A neural network combined with sliding mode controller for the two-wheel self-balancing robot

Duc-Minh Nguyen[1], Van-Tiem Nguyen[2], Trong-Thang Nguyen[3]
[1,3]Faculty of Electrical and Electronic Engineering, Thuyloi University, Vietnam
[2]Faculty of Electrical and Electronic Engineering, University of Transport and Communications, Vietnam

Article Info

Article history:

Received Dec 21, 2020
Revised May 4, 2021
Accepted May 20, 2021

Keywords:

Lyapunov theory
Neuron network
Sliding mode control
Two-wheeled self-balancing

ABSTRACT

This article presents the sliding control method combined with the self-adjusting neural network to compensate for noise to improve the control system's quality for the two-wheel self-balancing robot. Firstly, the dynamic equations of the two-wheel self-balancing robot built by Euler–Lagrange is the basis for offering control laws with a neural network of noise compensation. After disturbance-compensating, the sliding mode controller is applied to control quickly the two-wheel self-balancing robot reached the desired position. The stability of the proposed system is proved based on the Lyapunov theory. Finally, the simulation results will confirm the effectiveness and correctness of the control method suggested by the authors.

Corresponding Author:

Trong-Thang Nguyen
Faculty of Electrical and Electronic Engineering
Thuyloi University
175 Tay Son, Dong Da, Hanoi, Vietnam
Email: nguyentrongthang@tlu.edu.vn

1. INTRODUCTION

There has been much research on the two-wheel self-balancing (TWSB) robot in recent years because its application is expanding and developing [1]. Control models and algorithms are also optimized and become an open source for researchers. Especially, manufacturer Segway has also designed and produced a variety of TWSB vehicles for commerce.

There are many effective control methods available for the TWSB robot with nonlinear kinetics, such as the method using a proportional integral derivative (PID) controller [2], the optimal controller based on the linear–quadratic regulator (LQR) [3], [4], the robust-adaptive control [5], the backstepping control [6], [7], the sliding mode control [8], the fuzzy logic control structure [9] and the control system based on a neural network [10]. However, there is a lot of noise during outdoor operation that negatively affects the control system's quality. Therefore, to improve the control system's quality, the authors will propose a sliding control method combined with artificial neural networks (ANN) to eliminate the noise in this research.

In research studies [11]-[13], the authors have suggested two control methods for TWSB robots, such as PID and LQR. The authors made simulations and compared these methods, then conclude that LQR gives better results than PID. However, these studies did not mention disturbance components due to friction or uncertain interference from the environment. The simulation results also show that the control system's overshoot is relatively high, and the control quality at the start period has not achieved the desired results. Meanwhile, the study [14] compared four control methods: the pole-control, the PID control, the backstepping control, and the sliding mode control. The simulation results show that the backstepping method and the pole-control have better results. However, in controlling the angle of rotation, the extended noise still cannot be eliminated. With the proposal using fuzzy logic [15], [16], the authors used fuzzy laws to control the balance and position of the TWSB robot. The system has a fast response and limited interference. However, with variability and uncertainty disturbance, the ability to reject interference is still low. To overcome those disadvantage problems mentioned above, in this study, the authors will develop from experiences in building control systems and noise compensation in previous articles [17]-[19] for the construction of the controller of TWSB robot with a nonlinear dynamic structure and uncertainty disturbance. This study's main contributions include designing a sliding mode controller combined with adaptive neural networks with online self-aligning parameters to eliminate the noise and uncertain components in the TWSB robot with nonlinear kinetics and determination of the uncertainty parameters. The remaining content of this article includes as shown in: Section 2 presents the equations of control object for TWSB robot; in section 3, the authors build the neural network to compensate for the disturbance and prove the stability of the system; the results and discussion are presented in section 4, and the conclusions are presented in section 5.

2. BUILDING THE EQUATIONS OF CONTROL OBJECT FOR TWSB ROBOT

2.1. The model of the TWSB robot

The model of the TWSB robot is shown in Figure 1. The TWSB robot is a three-degrees-freedom robot with three joint variables: moving *x*, turning angle θ, and tilting variable φ.

The energy of the system: $L = K - U$

where K is the total kinetic energy and U is the total potential energy.

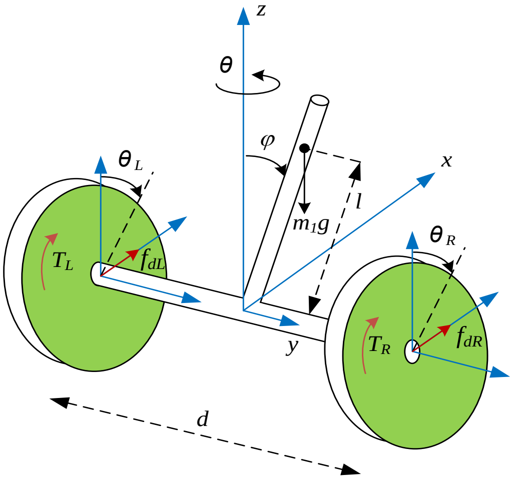


Figure 1. The model of the TWSB robot

2.2. Lagrange equations

The TWSB robot is an object with nonlinear kinetics. Therefore, we ignore external forces and the friction between the body and two-wheels to simplify the kinetic modeling process. These friction components and external forces will be added to the general equations and will be discussed in the next section. The total kinetic energy in the system is calculated as shown in:

$$K = K_1 + K_2 + K_3 + K_4 \tag{1}$$

where K_1 is the translational kinetic energy of the wheel:

$$K_1 = \frac{1}{2} m_{\mathrm{w}} (\dot{x}_L^2 + \dot{x}_R^2) \tag{2}$$

K_2 is the rotational kinetic energy of the wheel:

$$K_2 = \frac{1}{2}J_{\mathrm{wy}}(\dot{\theta}_L^2 + \dot{\theta}_R^2) \tag{3}$$

K₃ is the kinetic energy of the pedestal:

$$K_3 = \frac{1}{2}m_2\left(\frac{\dot{x}_L+\dot{x}_R}{2}\right)^2 + \frac{1}{2}J_c\dot{\varphi}^2 + \frac{1}{2}J_v\left(\frac{\dot{x}_L-\dot{x}_R}{d}\right)^2 \tag{4}$$

K_4 is the kinetic energy of the pendulum:

$$K_4 = \frac{1}{2}m_1\left[\left(\frac{\dot{x}_L+\dot{x}_R}{2}\right)^2 + l(\dot{x}_L + \dot{x}_R)\dot{\varphi}\cos(\varphi)\right] + \frac{1}{2}m_1 l^2\dot{\varphi}^2 + \frac{1}{2}J_p\dot{\varphi}^2 \tag{5}$$

where:
x_L, x_R are displacement of left wheel, right wheel along *x*-axis, respectively.
m_w is the mass of the wheel.
θ_L, θ_R are the rotation angle of the left wheel and the right wheel relative to the *z*-axis.
J_{wy} is the inertia moment of the wheel along *y*-axis.
m_1, m_2 are the mass of the pendulum, the mass of the pedestal, respectively.
J_c is the inertia moment of the pedestal around *y*-axis.
J_v is the inertia moment of the pedestal and the pendulum around the *z*-axis.
J_p is the inertia moment of the pendulum around the *y*-axis.
l is the distance from the base to the center of the pendulum.
φ is the tilt angle of the pendulum.

Since the potential of the wheel and the pedestal are zero, the total potential of the system is as shown in:

$$U = U_4 = mgl\cos(\varphi) \tag{6}$$

Where:

$$\begin{aligned} x_L &= \theta_L r \qquad \dot{x}_L = \dot{\theta}_L r \qquad \ddot{x}_L = \ddot{\theta}_L r \\ x_R &= \theta_R r \qquad \dot{x}_R = \dot{\theta}_R r \qquad \ddot{x}_R = \ddot{\theta}_R r \end{aligned} \tag{7}$$

Replacing (7) into (3) and replacing (2)–(6) into (1), we get the Lagrange equation of the system according to the total kinetic energy and the total potential energy as shown in:

$$\begin{aligned} L = &\left(m_{\mathrm{w}} + \frac{J_{wy}}{r^2}\right)\frac{\dot{x}_L^2+\dot{x}_R^2}{2} + \frac{m_1+m_2}{2}\left(\frac{\dot{x}_L+\dot{x}_R}{2}\right)^2 + m_1 l\left(\frac{\dot{x}_L+\dot{x}_R}{2}\right)\dot{\varphi}\cos(\varphi) \\ &+ \frac{1}{2}(m_1 l^2 + J_p + J_c)\dot{\varphi}^2 + \frac{1}{2}J_v\left(\frac{\dot{x}_L-\dot{x}_R}{d}\right)^2 - m_2 gl\cos(\varphi) \end{aligned} \tag{8}$$

2.3. The forward kinematic of TWSB robot

To calculate the forces in the system affected by the joint variables, we use the following general kinetic:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \tau_i \tag{9}$$

in which, τ_i is the forces effected by the joint variables. The variable q_i is a component of the joint variable vector. Applying in (9) with the displacement (x_L) of the left wheel along *x*-axis, we get:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}_L}\right) - \frac{\partial L}{\partial x_L} = \tau_L \tag{10}$$

With τ_L is the total moment of the system effected on the left wheel of the robot. Executing in (10) based on the (8), we obtain:

$$(m_{\mathrm{w}} + \frac{J_{wy}}{r^2})\ddot{x}_L + \frac{m_1+m_2}{4}(\ddot{x}_L + \ddot{x}_R) + \frac{1}{2}m_1 l(\ddot{\varphi}\cos\varphi - \dot{\varphi}^2\sin\varphi) + \frac{1}{d^2}J_v(\ddot{x}_L - \ddot{x}_R) = \tau_L \tag{11}$$

Applying in (9) with the displacement (x$_R$) of the right wheel along *x*-axis, we get:

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{x}_R}) - \frac{\partial L}{\partial x_R} = \tau_R \tag{12}$$

with τ_R is the total moment of the system effected on the right wheel of the robot. Executing in (12) based on the (8), we obtain:

$$(m_{\mathrm{w}} + \frac{J_{wy}}{r^2})\ddot{x}_R + \frac{m_1+m_2}{4}(\ddot{x}_L + \ddot{x}_R) + \frac{1}{2}m_1 l(\ddot{\varphi}\cos\varphi - \dot{\varphi}^2\sin\varphi) - \frac{1}{d^2}J_v(\ddot{x}_L - \ddot{x}_R) = \tau_R \tag{13}$$

Executing in (9) with the tilt angle variable (φ) of the pendulum, we obtain:

$$\frac{d}{dt}(\frac{\partial L}{\partial \dot{\varphi}}) - \frac{\partial L}{\partial \varphi} = 0 \tag{14}$$

Executing in (14) based on (8), we have:

$$m_1 l(\frac{\ddot{x}_L+\ddot{x}_R}{2})\cos\varphi + (m_1 l^2 + J_p + J_c)\ddot{\varphi} - m_2 g l\sin\varphi = 0 \tag{15}$$

We have relationships between the variables as shown in:

$$\begin{aligned} &x = \frac{x_L + x_R}{2} \qquad \dot{x} = \frac{\dot{x}_L + \dot{x}_R}{2} \qquad \ddot{x} = \frac{\ddot{x}_L + \ddot{x}_R}{2} \\ &\theta = \frac{x_L - x_R}{d} \qquad \dot{\theta} = \frac{\dot{x}_L - \dot{x}_R}{d} \qquad \ddot{\theta} = \frac{\ddot{x}_L - \ddot{x}_R}{d} \end{aligned} \tag{16}$$

Adding (11) and (13), then replacing variables as in (16), we get:

$$\left(2m_w + 2\frac{J_{wy}}{r^2} + \frac{m_1+m_2}{2}\right)\ddot{x} + m_1 l(\ddot{\varphi}\cos\varphi - \dot{\varphi}^2\, sin\,\varphi) = \tau_R + \tau_L \tag{17}$$

Taking (11) minus (13), then replacing variables as in (16), we get:

$$d(m_{\mathrm{w}} + \frac{J_{wy}}{r^2} + \frac{2}{d^2}J_v)\ddot{\theta} = \tau_L - \tau_R \tag{18}$$

2.4. The mathematical model of the control object

The general kinetic equation system for robots has the following general form [20], [21]:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F_r(\dot{q}) + \tau_d = \tau \tag{19}$$

where $D(q) \in R^{nxn}$ is the asymmetric and positive definite matrix $C(q,\dot{q}) \in R^n$ is the centrifugal and Coriolis forces.$F_r(\dot{q}) \in R^n$is the friction.$G(q) \in R^n$is the gravity force. $\tau_d \in R^n$is a general nonlinear disturbance.$\tau \in R^n$ represent the torque input controls vector. $q, \dot{q}, \ddot{q} \in R^n$ denotes the angle, velocity, and acceleration vector of the link, respectively. Because the TWSB robot is a three-degrees of freedom robot with three variables such as the moving variable *x*, the turning angle variable *θ*, and tilt angle variable *φ*, we have the vector variables of the joints:

$$q = [x \quad \theta \quad \varphi]^T \qquad \dot{q} = [\dot{x} \quad \dot{\theta} \quad \dot{\varphi}]^T \qquad \ddot{q} = [\ddot{x} \quad \ddot{\theta} \quad \ddot{\varphi}]^T$$

- Property 1. The inertia matrix D(q) is symmetric and uniformly positive definite, and satisfied

  $$\forall q \in R^n, \quad \underline{g}q^T q \le q^T D q \le \overline{g}q^T q$$

  where$\underline{g}$ and $\overline{g}$ are known positive constants.
- Property 2. The Coriolis and centrifugal matrix$C(q,\dot{q})$ can be appropriately defined in order for $(\dot{D} - 2C)$ is skew-symmetric. We get:

$$q^T(\dot{D} - 2C)q = 0, \qquad \forall q \neq 0. \tag{20}$$

Rewriting (17), (18), and (15) according to (19) we get the following coefficient matrices:

$$D = \begin{bmatrix} 2m_{\mathrm{w}} + 2\frac{J_{wy}}{r^2} + \frac{m_1+m_2}{2} & 0 & m_1 l\cos\varphi \\ 0 & d(m_{\mathrm{w}} + \frac{J_{wy}}{r^2} + \frac{2}{d^2}J_v) & 0 \\ m_1 l\cos\varphi & 0 & m_1 l^2 + J_p + J_c \end{bmatrix} \tag{21}$$

$$C = \begin{bmatrix} 0 & 0 & -m_1 l\dot{\varphi}\sin\varphi \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{22}$$

Demonstration of property 2**:** The derivative of matrix D:

$$\dot{D} = \begin{bmatrix} 0 & 0 & -m_1 l\dot{\varphi}\sin\varphi \\ 0 & 0 & 0 \\ -m_1 l\dot{\varphi}\sin\varphi & 0 & 0 \end{bmatrix} \tag{23}$$

Replacing (22) and (23) in (20), we get:

$$\begin{aligned} q^T(\dot{D} - 2C)q &= [x \quad \theta \quad \varphi]\left(\begin{bmatrix} 0 & 0 & -m_1 l\dot{\varphi}\sin\varphi \\ 0 & 0 & 0 \\ -m_1 l\dot{\varphi}\sin\varphi & 0 & 0 \end{bmatrix} - 2\begin{bmatrix} 0 & 0 & -m_1 l\dot{\varphi}\sin\varphi \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\right)\begin{pmatrix} x \\ \theta \\ \varphi \end{pmatrix} \\ &= [x \quad \theta \quad \varphi]\begin{bmatrix} 0 & 0 & m_1 l\dot{\varphi}\sin\varphi \\ 0 & 0 & 0 \\ -m_1 l\dot{\varphi}\sin\varphi & 0 & 0 \end{bmatrix}\begin{pmatrix} x \\ \theta \\ \varphi \end{pmatrix} \\ &= [-\varphi m_1 l\dot{\varphi}\sin\varphi \quad 0 \quad x m_1 l\dot{\varphi}\sin\varphi]\begin{pmatrix} x \\ \theta \\ \varphi \end{pmatrix} = 0 \end{aligned} \tag{24}$$

Thus, based on the results in (24) with the matrix parameters in (21) and (22), the TWSB robot kinetic equation has the property as (20).

3. BUILDING THE NEURAL NETWORK TO COMPENSATE FOR DISTURBANCE

Commonly, the sliding mode control method [22]-[24] includes defining the sliding surface as the system state functions and proving the trajectories of the closed-loop system reach this surface in a finite time based on stability theory. To solve the disadvantages of the common sliding mode control, we add a neural network. The neural network and sliding mode controller's learning algorithm is constructed relying on the Lyapunov theorem to ensure the system's stability.

We ignore some external forces that affected the system to build the kinetic model [25] of the robot. After making the general equations, we will add these external forces. In (19), $F_r(\dot{q})$ is the composition of friction on the two wheels. It includes F_L, F_R -friction force impact on the left-wheel and right-wheel;τ_d-the external force impacts the two wheels. τ_d has f_L , f_R - the interaction force between the left and right wheel and pedestal; f_{dL} , f_{dR} -the external force impact on the left wheel and right wheel. Setting:

$$F(q, \dot{q}, \ddot{q}, t) = F_r(\dot{q}) + \tau_d \tag{25}$$

In the nonlinear kinetic system, noise and friction are the leading causes of control quality loss. Therefore, to increase control quality, the authors will build a neural network for compensation for the disturbances and uncertainty parts in this study.

In studies on robust adaptive control, NN in [17], [26], [27] are mostly used for the unknown nonlinearities as approximation models because of their inherent capabilities of approximation. A simple artificial neural network structure for approximating function may be rewritten as:

$$F(q, \dot{q}, \ddot{q}, t) = F(s) = W\sigma + \varepsilon \tag{26}$$

where ε denotes the approximation error and $\bar{\varepsilon}$ is the limit of ε, ($|\varepsilon| \leq \bar{\varepsilon}$). σ denotes the Gaussian function. We define the weights w_{ji} to construct an approximated neural network, so the (9) could be rewritten as:

$$F(s) = \sum_{j=1}^{n} w_{ji}\,\sigma_i + \varepsilon \quad i = 1,2,\dots,n \tag{27}$$

The Gaussian function is as shown in:

$$\sigma_i = \exp\left(-\frac{s_i - c_i}{\lambda_i^2}\right) \tag{28}$$

where S_i= $[S_{i1}, S_{i2}, \dots, S_{in}]^T$ denotes the input of radial basis function (RBF) network, c_i=$[c_1, c_2, \dots, c_n]^T$ is the centers of the basic function and λ_i is the widths of the basic function, freely chosen. We set a Lyapunov function $V(t)$. Basing on the Lyapunov stability theory, if we make a control law so that the time derivative of $V(t)$ is negative, then the agent's trajectory will converge to the sliding surfaces$s = 0$ in a finite time and keep them on the sliding surface. Setting the Lyapunov function as shown in:

$$V(t) = \frac{1}{2}(s^T D s + \sum_{j=1}^{n} w_j^T w_j) \tag{29}$$

It is easy to see that this function is positive definite:

$$V(t) > 0, \forall s \neq 0$$

Differentiating in (29) concerning time, we get:

$$\begin{aligned} \dot{V}(t) &= \frac{1}{2}[\dot{s}^T D\mathrm{s} + s^T \dot{D} s + s^T D \dot{s} + \sum_{j=1}^{n} (\dot{w}_j^T w_j + w_j^T \dot{w}_j)] \\ &= \frac{1}{2} s^T \dot{D} s + s^T D \dot{s} + \sum_{j=1}^{n} w_j^T \dot{w}_j \end{aligned} \tag{30}$$

According to dynamic in (19), both $C(q, \dot{q})$ and $D(q)$ satisfy Property 2:

$$\begin{aligned} &s^T(\dot{D} - 2C)s = 0, \qquad \forall s \in R^n \\ &\quad \leftrightarrow \quad s^T \dot{D} s = 2 s^T C s \end{aligned} \tag{31}$$

The matrix$(\dot{D} - 2C)$ is symmetric. This characteristic ensures the system unaffected by the force is defined by$C(q, \dot{q})\dot{q}$. Therefore, from (30) and (31), we have:

$$\dot{V}(t) = s^T C s + s^T D \dot{s} + \sum_{j=1}^{n} w_j^T \dot{w}_j \tag{32}$$

Setting the sliding surface structured with PD and neural network (26) to cling to the reference trajectory of the Euler - Lagrange system (19) with the deviation of e equal to 0, we propose the following control law:

$$\begin{aligned} \tau = D\ddot{q}_d + C\dot{q}_d + G - D\Lambda\dot{e} - C\Lambda e \\ - Ks - \gamma \frac{s}{\|s\|} + (1+\eta) W \sigma \end{aligned} \tag{33}$$

The learning algorithm:

$$\dot{w}_i = -\eta s \sigma_i \tag{34}$$

where K is a symmetric positive matrix and$\gamma, \eta > 0$.

The equation of sliding surface:

$$\mathrm{s} = \dot{\mathrm{e}} + \Lambda \mathrm{e} \tag{35}$$

Theorem 1 considers the Euler–Lagrange system (19), sliding mode (35), approximating neural network defined in (27) with Gaussian function described in (28), the proposed state feedback control law (33), and learning algorithm (34). All the system signals are limited, and the tracking error congregates to sliding surfaces in a limited time.

Demonstration of theorem 1. Substituting the PD sliding surface described in the form (35) to the (32), we get:

$$\begin{aligned}\dot{V}(t) &= s^T[C(\dot{e} + \Lambda e + D(\ddot{e} + \Lambda\dot{e})] + \sum\nolimits_{j=1}^{n} w_j^T\, \dot{w}_j \\ &= s^T[C(\dot{q} - \dot{q}_d + \Lambda e) + D(\ddot{q} - \ddot{q}_{\mathrm{d}} + \Lambda\dot{e})] + \sum\nolimits_{j=1}^{n} w_j^T\, \dot{w}_j \\ &= s^T[C(-\dot{q}_d + \Lambda e) + D(-\ddot{q}_{\mathrm{d}} + \Lambda\dot{e}) + C\dot{q} + D\ddot{q}] + \sum\nolimits_{j=1}^{n} w_j^T\, \dot{w}_j \end{aligned} \tag{36}$$

Basing on (19) and (25), we get:

$$D\ddot{q} + C\dot{q} = \tau - G - F(q, \dot{q}, \ddot{q}, t) = \tau - G - W\sigma - \varepsilon \tag{37}$$

Replacing (37) in (36), we get:

$$\begin{aligned}\dot{V}(t) = s^T[&C(-\dot{q}_d + \Lambda e) + D(-\ddot{q}_{\mathrm{d}} + \Lambda\dot{e}) + \\ &+ \tau - G - W\sigma - \varepsilon] + \sum\nolimits_{j=1}^{n} w_j^T\, \dot{w}_j\end{aligned} \tag{38}$$

Replacing τ in (33) and (34) into (38), we get:

$$\dot{V}(t) = s^T[-Ks - \gamma\frac{s}{\|s\|} + \eta W\sigma - \varepsilon] - \eta\sum_{j=1}^{n} w_j^T\, s\sigma_j \tag{39}$$

We have$w_j^T s = s^T w_j$. Basing on (27) and (28), the last part in (39) can be rewritten to the following:

$$\eta\sum\nolimits_{j=1}^{n} w_j^T\, s\sigma_j = \eta s^T\sum\nolimits_{j=1}^{n} w_j \;\; \sigma_j = \eta s^T W\sigma \tag{40}$$

In (39) is rewritten as shown in:

$$\begin{aligned}\dot{V}(t) &= s^T[-Ks - \gamma\frac{s}{\|s\|} + \eta W\sigma - \varepsilon] - \eta s^T W\sigma \\ &= s^T[-Ks - \gamma\frac{s}{\|s\|} - \varepsilon]\end{aligned} \tag{41}$$

From (41) we can see that $\dot{V}(t) < 0$ for all$s \neq 0$and $\dot{V}(t) = 0$ if and only if$s = 0$. So, with the sliding surface (35), control law (33), learning algorithm (34), and neural network (26), the agent will be tracking the desired trajectory $q_d$with the error$e \to 0$.

4. RESULTS AND DISCUSSION

Firstly, we run the neural network to identify the robot with the sample period time T_{sp} = 0.01. After training, we get the following results. The mse deviation between NN model output and object output is 2.5 $*10^{-12}$. The biggest deviation between NN model output and robot output is 6 $* \; 10^{-5}$. The number of training is 1350. The neural network used for identification as (17), the Gaussian function is described in (28), where the Gaussian function parameters are as shown in:

$$\lambda_1 = 1.2; \lambda_2 = 1.2; c_1 = 0.01; c_2 = 0.01$$

The initial conditions of the neural network are selected as shown in:

$$w_{0,0} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Learning rate $\eta = 0.1$and$\gamma = 2$. The input/output responses of the reference model, the output responses of the object, and the responses of the controller reference model are shown in Figures 2 and Figure 3. The error between the NN model and the object is shown in Figure 4.

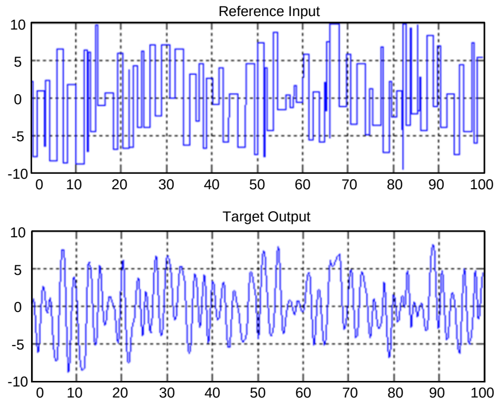


Figure 2. The signal of reference model input and output

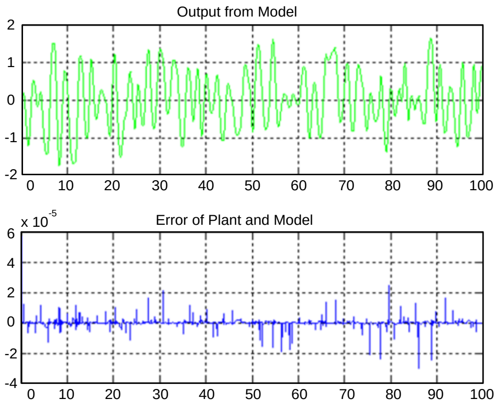


Figure 3. The NN model output, the error between NN output and object output

The parameters of the model are chosen as shown in:

$m_w = 0.3\text{kg}$; $m_1 = 1\text{kg}$; $m_2 = 2\text{kg}$; $l = 0.1\text{m}$; $d = 0.2\text{m}$; $g = 10\ \text{m}^{*}\text{s}^{-2}$; $r = 0.035\text{m}$

Setting the sinusoidal random noise, the force acting on the two wheels are as shown in:

$f_{dL} = - f_{dR} = 2.85 sin(((2t) + rank)\ N$

The torque response on the two wheels is shown in Figure 5.

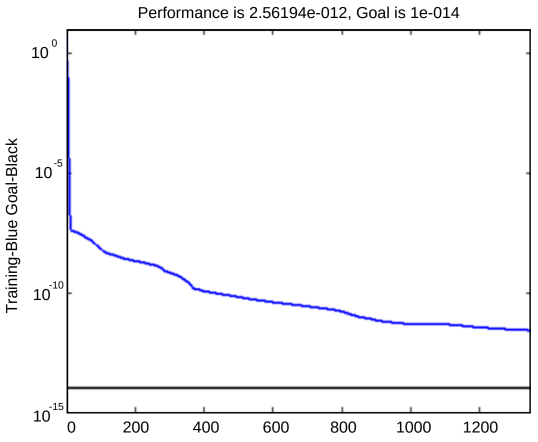


Figure 4. The mse error between NN output and object output

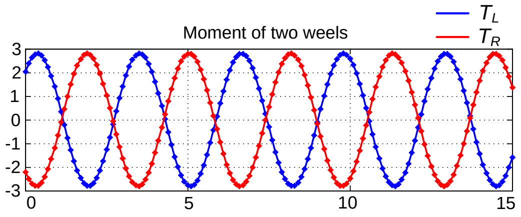
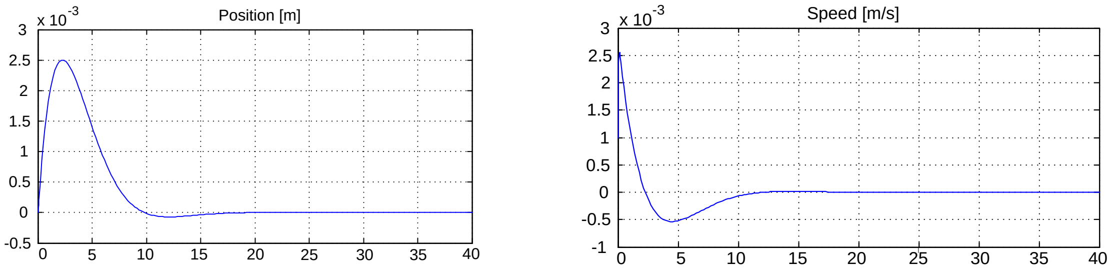


Figure 5. The torque response on the two wheels

The initial states of the autonomous robot system are selected as shown in:

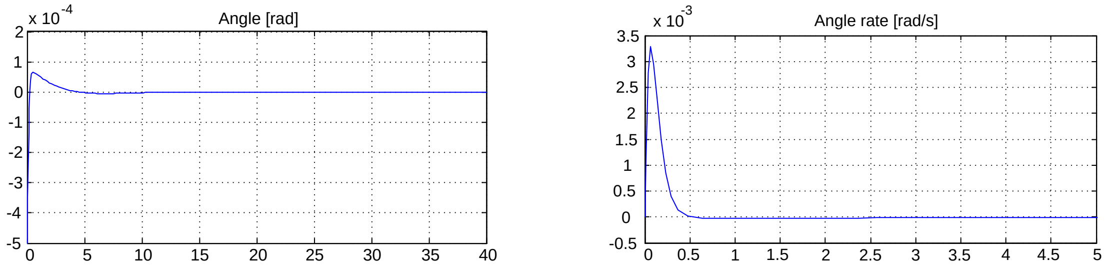
$[x;\dot{x};\theta;\dot{\theta}] = [0; 0.001; -0.0005; 0]$

The system responses are shown in Figures 6 and 7, respectively. Figure 6(a) shows the position response and Figure 6(b) shows speed response of the robot. Figure 7(a) shows the angle response and Figure 7(b) shows angle rate response of the robot. The results show that the system is well-stabilized, matched the proven theoretical with high efficiency. The parameters of position, speed, angle, and angular velocity quickly return to the equilibrium position after no more than 10 seconds. Therefore, we can confirm that the control method proposed by the authors has worked effectively.



(a) (b)

Figure 6. The response of the robot, (a) the position and (b) the speed



(a) (b)

Figure 7. The response of the robot, (a) the angle and (b) the angle rate

5. CONCLUSION

In this paper, based on the neural network structure combined with the sliding mode controller with PD-surface, the authors have designed a TWSB robot controller that can compensate for noise in an uncertain environment. The authors have taken advantage of two methods: the neural network has a high adaption and the sliding mode controller has a fast response. The neural network parameters are always defined and updated online to ensure stability in varying noise. Compared to previous approaches, this method's advantage is that the noise detection and filtering system can compensate for any disturbance and ensure the system's stability quickly. The results of this study will be the foundation for the authors' further studies in improving the quality of the consensus control system for many TWSB robots.

ACKNOWLEDGEMENTS

Nguyen Duc Minh, VINIF.2020.TS.127 was funded by Vingroup Joint Stock Company and supported by the Domestic Master/ PhD Scholarship Programme of Vingroup Innovation Foundation (VINIF), Vingroup Big Data Institute (VINBIGDATA)”.

REFERENCES

[1] M. S. Arani, H. E. Orimi, W.-F. Xie, and H. Hong, “Sliding mode control design of a two-wheel inverted pendulum robot: simulation, design and experiments,” in *Advances in Motion Sensing and Control for Robotic Applications*, Springer, Cham, 2019, pp. 93-107, doi: 10.1007/978-3-030-17369-2_7.

[2] O. S. Bhatti, K. M.-ul-Hasan, and M. A. Imtiaz, “Attitude Control and Stabilization of a Two-Wheeled Self-Balancing Robot,” *Control Engineering and Applied Informatics (CEAI),* vol.17, no.3, pp. 98-104, 2015.

[3] M. O. Asali, F. Hadary, and B. W. Sanjaya, “Modeling, Simulation, and Optimal Control for Two-Wheeled Self-Balancing Robot,” *International Journal of Electrical and Computer Engineering (IJECE)* vol. 7, no. 4, pp. 2008-2017, 2017, doi: 10.11591/ijece.v7i4.pp2008-2017.

[4] S.-C. Lin, C.-C. Tsai, and H.-C. Huang, “Adaptive Robust Self-Balancing and Steering of a Two-Wheeled Human Transportation Vehicle,” *Journal of Intelligent & Robotic Systems*, vol. 62, no. 1, pp. 103-123, 2010, doi: 10.1007/s10846-010-9460-5.

[5] J. Fang, “The LQR Controller Design of Two-Wheeled Self-Balancing Robot Based on the Particle Swarm Optimization Algorithm,” *Mathematical Problems in Engineering*, vol. 2014, doi: 10.1155/2014/729095.

[6] M. Nadour, A. Essadki, T. Nasser, and M. Fdaili, "Robust coordinated control using backstepping of flywheel energy storage system and DFIG for power smoothing in wind power plants," *International Journal of Power Electronics and Drive Systems (IJPEDS)*, vol. 10, no. 2, p. 1110, 2019, doi: 10.11591/ijpeds.v10.i2.pp1110-1122.

[7] N. G. M. Thao, D. H. Nghia, and N. H. Phuc, "A PID backstepping controller for two-wheeled self-balancing robot," in *International Forum on Strategic Technology 2010*, Ulsan, Korea (South), 2010, pp. 76-81, doi: 10.1109/IFOST.2010.5668001.

[8] E. H. Karam and N. Mjeed, "Modified Integral Sliding Mode Controller Design based Neural Network and Optimization Algorithms for Two Wheeled Self Balancing Robot," *International Journal of Modern Education and Computer Science (IJMECS)*, vol.10, no.8, pp. 11-21, 2018, doi: 10.5815/ijmecs.2018.08.02.

[9] S. Miasa, M. Al-Mjali, A. Al-Haj Ibrahim, and T. A. Tutunji, "Fuzzy control of a two-wheel balancing robot using DSPIC," in *2010 7th International Multi- Conference on Systems, Signals and Devices*, Amman, Jordan, 2010, pp. 1-6, doi: 10.1109/SSD.2010.5585525.

[10] A. Unluturk and O. Aydogdu, "Adaptive control of two-wheeled mobile balance robot capable to adapt different surfaces using a novel artificial neural network-based real-time switching dynamic controller," *International Journal of Advanced Robotic Systems*, vol. 14, no. 2, p. 1729881417700893, doi: 10.1177/1729881417700893.

[11] W. An and Y. Li, "Simulation and control of a two-wheeled self-balancing robot," in *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Shenzhen, China, 2013, pp. 456-461, doi: 10.1109/ROBIO.2013.6739501.

[12] A. A. Bature, S. Buyamin, M. N. Ahmad, and M. Muhammad, "A Comparison of Controllers for Balancing Two Wheeled Inverted Pendulum Robot," *International Journal of Mechanical & Mechatronics Engineering*, vol. 14, no. 3, pp. 62-68, 2014.

[13] N. N. Son and H. P. H. Anh, "Adaptive Backstepping Self-Balancing Control of a Two-wheel Electric Scooter," *International Journal of Advanced Robotic Systems*, vol. 11, no. 10, p. 165, 2014, doi: 10.5772/59100.

[14] N. G. M. Thao, M. T. Dat, D. H. Nghia, and N. H. Phuc, "Nonlinear Controllers for Two-Wheeled Self-Balancing Robot," in *Proceedings of the ASEAN Symposium on Automatic Control 2011 (ASAC 2011)*, Ho Chi Minh City, Vietnam, 2011, pp 8-13.

[15] H. W. Kim and S. Jung, "Fuzzy Logic Application to a Two-wheel Mobile Robot for Balancing Control Performance," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 12, no. 2, pp. 154-161, June 2012, doi: 10.5391/IJFIS.2012.12.2.154.

[16] J. Wu, W. Zhang, and S. Wang, "A Two-Wheeled Self-Balancing Robot with the Fuzzy PD Control Method," *Mathematical Problems in Engineering*, vol. 2012, pp. 1-13, 2012, doi: 10.1155/2012/469491.

[17] M. N. Duc and T. N. Trong, "Neural network structures for identification of nonlinear dynamic robotic manipulator," in *2014 IEEE International Conference on Mechatronics and Automation*, Tianjin, China, 2014, pp. 1575-1580, doi: 10.1109/ICMA.2014.6885935.

[18] N. D. Minh, W. Wei, and Z. Yan, "Consensus of Multi-Agent Systems with Euler–Lagrange System Using Neural Networks Controller," *ICIC Express Letters*, vol. 4, no. 5, pp. 1-6, 2016.

[19] N. D. Minh and N. T. Thang, "Sliding Surface in Consensus Problem of Multi-Agent Rigid Manipulators with Neural Network Controller," *Energies*, vol. 10, no. 2, p. 2127, 2017, doi: 10.3390/en10122127.

[20] Y. Liu, X. Huang, T. Wang, Y. Zhang, and X. Li, "Nonlinear dynamics modeling and simulation of two-wheeled self-balancing vehicle," *International Journal of Advanced Robotic Systems*, vol. 13, no. 6, p. 1729881416673725, doi: 10.1177/1729881416673725.

[21] Q. Qian, J. Wu, and Z. Wang, "A Novel Configuration of Two-Wheeled Self-Balancing Robot," *Tehnički vjesnik*, vol. 24, no. 2, pp. 459-464, 2017, doi: 10.17559/TV-20160608105436.

[22] H. Nurhadi, E. Apriliani, T. Herlambang, and D. Adzkiya, "Sliding mode control design for autonomous surface vehicle motion under the influence of environmental factor," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 5, pp. 4789-4797, 2020, doi: 10.11591/ijece.v10i5.pp4789-4797.

[23] T. T. Nguyen, "Sliding mode control-based system for the two-link robot arm," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 4, p. 2771, 2019, doi: 10.11591/ijece.v9i4.pp2771-2778.

[24] H. Aithal and S. Janardhanan, "Trajectory tracking of two wheeled mobile robot using higher order sliding mode control," in *2013 International Conference on Control, Computing, Communication and Materials (ICCCCM)*, Allahabad, India, 2013, pp. 1-4, doi: 10.1109/ICCCCM.2013.6648908.

[25] Y. Qin, Y. Liu, X. Zang, and J. Liu, "Balance control of two-wheeled self-balancing mobile robot based on TS fuzzy model," in *Proceedings of 2011 6th International Forum on Strategic Technology*, Harbin, China, 2011, pp. 406-409, doi: 10.1109/IFOST.2011.6021051.

[26] T. T. Nguyen, "The neural network-based control system of direct current motor driver," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 2, pp. 1445-1452, 2019, doi: 10.11591/ijece.v9i2.pp1445-1452.

[27] T. T. Nguyen, "The neural network-combined optimal control system of induction motor," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 4, pp. 2513-2522, 2019, doi: 10.11591/ijece.v9i4.pp2513-2522.