x_2 - \dot{x}_{1,d}) \tag{11}$$

For the system to be stable, the Lyapunov stability $\dot{V}_1 < 0$ must be satisfied. Therefore, $\dot{x}_{1,d}$ is defined as

$$\dot{x}_{1,d} = x_2 + \varsigma = x_2 + k_1 e_1 \tag{12}$$

where k_1 is positive control gain and $\varsigma = k_1 e_1$ is virtual input. So, $x_{2,d}$ is

$$x_{2,d} = \dot{x}_{1,d} - k_1 e_1 \tag{13}$$

Using (13), (8), and (9) can be expressed as follows:

$$\dot{e}_1 = x_2 - x_{2,d} - k_1 e_1 = e_2 - k_1 e_1 \tag{14}$$
$$e_2 = x_2 - \dot{x}_{1,d} + k_1 e_1 \tag{15}$$
$$\dot{e}_2 = \dot{x}_2 - \ddot{x}_{1,d} + k_1 \dot{e}_1 \tag{16}$$
$$\dot{e}_2 = -M^{-1}(C + G) + M^{-1}u + D - \ddot{x}_{1,d} + k_1\dot{e}_1 \tag{17}$$

Next, considering the Lyapunov function candidate positive definite

$$V_2 = V_1 + e_2^2/2 \tag{18}$$

Its time derivative is

$$\dot{V}_2 = e_1\dot{e}_1 + e_2\dot{e}_2 = e_1(e_2 - k_2 e_1) + e_2(-M^{-1}(C + G) + M^{-1}u + D - \ddot{x}_{1,d} + k_1\dot{e}_1) \tag{19}$$

To realize Lyapunov stability $\dot{V}_2 < 0$, we can get control input as

$$u = (C + G) + M(-D + \ddot{x}_{1,d} - k_1\dot{e}_1 - k_2e_2 - e_1) \quad (20)$$

Equation (20) is the BSC control input. D includes unmeasurable. This paper D is tracked and compensated using an RBF neural network.

B. RBF NEURAL NETWORK DESIGN

The RBF neural network has three layers: the input layer, the hidden layer, and the output layer. So, the RBF network has a simple network structure and good generalization ability to avoid unnecessary long computations compared to multi-layer networks. In general, the greater the number of nodes in the hidden layer in a neural network, the richer the approximation can be, but there is a disadvantage in that the computation speed increases. In machine control systems, computational speed is very important. Therefore, the RBF network is suitable for application to mechanical control systems [37], [38].

In this paper, to minimize the network input and nodes of the RBF neural network, we use an error-based input and a network of 2-5-1 structure as shown in Figure. 4, respectively.

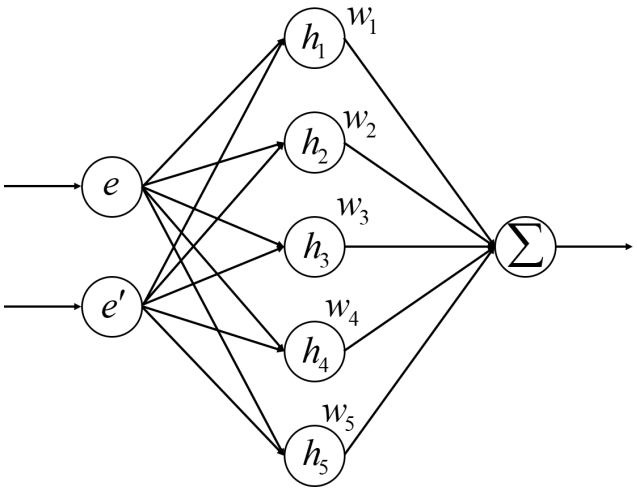


FIGURE 4. 2-5-1 RBF neural network structure.

Define the neural network output for integration with the BSC.

$$Y(Z) = \sum W^T H(Z) \quad (21)$$

$Z = [e_1\ e_2]^T$ is the input vector, $W = [w_1 \dots w_5]^T$ is the weight vector, $H = [h_1 \dots h_5]^T$ is the activation function. The activation function is expressed as a Gaussian function as follows.

$$h_{ij} = \exp(-\left\|x_i - c_j\right\|^2 / 2b_j^2) \quad (22)$$

x_i is the input vector, c_j represents the coordinate value of the center point of the Gaussian function. b_j represents the width of the Gaussian function.

Next, approximate the disturbance D [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], we can get

$$D = W^{*T} H(Z) + \varepsilon \quad (23)$$

W^* is optimal weight vector and ε is 2×1 network approximation error [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48]. W^* is generally unknown and is found through learning. The estimated D is expressed as follows:

$$\hat{D} = \hat{W}^T H(Z) \quad (24)$$

where $\hat{D}$ is network estimation disturbance and $\hat{W}$ is weight estimation vector. Next, add the network output to the BSC input.

$$u = (C + G) + M(-\hat{D} + \ddot{x}_{1,d} - k_1\dot{e}_1 - k_2e_2 - e_1) \quad (25)$$

Substitute (24) and (25) into Equation (5).

$$\dot{x}_2 = \ddot{x}_{1,d} - k_1\dot{e}_1 - k_2e_2 - e_1 + (W^{*T} - \hat{W}^T)H(Z) + \varepsilon \quad (26)$$

Through (16) and (26), $\dot{e}_2$ can be expressed as

$$\dot{e}_2 = -k_2e_2 - e_1 - \tilde{W}^T H(Z) + \varepsilon \quad (27)$$

$\tilde{W} = \hat{W} - W^*$ is weight estimation error vector.

Next, considering the Lyapunov function candidate positive definite as follows:

$$V_3 = V_2 + \tilde{W}^T\tilde{W}/2\gamma \quad (28)$$

Its time derivative is

$$\dot{V}_3 = e_1\dot{e}_1 + e_2\dot{e}_2 + \tilde{W}^T\dot{\tilde{W}}/\gamma \quad (29)$$

Substitute (14) and (27) into (29).

$$\dot{V}_3 = -k_1e_1^2 - k_2e_2^2 + \tilde{W}^T(-e_2H(Z) + \dot{\hat{W}}/\gamma) + e_2\varepsilon \quad (30)$$

Assume that the network estimation error ε is very small, $\varepsilon \simeq 0$ is satisfied. To realize Lyapunov stability $\dot{V}_3 < 0$, we can get weight update law as follows:

$$\dot{\hat{W}} = \gamma(e_2H(Z) - \sigma\hat{W}) \quad (31)$$

where, γ and σ is positive control gain. Weight vector update by measuring the error after system behavior at every sampling time according to (31).

C. ADAPTIVE ROBUST TERM DESIGN

The network approximation error is an error that occurs during learning. It becomes very small when training is complete but may affect the transient state during learning. The network approximation error usually occurs when the signal changes rapidly or at the beginning of learning and is included in the overall system error. In the previous section, the weight update law was designed by *Lyapunov's theory* using the assumption that the network estimation error is very small. The network estimation error may affect control performance when the estimation error increases due to disturbance or a

sudden signal. To compensate for this, the mathematical proof is also not easy.

In previous research [34], [35], [38], [39], [40], [41], [42], [43], [44], [45], they used the robust term as $\psi \mathrm{sgn}(e)$. ψ is positive control gain, sgn() is sign function, and is an error. Therefore, it is an input to which the control gain is added or subtracted according to the sign of the error. If tuning well, this method is simple and effective. However, this method generates the same output even for instantaneous large signal changes, so the effect of improvement is small. It can be improved by setting the control input to a large, but it can be seen that the input vibrates in a steady state. Therefore, we need to generate a large output depending on the situation. In addition, it also requires an output that does not vibrate in a steady state. In this section, considering stability, we will design the adaptive control gain using the Lyapunov function candidate and design the adaptive robust term without vibration in a steady state using the reverse saturation function.

To approximate the network estimation error through the system error, expressed as (32).

$$\varepsilon \cong K^* \cdot s \tag{32}$$

K^* is optimal estimation error adaptive coefficient, $s = e + \lambda \dot{e}$, λ is control gain. As K changes, it responds to the network estimation error that changes depending on the various situation.

$$\hat{\varepsilon} = \hat{K} \cdot s \tag{33}$$

$\hat{\varepsilon}$ is an estimation adaptive robust term. And $\hat{K}$ is the estimation error adaptive coefficient. Substitute (33) into (25).

$$u = (C + G) + M(-\hat{D} + \ddot{x}_{1,d} - k_1\dot{e}_1 - k_2 e_2 - e_1 - \hat{\varepsilon}) \tag{34}$$

Substitute (34) into (5).

$$\dot{x}_2 = \ddot{x}_{1,d} - k_1\dot{e}_1 - k_2 e_2 - e_1 - \tilde{W}^T H(Z) - \tilde{\varepsilon} \tag{35}$$

The error of the network estimation error is expressed as $\tilde{\varepsilon} = \hat{\varepsilon} - \varepsilon^*$. Substitute (34) into (16) to use the adaptive robust term in the proposal controller design.

$$\dot{e}_2 = -k_2 e_2 - e_1 - \tilde{W}^T H(Z) - \tilde{\varepsilon} \tag{36}$$

Next, we consider the Lyapunov function candidate as

$$V_4 = V_3 + \tilde{\varepsilon}^T \tilde{\varepsilon}/2\alpha \tag{37}$$

Its time derivative is

$$\dot{V}_4 = e_1\dot{e}_1 + e_2\dot{e}_2 + \tilde{W}^T \dot{\hat{W}}/\gamma + \tilde{\varepsilon}^T \dot{\tilde{\varepsilon}}/\alpha \tag{38}$$

Substitute (14) and (36) into (38).

$$\begin{aligned} \dot{V}_4 = & -k_1 e_1^2 - k_2 e_2^2 \\ & + \tilde{W}^T(-e_2 H(z) + \hat{W}'/\gamma) + \tilde{\varepsilon}(-e_2 + \dot{\hat{\varepsilon}}/\alpha) \end{aligned} \tag{39}$$

Applying the weight update laws devised in Equation (31), To realize Lyapunov stability $\dot{V}_4 < 0$, we can get update law.

$$\dot{\hat{\varepsilon}} = \alpha e_2 \tag{40}$$

Substituting (40) and (31) into (39), $\dot{V}_4 < 0$ is satisfied as follows

$$\dot{V}_4 = -k_1 e_1^2 - k_2 e_2^2 - \sigma \tilde{W}^T \hat{W} \tag{41}$$

In order to obtain the estimation error adaptive coefficient, differentiating (33) and substituting it into (40), the following equation can be obtained.

$$\hat{K} = \alpha e_2 / \dot{s} \tag{42}$$

where α is the control gain of the adaptive robust term.

The configured control input can improve control performance by compensating for the network approximation error, but when the neural network is trained, it can affect the steady state with an additional control input. Therefore, it is improved by using a reverse saturation function such as (43).

$$sat_{rev}(u) = \begin{cases} -|u| & \text{if: } u < -N \\ 0 & \text{if: } -N \le u \le N \\ +|u| & \text{if: } u > N \end{cases} \tag{43}$$

N is a positive threshold of reverse saturation function. The suggested control input is as follows:

$$\begin{aligned} \mathrm{u} = & (\mathrm{C} + \mathrm{G}) + \mathrm{M}(\ddot{\mathrm{x}}_{1,\mathrm{d}} - \mathrm{k}\dot{\mathrm{e}}_1 - \mathrm{k}_2\mathrm{e}_2 - \mathrm{e}_1 \\ & - \underbrace{\hat{\mathrm{W}}^{\mathrm{T}}\mathrm{H}(\mathrm{Z})}_{\text{RBF NN output}} - \underbrace{\mathrm{sat}_{\mathrm{rev}}(\hat{\varepsilon})}_{\text{adaptive robust term}}) \end{aligned} \tag{44}$$

D. STABILITY ANALYSIS

This section analyzes the stability of the proposal controller devised in the previous section. The proposed controller uses the reverse saturation function to improve the control input in the steady state. When the adaptive robust term control input is larger than the threshold value, stability is guaranteed according to the Lyapunov function candidate used in the adaptive robust term design. When the adaptive robust term control input is below the threshold value, the function does not generate output at a steady state. Therefore, stability can be guaranteed for the same reason as the previous RBF NN controller. In this section, we prove the system stability by investigating bounded when the adaptive robust term control input is below than threshold value.

As mentioned earlier, when the adaptive robust term control input is larger than the threshold value, the system is stable due to the Lyapunov function candidate considered in the adaptive robust term design. When the adaptive robust term control input is below than threshold value, the adaptive robust term control input becomes zero due to the reverse saturation function, and the network approximation error cannot be compensated. so it is assumed that there is an estimation error in the system. Therefore, it is assumed that a network approximation error exists in the control system.

To prove stability when adaptive robust term input is below than threshold value, we choose the Lyapunov function candidate (28) and substitute (31) into (30) [47], [48].

$$\begin{aligned} V &= e_1^2/2 + e_2^2/2 + \tilde{W}^T \tilde{W}/2\gamma \\ \dot{V} &= -k_1 e_1^2 - k_2 e_2^2 + e_2\varepsilon - \sigma \tilde{W}^T \hat{W} \end{aligned} \tag{45}$$

Equation (45) contains network approximation error. Next, using $\tilde{W} = \hat{W} - W^*$, rearrange (45) as follows

$$\begin{aligned}
\dot{V} &= -k_1 e_1^2 - k_2 e_2^2 + e_2\varepsilon - \sigma(\tilde{W}^T\hat{W} + \tilde{W}^T\tilde{W} + \tilde{W}^T W^*)/2 \\
&= -k_1 e_1^2 - k_2 e_2^2 + e_2\varepsilon \\
&\quad - \sigma(\tilde{W}^T\tilde{W} + (\hat{W}^T - W^{*T})\hat{W} + (\hat{W}^T - W^{*T})W^*)/2 \\
&= -k_1 e_1^2 - k_2 e_2^2 + e_2\varepsilon \\
&\quad - \sigma(\tilde{W}^T\tilde{W} + \hat{W}^T\hat{W} - W^{*T}W^*)/2 \\
&\le -k_1 e_1^2 - k_2 e_2^2 + e_2\varepsilon - \sigma(\tilde{W}^T\tilde{W} - W^{*T}W^*)/2 \\
&= -k_1 e_1^2 - k_2 e_2^2 - (e_2 - \varepsilon)^2/2 + (e_2^2 + \varepsilon^2)/2 \\
&\quad - \sigma(\tilde{W}^T\tilde{W} - W^{*T}W^*)/2 \\
&= -k_1 e_1^2 - (k_2 - 1/2)e_2^2 - (e_2 - \varepsilon)^2/2 + \varepsilon^2/2 \\
&\quad - \sigma(\tilde{W}^T\tilde{W} - W^{*T}W^*)/2 \\
&\le -k_1 e_1^2 - (k_2 - 1/2)e_2^2 + \varepsilon^2/2 \\
&\quad - \sigma(\tilde{W}^T\tilde{W} - W^{*T}W^*)/2
\end{aligned} \tag{46}$$

If $\kappa_{\min} > 0, k_1 \ge \kappa_{\min}/2, k_2 \ge (\kappa_{\min}+1)/2, \sigma \ge \kappa_{\min}/\gamma$, equation (46) can be written as follows

$$\begin{aligned}
\dot{V} &\le -k_1 e_1^2 - (k_2 - 1/2)e_2^2 + \varepsilon^2/2 \\
&\quad - \sigma(\tilde{W}^T\tilde{W} - W^{*T}W^*)/2 \\
&= -\kappa_{\min} e_1^2/2 - \kappa_{\min} e_2^2/2 - \kappa_{\min}\tilde{W}^T\tilde{W}/2\gamma \\
&\quad + \varepsilon^2/2 + \kappa_{\min} W^{*T}W^*/2\gamma \\
&= -\kappa_{\min} V + \varepsilon^2/2 + \kappa_{\min} W^{*T}W^*/2\gamma
\end{aligned} \tag{47}$$

Substituting $\varepsilon^2/2 + \kappa_{\min} W^{*T}W^*/2\gamma = \rho > 0$, equation (47) can be rewritten as follows

$$\dot{V} \le -\kappa_{\min} V + \rho \tag{48}$$

From the above (48), we can get

$$V(t) \le (V(t_0) - \rho/\kappa_{\min})e^{-\kappa_{\min}(t-t_0)} + \rho/\kappa_{\min}, \forall t \ge t_0 \tag{49}$$

V is bounded as $\kappa_{\min}/\rho$, and all system variables related to V are also bounded. So, the closed-loop system is bounded [46], [47], [48], [54], [55], [56]. Therefore, the system is stable.

IV. SIMULATIONS STUDIES

A. SIMULATION SETUP

To compare the performance of the proposal controller with the conventional controller, we verify it using the snake robot head system. The locomotion of the snake robot is realized by combining different phases of sinusoidal waves. Consequently, the snake robot's head also moves in a sinusoidal wave while driving. Therefore, the proposal controller is necessary to check the sinusoidal signal tracking performance. So control input used in the simulation, we use a continuous sinusoidal signal as shown in Figure. 5(a). In addition, to check the additional performance of the proposal controller a control input that is a mixture of continuous and abruptly changing signals is generated, such as a sudden operation due to an obstacle while driving or passing through a narrow pipe.

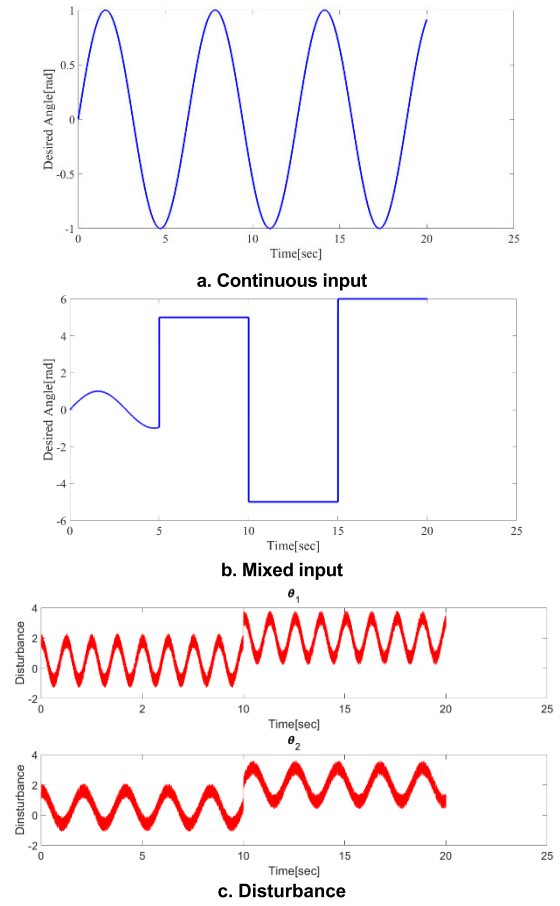


FIGURE 5. Simulation input and disturbance.

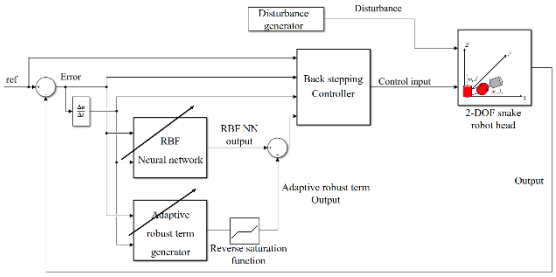
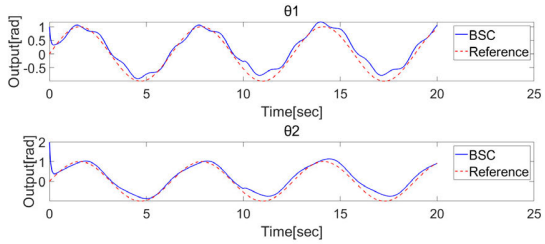


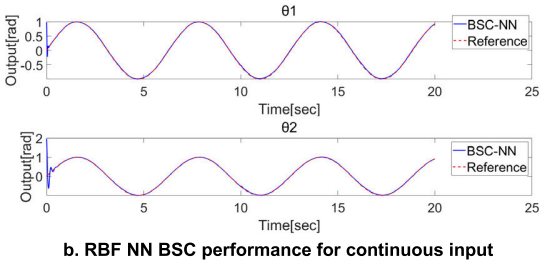
FIGURE 6. Proposed control system block diagram.

These control inputs are worst-case and do not correspond to actual operating conditions. and the control input is shown in the Figure. 5(b). The disturbance resembles the head motion of a snake robot as shown in Figure. 5(c) and uses a fast sinusoidal signal. We also assume that noise is included for unknown reasons, such as topographical conditions, and a large DC bias occurs within 10 seconds.

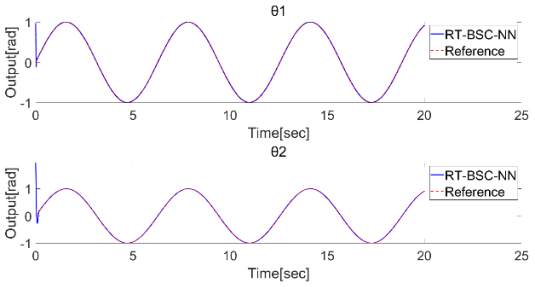
The 2-DOF snake robot's head system parameter is $m_1 = 0.1$ kg, $m_2 = 0.2$ kg, $l_{1,2} = 0.1$ m. The Snake robot's head system initial angle value is $\theta_1 = 1$ rad, $\theta_2 = 2$ rad, $\dot{\theta}_{1,2} = 0$ rad/s, $\ddot{\theta}_{1,2} = 0$ rad/s^2 BSC controller parameters are $k_1 = 5$, $k_2 = 2$, RBF neural network parameters are



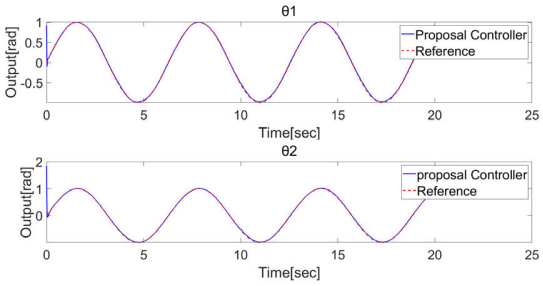
a. BSC performance for continuous input



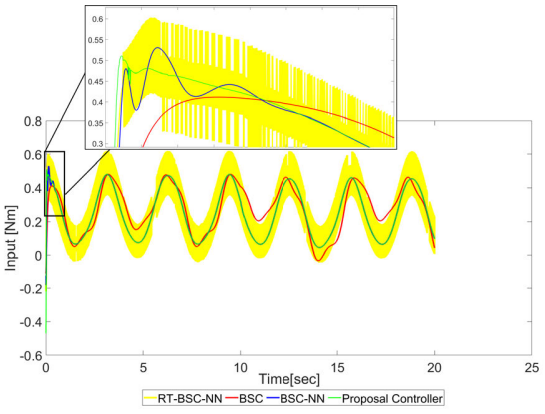
b. RBF NN BSC performance for continuous input



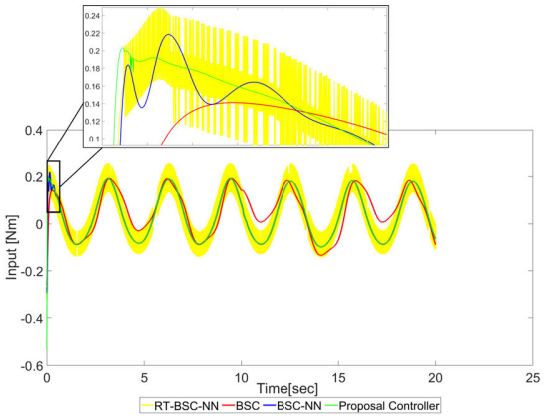
c. Previous controller performance for continuous input



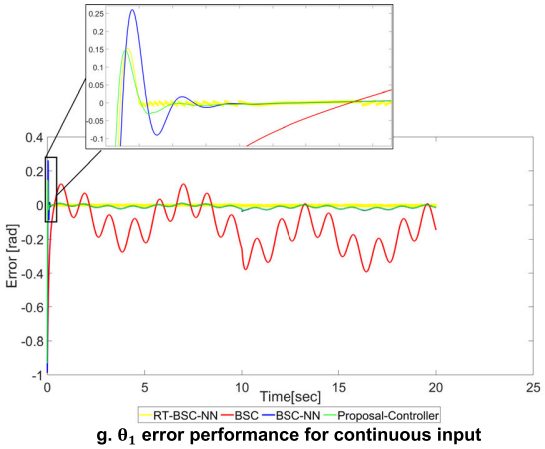
d. Proposal controller performance for continuous input



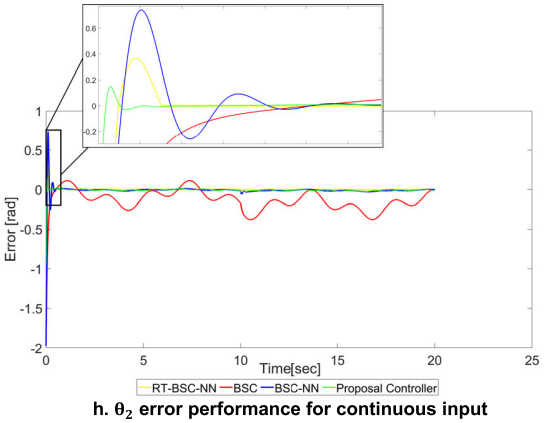
e. Controller input τ_1 performance for continuous input



f. Controller input τ_2 performance for continuous input



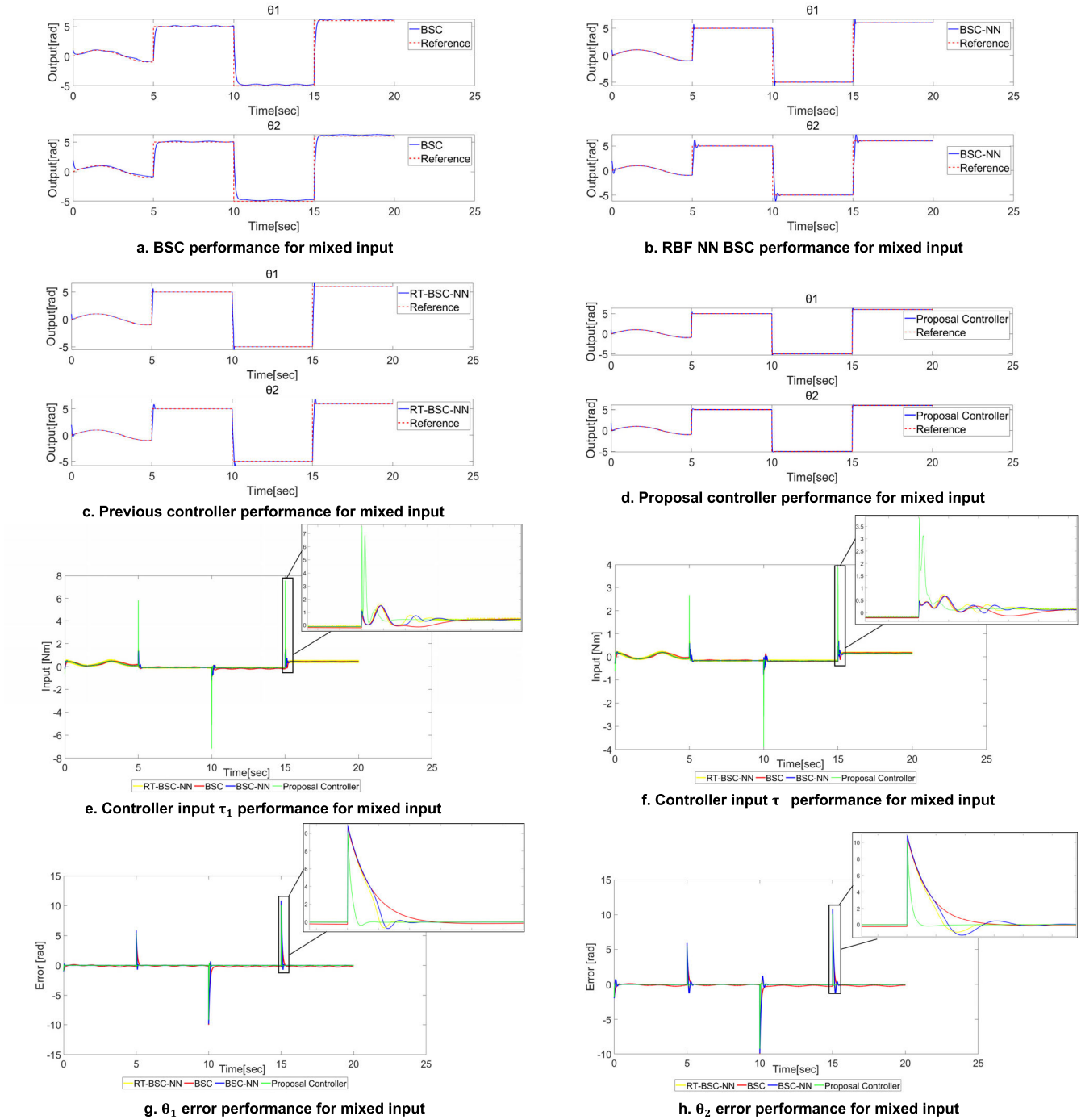
g. θ_1 error performance for continuous input



h. θ_2 error performance for continuous input

FIGURE 7. Controller performance comparison for continuous input.

$c = 0.5 \times [-1, \ -0.5, 0, 0.5, 1]$, $b = 10$. All initial weights are set to 0.1 on the assumption that offline learning is not performed. $\gamma = diag[500, 50, \ 500, 50]$, $\sigma = 0.1$, $\alpha = 15$, $N = 0.01$. The sampling time is 1 ms. In order to compare the proposal controller with the previous robust term, the previous robust term is $\psi \text{sgn}(s)$. s is the same as the proposal controller, and ψ is 10. The Control system block diagram is shown in Figure. 6. Sine all control parameters can affect the



a. BSC performance for mixed input

b. RBF NN BSC performance for mixed input

c. Previous controller performance for mixed input

d. Proposal controller performance for mixed input

e. Controller input τ_1 performance for mixed input

f. Controller input τ performance for mixed input

g. θ_1 error performance for mixed input

h. θ_2 error performance for mixed input

FIGURE 8. Controller performance comparison for mixed input.

control performance, the control parameter design sequence is designed step by step from BSC. All control parameters select through repeated simulation. While simulation, Back-stepping control call BSC, Back-stepping control with RBF NN call BSC-NN, Back-stepping control with RBF NN, and the previous robust term call RT-BSC-NN.

B. CONTINOUS INPUT SIMULATION RESULT

The simulation proceeds using the Figure. 5(a).

The Figure. 7 compares the proposal controller with the previous controller for continuous input. The BSC control in Figure. 7(a) has low tracking performance and steady-state errors due to disturbance. Figure. 7(b) is the output of BSC with RBF NN. This controller shows high tracking performance because it can compensate the disturbance estimation through neural network. However, it can be seen that an overshoot occurs due to a network approximation error in the weight update process at the beginning of the controller

operation. Figure. 7(c) shows the result of RBF NN BSC with robust term in the previous research. It can be seen that this controller can greatly reduce the overshoot occurring in the weight update process at the beginning of the controller operation. This shows that transient state can be improved. Also tracking performance in steady state is satisfactory. Figure. 7(d) shows the result of proposal controller. The proposal controller can improve the transient state more in Figure. 7(c) controller. And tracking performance in steady state is also good. Next, Figure. 7(e) and 7(f) show a comparison of control inputs. As common in Figure. 7(e) and 7(f), previous robust term controller produces tracking performance similar to that of the proposed controller, but it can be seen that large chattering occurs in the control input. The proposal controller produces a bigger control force as the error increases, improving the transient state. The control input generates a signal without chattering due to the reverse saturation function. Looking at Figures 7(g) and 7(h), which compare the errors, we can see that the proposal controller converges to zero faster than the previous controller. A comparative analysis of the input and the error shows that the adaptive robust term can improve more greatly. Furthermore, we have confirmed that the reverse saturation function does not generate unnecessary inputs in a steady state.

C. MIXED INPUT SIMULATION RESULT

The simulation proceeds using the Figure. 5(b).

Figure. 8 compares the proposed controller with the previous controller for mixed inputs. Figure. 8(a) and 8(b) are shown Similar to previous simulation results, adding RBF-NN to BSC could improve steady-state performance, but overshoot occurs on average at about 14% and 25% in abruptly changing signals intervals. Figure. 8(c) shows the result of RBF NN BSC with the robust term in the previous research. The controller had overshoots of 10% and 15%, but the transient state was improved compared to the BSC and BSC with RBF NN. As shown in Figure. 8(d), applying the proposed controller can improve the overshoot respectively by an average of 5% and 2%, and faster. As shown in Figure. 8(e) and 8(f), even a change in the input produces a large control force to improve the transient state when the control signal changes abruptly, as in the previous simulation results. Error compensation is shown in Figure. 8(g) and 8(h). The proposal controller quickly converges to zero by generating a large control force when a large error occurs due to the adaptive coefficient. And the reverse saturation function eliminates unnecessary inputs to maintain a steady state.

V. CONCLUSION

In this paper, we propose an error-based adaptive robust RBF-NN-BSC to the snake robot head system for effective image data reading when the snake robot is driving. To accommodate unmeasurable disturbances, we designed an RBF-NN BSC with minimal nodes that take the error as input. Based on *Lyapunov's theory*, we designed the weight update law and applied adaptive robust term. In order to compensate for the network approximation error caused by the initial weight update or large signal changes, the network approximation error is approximated by the error function and applying the estimation error adaptive coefficient. The proposal controller stability is proven through *Lyapunov's theory* and investigating bounded.

The proposal controller was compared and verified with the previous controller by simulation with disturbances using both continuous inputs such as the snake robot's head movements caused by the snake robot driving, and control inputs with large signal changes. As a result of the simulation, we improved the tracking performance of the BSC via RBF-NN. However, when the initial weight update, overshoot occurs due to the approximation error, and the application of the proposal controller improves the overshoot and confirms better tracking performance. In addition, in order to compensate for the control input chattering that occurs in the previous robust term, when a steady state is reached, additional control inputs are not used compared to BSCNN due to the reverse saturation function. In the future, we plan to apply the proposed controller to an actual snake robot system and verify its performance.

REFERENCES

[1] J. Lim, W. Yang, Y. Shen, and J. Yi, "Analysis and validation of serpentine locomotion dynamics of a wheeled snake robot moving on varied sloped environments," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Boston, MA, USA, Jul. 2020, pp. 1069–1074.

[2] Y. A. Baysal and I. H. Altas, "Modelling and simulation of a wheelless snake robot," in *Proc. 7th Int. Conf. Electr. Electron. Eng. (ICEEE)*, Antalya, Turkey, Apr. 2020, pp. 285–289.

[3] A. Crespi and A. J. Ijspeert, "Online optimization of swimming and crawling in an amphibious snake robot," *IEEE Trans. Robot.*, vol. 24, no. 1, pp. 75–87, Feb. 2008.

[4] S. Yu, S. Ma, B. Li, and Y. Wang, "Locomotion control and gaits' modality of a 3D snake-like robot," in *Proc. SICE Annu. Conf.*, Chofu, Japan, Aug. 2008, pp. 3026–3030.

[5] S. Yu, S. Ma, B. Li, and Y. Wang, "An amphibious snake-like robot with terrestrial and aquatic gaits," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 2011, pp. 2960–2961.

[6] Y. Zhou, Y. Zhang, F. Ni, and H. Liu, "A spring-like pipe climbing gait for the snake robot," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Macau, Macao, Dec. 2017, pp. 1886–1891.

[7] W. Zhen, C. Gong, and H. Choset, "Modeling rolling gaits of a snake robot," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Seattle, WA, USA, May 2015, pp. 3741–3746.

[8] J. Pyo, M. Lee, D.-G. Shin, K.-H. Seo, H. Joe, J.-H. Suh, and M. Jin, "Generation of snake robot locomotion patterns using genetic algorithm," *J. Korean Soc. Precis. Eng.*, vol. 38, no. 10, pp. 717–724, Oct. 2021.

[9] C. Gong, R. L. Hatton, and H. Choset, "Conical sidewinding," in *Proc. IEEE Int. Conf. Robot. Autom.*, Saint Paul, MN, USA, May 2012, pp. 4222–4227.

[10] J. Dai, M. Travers, T. Dear, C. Gong, H. C. Astley, D. I. Goldman, and H. Choset, "Robot-inspired biology: The compound-wave control template," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Seattle, WA, USA, May 2015, pp. 5879–5884.

[11] J. Mukherjee, S. Mukherjee, and I. N. Kar, "Sliding mode control of planar snake robot with uncertainty using virtual holonomic constraints," *IEEE Robot. Automat. Lett.*, vol. 2, no. 2, pp. 1077–1084, Apr. 2017.

[12] A. M. Kohl, K. Y. Pettersen, E. Kelasidi, and J. T. Gravdahl, "Planar path following of underwater snake robots in the presence of ocean currents," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 383–390, Jan. 2016.

[13] Z. Zhou, H. Wang, D. Li, and H. Deng, "Motion control curve of snake-like robot based on centroid stability," in *Proc. IEEE Int. Conf. Unmanned Syst. (ICUS)*, Beijing, China, Oct. 2019, pp. 826–830.

[14] P. Liljebäck, K. Y. Pettersen, Ø. Stavdahl, and J. T. Gravdahl, "A review on modelling, implementation, and control of snake robots," *Robot. Auto. Syst.*, vol. 60, no. 1, pp. 29–40, Jan. 2012.

[15] R. L. Hatton and H. Choset, "Generating gaits for snake robots by annealed chain fitting and keyframe wave extraction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, St. Louis, MO, USA, Oct. 2009, pp. 840–845.

[16] R. Ariizumi and F. Matsuno, "Dynamic analysis of three snake robot gaits," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1075–1087, Oct. 2017.

[17] M. Malayjerdi and A. Akbarzadeh, "Analytical modeling of a 3-D snake robot based on sidewinding locomotion," *Int. J. Dyn. Control*, vol. 7, no. 1, pp. 83–93, Mar. 2019.

[18] A. A. Transeth, K. Y. Pettersen, and P. Liljebäck, "A survey on snake robot modeling and locomotion," *Robotica*, vol. 27, no. 7, pp. 999–1015, Dec. 2009.

[19] S. Hirose and M. Mori, "Biologically inspired snake-like robots," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Shenyang, China, Aug. 2005, pp. 1–7.

[20] J. Liu, Y. Tong, and J. Liu, "Review of snake robots in constrained environments," *Robot. Auto. Syst.*, vol. 141, Jul. 2021, Art. no. 103785.

[21] S. Kim, D.-G. Shin, J. Pyo, J. Shin, M. Jin, and J. Suh, "A multi-sensor module of snake robot for searching survivors in narrow space," *J. Korea Robot. Soc.*, vol. 16, no. 4, pp. 291–298, Dec. 2021.

[22] J. Windau and L. Itti, "Multilayer real-time video image stabilization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Francisco, CA, USA, Sep. 2011, pp. 2397–2402.

[23] N. Sergiienko and L. Chen, "Adaptive head stabilization system for a snake-like robot," in *Proc. Int. Conf. Robot. Autom.*, Melbourne, VIC, Australia, 2014, pp. 1–7.

[24] G. Qiao, G. Song, Y. Zhang, J. Zhang, and Y. Li, "Head stabilization control for snake-like robots during lateral undulating locomotion," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Bali, Indonesia, Dec. 2014, pp. 392–397.

[25] Z. Bing, L. Cheng, K. Huang, Z. Jiang, G. Chen, F. Rohrbein, and A. Knoll, "Towards autonomous locomotion: Slithering gait design of a snake-like robot for target observation and tracking," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Vancouver, BC, Canada, Sep. 2017, pp. 2698–2703.

[26] S. Gay, A. Ijspeert, and J. S. Victor, "Predictive gaze stabilization during periodic locomotion based on adaptive frequency oscillators," in *Proc. IEEE Int. Conf. Robot. Autom.*, Saint Paul, MN, USA, May 2012, pp. 271–278.

[27] C. Au and P. Jin, "Investigation of serpentine gait of a snake robot with a wireless camera," in *Proc. 12th IEEE/ASME Int. Conf. Mech. Embedded Syst. Appl. (MESA)*, Auckland, New Zealand, Aug. 2016, pp. 1–6.

[28] M. Mutlu, K. Melo, M. Vespignani, A. Bernardino, and A. J. Ijspeert, "Where to place cameras on a snake robot: Focus on camera trajectory and motion blur," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot. (SSRR)*, West Lafayette, IN, USA, Oct. 2015, pp. 1–8.

[29] J. Florez, F. Calderon, and C. Parra, "Video stabilization taken with a snake robot," in *Proc. Symp. Signals, Images Artif. Vis.*, Bogota, Colombia, Sep. 2013, pp. 1–5.

[30] S. Hasanzadeh and A. A. Tootoonchi, "Ground adaptive and optimized locomotion of snake robot moving with a novel gait," *Auton. Robots*, vol. 28, pp. 457–470, Mar. 2010.

[31] H. Yamada, M. Mori, and S. Hirose, "Stabilization of the head of an undulating snake-like robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Diego, CA, USA, Oct. 2007, pp. 3566–3571.

[32] S. S. Ge and C. Wang, "Direct adaptive NN control of a class of nonlinear systems," *IEEE Trans. Neural Netw.*, vol. 13, no. 1, pp. 214–221, Jan. 2002.

[33] N. Cai, M. He, Q. Wu, and M. J. Khan, "On almost controllability of dynamical complex networks with noises," *J. Syst. Sci. Complex.*, vol. 32, no. 4, pp. 1125–1139, 2019.

[34] H. Lee and M. Tomizuka, "Robust adaptive control using a universal approximator for SISO nonlinear systems," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 1, pp. 95–106, Feb. 2000.

[35] S. Han, H. Wang, Y. Tian, and N. Christov, "Time-delay estimation based computed torque control with robust adaptive RBF neural network compensator for a rehabilitation exoskeleton," *ISA Trans.*, vol. 97, pp. 171–181, Feb. 2020.

[36] Y. Tian, W. Feng, M. Ouyang, H. Bian, and Q. Chen, "A positioning error compensation method for multiple degrees of freedom robot arm based on the measured and target position error," *Adv. Mech. Eng.*, vol. 14, no. 5, pp. 1–13, May 2022.

[37] S. Slama, A. Errachdi, and M. Benrejeb, "Neural adaptive PID and neural indirect adaptive control switch controller for nonlinear MIMO systems," *Math. Problems Eng.*, vol. 2019, pp. 1–11, Aug. 2019.

[38] J. Liu, *Radial Basis Function (RBF) Neural Network Control for Mechanical Systems: Design, Analysis and MATLAB Simulation*. Cham, Switzerland: Springer, 2013, pp. 1–280.

[39] J. Liu, *Intelligent Control Design and MATLAB Simulation*. Cham, Switzerland: Springer, 2018, pp. 113–233.

[40] Q. Song and L. Yin, "Robust adaptive fault accommodation for a robot system using a radial basis function neural network," *Int. J. Syst. Sci.*, vol. 32, no. 2, pp. 195–204, Jan. 2001.

[41] A.-K. Seghouane and N. Shokouhi, "Adaptive learning for robust radial basis function networks," *IEEE Trans. Cybern.*, vol. 51, no. 5, pp. 2847–2856, May 2021.

[42] H. Hu, Y. Ren, and H. Liu, "Design of neural network-based backstepping controller for the folding-boom aerial platform vehicle," in *Proc. Int. Conf. Mechatronics, Electron., Ind. Control Eng.*, Shenyang, China, 2015, pp. 265–268.

[43] S.-W. Kang, H. Bang, and S.-R. Lee, "Adaptive backstepping radial basis function neural network controller design for a Mars lander during the powered descent phase," *Proc. Inst. Mech. Eng., G, J. Aerosp. Eng.*, vol. 232, no. 11, pp. 2091–2107, Sep. 2018.

[44] X. Cao, P. Shi, Z. Li, and M. Liu, "Neural-network-based adaptive backstepping control with application to spacecraft attitude regulation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4303–4313, Sep. 2018.

[45] Y. Zhang, P.-Y. Peng, and Z.-P. Jiang, "Stable neural controller design for unknown nonlinear systems using backstepping," *IEEE Trans. Neural Netw.*, vol. 11, no. 6, pp. 1347–1360, Nov. 2000.

[46] W.-Y. Wang, M.-F. Kuo, T.-T. Lee, C.-M. Hong, and Y.-G. Leu, "RBF neural network adaptive backstepping controllers for MIMO nonaffine nonlinear systems," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, San Antonio, TX, USA, Oct. 2009, pp. 4946–4951.

[47] J. M. Zhu, Z. Q. Cao, T. P. Zhang, Y. Q. Yang, and Y. Yi, "Sufficient condition for the existence of the compact set in the RBF neural network control," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 3277–3282, Jul. 2018.

[48] H.-W. Kim, J.-H. Park, and S.-H. Park, "Design of error state-based RBF neural network adaptive backstepping controller for estimation and compensation of various unknown errors," *J. Inst. Control, Robot. Syst.*, vol. 24, no. 6, pp. 473–481, Jun. 2018.

[49] H. W. Kim, Y. H. Yoon, J. H. Jeong, and J. H. Park, "The adaptive backstepping controller of RBF neural network which is designed on the basis of the error," *J. Korean Soc. Precis. Eng.*, vol. 34, no. 2, pp. 125–131, Feb. 2017.

[50] G. Li, Y. Liu, Y. Li, and X. Bu, "Adaptive back-stepping control of high-order uncertain nonlinear systems that a funnel control scheme with uncertain dynamics," in *Proc. Int. Conf. Electr. Eng. Control Technol. (CEECT)*, Dec. 2020, pp. 1–8.

[51] X. Bu, B. Jiang, and H. Lei, "Performance guaranteed finite-time non-affine control of waverider vehicles without function-approximation," *IEEE Trans. Intell. Transp. Syst.*, early access, Nov. 29, 2022, doi: 10.1109/TITS.2022.3224424.

[52] X. Bu, Q. Qi, and B. Jiang, "A simplified finite-time fuzzy neural controller with prescribed performance applied to waverider aircraft," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 7, pp. 2529–2537, Jul. 2022.

[53] X. Bu, B. Jiang, and H. Lei, "Low-complexity fuzzy neural control of constrained waverider vehicles via fragility-free prescribed performance approach," *IEEE Trans. Fuzzy Syst.*, early access, Oct. 26, 2022, doi: 10.1109/TFUZZ.2022.3217378.

[54] X. Bu, B. Jiang, and H. Lei, "Nonfragile quantitative prescribed performance control of waverider vehicles with actuator saturation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 4, pp. 3538–3548, Aug. 2022.

[55] X. Yali and J. Changsheng, "Trajectory linearization control of an aerospace vehicle based on RBF neural network," *J. Syst. Eng. Electron.*, vol. 19, no. 4, pp. 799–805, Aug. 2008.

[56] Y. Fang, J. Fei, and K. Ma, "Model reference adaptive sliding mode control using RBF neural network for active power filter," *Int. J. Elect. Power Energy Syst.*, vol. 73, pp. 249–258, Dec. 2015.

[57] Q. Liu, D. Li, S. S. Ge, R. Ji, Z. Ouyang, and K. P. Tee, "Adaptive bias RBF neural network control for a robotic manipulator," *Neurocomputing*, vol. 447, pp. 213–223, Aug. 2021.

SUNG-JAE KIM received the B.S. degree in mechanical system engineering and the master's degree in robotics from Pukyong National University, Busan, South Korea, in 2020, where he is currently pursuing the Ph.D. degree with the Department of Intelligent Robot Engineering. His research interests include control engineering, robust control of nonlinear plants, intelligent control systems, disaster response robot, robot sensor application systems, mobile robot, and drone control systems.

MAOLIN JIN (Senior Member, IEEE) received the B.S. degree in material science and mechanical engineering from the Yanbian University of Science and Technology, Jilin, China, in 1999, and the M.S. and Ph.D. degrees in mechanical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2004 and 2008, respectively. He was a Postdoctoral Researcher with the Mechanical Engineering Research Institute, KAIST, in 2008. He was also a Senior Researcher with the Research Institute of Industrial Science and Technology, Pohang, South Korea, from November 2008 to February 2016. He is currently the Director and the Chief Researcher of the Human-Centered Robotics Center, KIRO, Pohang. His research interests include robust control of nonlinear plants, time-delay control, robot motion control, electro-hydraulic actuators, winding machines, disaster robotics, and factory automation. He serves as an Associate Editor for the *International Journal of Control, Automation, and Systems* (IJCAS), *Journal of Drive and Control*, and *Journal of the Korean Society for Precision Engineering*.

JIN-HO SUH (Senior Member, IEEE) received the B.S. degree in mathematics from Hanyang University, in 1993, the M.S. degree in mechanical, environment, and ocean engineering from Pukyong National University, South Korea, in 1998, and the Ph.D. degree in control engineering (robotics) from the Tokyo Institute of Technology, Tokyo, Japan, in 2002. He was a Postdoctoral Researcher with the National Research Laboratory (NRL), Dong-A University, from 2004 to 2006. From 2006 to 2018, he was the Director with the Disaster Robotics Research and Development Center, Korea Institute of Robotics and Technology Convergence (KIRO). He was also an Adjunct Professor of mechanical engineering with the Pohang University of Science and Technology (POSTECH), South Korea. Since 2018, he has been a Professor of mechanical system engineering with the Division of Energy Transport Systems Engineering, College of Engineering, Pukyong National University, South Korea. His research interests include disaster response robots, underwater robots, system control, and integration in field robotics.