International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I7P105

Volume 72 Issue 7, 45-57, July 2024

Original Article

Balancing and Trajectory Tracking Control for Two-Wheeled Self-Balancing Robot

Nguyen Cao Cuong[1], Hoang Dinh Co[2]

[1,2]*Faculty of Electrical Engineering - Automation, University of Economics – Technology for Industries, Ha Noi, Viet Nam.*

[1]*Corresponding Author : nccuong@uneti.edu.vn*

Received: 22 February 2024 Revised: 25 May 2024 Accepted: 05 June 2024 Published: 26 July 2024

Abstract *- The Two-Wheeled Self-Balancing Robot (TWSBR) system is a system widely applied in automatic control experiments. This is a highly theoretical and practical Multi-Input and Multi-output (MIMO) system that has been applied in life. However, most of the research only revolves around balance control using trial-and-error search algorithms or simple mathematical equations. There are not many detailed studies on system mathematical equations and main application algorithms based on model understanding. This article presents a method to design and control a self-balancing two-wheeled robot. The main contents include the design of a robust adaptive controller based on the nominal model for TWSBR. The goal of the proposed controller is to stabilize the TWSBR body tilt angle and control the desired trajectory tracking for the robot. Simulation results show that the proposed controller has good quality and is robust to disturbances.*

Keywords *- Two-Wheeled Self-Balancing Robot, Adaptive controller, Robust controller, Trajectory tracking control, Robot.*

1. Introduction

TWSBR is an unstable dynamic system, which means the robot is free to fall forward or backwards without any external force. Self-balancing means that the robot is in a state where its position is like standing upright at 90 degrees from the horizontal plane. However, the system itself is not self-balanced, meaning it continues to move away from the vertical axis. Therefore, a combination of gyroscope and accelerometer is needed to read the angular position of the robot and input it to the microcontroller, which performs the balancing procedure. Self-balancing robots are based on the inverted pendulum theory - an inherently unstable generation that must be applied properly to keep the system intact. TWBSR is a nonlinear system, so to achieve this, an appropriate control theory is required. There have been many control algorithms that have been published and experimented such as PID control [1-3], control by Fuzzy Logic method [4-7], and artificial neural network [8-11]. In [12], a low-cost two-wheeled inverted pendulum robot is presented with friction compensation, and two sliding mode controllers have been designed based on the dynamic model derived from the Lagrange method. In [13,14], a partial feedback linearized controller is proposed to make the closed-loop system locally stable around the equilibrium point by constructing a suitable Lyapunov candidate function. In the document [15], the sliding mode controller with proportional-integral (PID) sliding surface parameters is optimized using the Particle Swarm Optimization (PSO) method. In the document [16], a high-order sliding controller is presented that uses a function similar to the sign function, the tanh function, to create smoother switching signals and reduce chattering for the control signal. In [17], an indirect adaptive fuzzy controller is proposed, along with a sliding mode control technique, so that the controller is enhanced to uncertain disturbances. In [18], a desired velocity control law was first designed using the Lyapunov analysis method and the arctan function, and a nonlinear turbulence observer was developed to deal with the unknown turbulence. Using the output of the nonlinear disturbance observer, the tracking control scheme has been designed using sliding mode techniques to ensure that all final closed-loop signals are uniformly limited. In [19], a rolling PID controller with three control loops is proposed; the first loop uses a rolling controller to maintain the robot in a balanced state, the second loop uses a PD controller to control the robot's position, and the third one uses a PI controller to control the robot's movement direction. In the document [20], the design of a fuzzy logic adaptive controller is proposed to balance the dynamics and stably follow a given trajectory of an inverted pendulum mobile robot, and the controller ensures blocked stability. Uniform semi-global and steady-state sets to which the closed-loop error signals converge are derived. In the document [21], an adaptive controller using a radial basis function network is proposed to control a two-wheeled self-balancing vehicle; the effectiveness of the controller is realized through several simulations and Experimenting with the model; the adaptive controller can achieve self-balancing, direction angle control, and slow-speed movement. In the document [22], a terminal sliding controller is presented that uses the RBF neural network to approximate the unknown nonlinear function of the robot arm model, the convergence and stability are guaranteed by Lyapunov's theorem. Park and Sandberg [23] demonstrated that an RBF network with the same impact coefficients for appropriately chosen kernels can approximate any continuous objective function. In control engineering, RBF networks are often used as a tool to model nonlinear functions because of their good ability in function approximation; Ge and Wang [24] used RBF networks to approximate interconnect functions customarily. In the literature [25], the RBF network is used to compensate for unknown dynamics, especially due to unknown loads, to ensure the steady-state performance of the controller. A large number of researchers have extensively tested the effectiveness of the linear parametric RBF network, and it has been theoretically proven that the RBF network can approximate any continuous function. However, most of the published works only consider the problem of balancing control for robots and do not consider the problem of orbital tracking in real applications. Therefore, this article proposes the design of a sustainable adaptive controller that ensures two factors: robot balance and tracking of the desired trajectory. The results show that the proposed controller has the advantage of operating well against disturbances affecting the system.

2. Mathematical Model of TWSBR

Consider the dynamic equations of a TWSBR described by the following Lagrange equation [26-28]:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) + D = B(q)\tau + f \qquad (1)$$

Where $q = [x \quad y \quad \theta \quad \alpha]^T$, $q \in \mathbb{R}^4$ is the vector of general coordinates where x and y are position coordinates, θ is the direction angle, and α is the tilt angle of the robot, as shown in Figure 1.

If q is decomposed into $q_d = [x \quad y \quad \theta]^T$ and α we have

$M(q) = \begin{bmatrix} M_v & M_{v\alpha} \\ M_{\alpha v} & M_\alpha \end{bmatrix}$, where M_v and M_α describe the inertia matrix for the mobile platform and inverted pendulum, respectively, $M_{v\alpha}$ and $M_{\alpha v}$ are the coupled inertia matrices of the mobile platform and inverted pendulum.

$C(q,\dot{q})) = \begin{bmatrix} C_v & C_{v\alpha} \\ C_{\alpha v} & C_\alpha \end{bmatrix}$, C_v and C_α represent the Coriolis and centrifugal moment for the mobile platform and inverted pendulum, respectively. $C_{v\alpha}$ and $C_{\alpha v}$ are the coupling of the Coriolis and centrifugal moments of the mobile platform and the inverted pendulum.

$G(q) = \begin{bmatrix} G_v \\ G_\alpha \end{bmatrix}$, G_v and G_α are the gravitational moment vectors for the mobile platform and the inverted pendulum, respectively.

$F(\dot{q}) = \begin{bmatrix} f_v \\ f_\alpha \end{bmatrix}$, $D = \begin{bmatrix} d_v \\ d_\alpha \end{bmatrix}$, $B(q) = \begin{bmatrix} B_v & 0 \\ 0 & B_\alpha \end{bmatrix}$, $\tau = \begin{bmatrix} \tau_v \\ 0 \end{bmatrix}$.

τ_v is the control input vector for the mobile platform, d_v and d_v represent the external disturbances on the mobile platform and the inverted pendulum, respectively. The control objective is to design a controller that ensures tracking errors of $q_v(t)$ and $\alpha(t)$ from the corresponding desired orbits of $q_{vd}(t)$ and $\alpha_d(t)$ in a neighborhood small bound is 0.

Defining J_v as the kinematic constraint matrix related to the nonholonomic constraints, the nonholonomic constraints on the means can be described as [29]:

$$J_v \dot{q}_v = 0 \qquad (2)$$

We can find a complete matrix of rank $\Gamma = [\Gamma_1(q) \quad \Gamma_2(q)] \in \mathbb{R}^{3\times 2}$, where $\Gamma_1(q)$ and $\Gamma_2(q)$ are vector fields -linearly independent smoothing satisfies [3].

$$\Gamma^T J_v^T = 0 \qquad (3)$$

Table 1. Meaning of matrices in the dynamic model

$M(q)$	is inertia matrix (dimension 4 × 4)
$C(q,\dot{q})$	is the Coriolis matrix and centrifugal force (dimension 4 × 4)
$G(q)$	is the vector of gravitational force (dimension 4 × 1)
$F(\dot{q})$	is the vector of force friction (dimension 4 × 1)
D	is the blocked vector of disturbances coming from the external environment (dimension 4 × 1)
$B(q)$	is the matrix of control coefficients (dimension 4 × 2)
τ	is the vector of control inputs (dimension 3 × 1)
$f = J^T\lambda$	represents the constraint forces (dimension 4 × 1)
J^T	is the matrix Jacobi (dimension 4 × 1)
λ	are sets of Lagrangian numbers corresponding to nonholonomic constraints

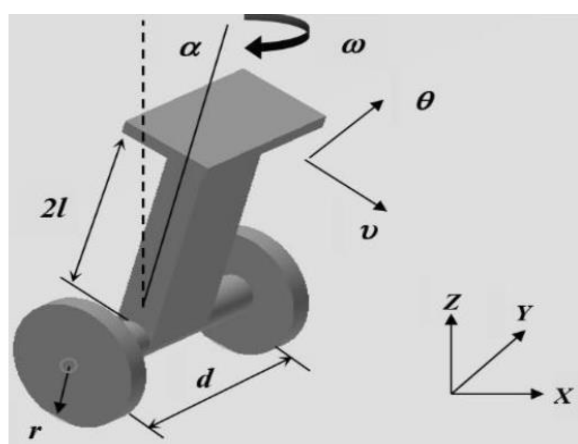

Fig. 1 Model of a TWSBR

In detail, $J_v = [0 \quad sin\theta \quad -cos\theta]$ and Γ can be written as follows

$$\Gamma = \begin{bmatrix} 1 & 0 \\ 0 & cos\theta \\ 0 & sin\theta \end{bmatrix} \tag{4}$$

The constraint equation (2) implies the existence of a vector. $\dot{\eta} = [\omega \quad v]^T \in \mathbb{R}^2$, so we have

$$\dot{q}_v = \Gamma(\mathrm{q})\dot{\eta} \tag{5}$$

Where $\omega \in \mathbb{R}$ is the angular velocity of the robot, $v \in \mathbb{R}$ is the longitudinal velocity of the robot.

Defining new variables $\zeta = [\eta^T \quad \alpha]^T$, we have $\dot{\zeta} = [\dot{\eta}^T \quad \dot{\alpha}]^T = [\dot{\zeta}_1 \quad \dot{\zeta}_2 \quad \dot{\zeta}_3]^T = [\omega \quad v \quad \dot{\alpha}]^T$, and multiply $diag[\Gamma^T \quad I]$, into both sides of (3) to eliminate J^T due to the properties of (3). Then, the dynamics of a self-balancing two-wheeled robot can be written as follows:

$$M_1(\zeta)\ddot{\zeta} + C_1(\zeta,\dot{\zeta})\dot{\zeta} + G_1(\zeta) + F_1(\dot{\zeta}) + D_1 = B_1(\zeta)\tau \tag{6}$$

Where

$$M_1(\zeta) = \begin{bmatrix} \Gamma^T M_v \Gamma & \Gamma^T M_{v\alpha} \\ M_{\alpha v}\Gamma & M_\alpha \end{bmatrix}$$

$$C_1(\zeta,\dot{\zeta}) = \begin{bmatrix} \Gamma^T M_v \dot{\Gamma} + \Gamma^T C_v \Gamma & \Gamma^T C_{v\alpha} \\ M_{\alpha v}\dot{\Gamma} + \Gamma^T C_{\alpha v} & C_\alpha \end{bmatrix}$$

$$G_1(\zeta) = \begin{bmatrix} \Gamma^T G_v \\ G_\alpha \end{bmatrix}$$

$$F_1(\dot{\zeta}) = \begin{bmatrix} \Gamma^T f_v \\ f_\alpha \end{bmatrix}$$

$$D_1 = \begin{bmatrix} \Gamma^T d_v \\ d_\alpha \end{bmatrix}$$

$$B_1(\zeta) = \begin{bmatrix} \Gamma^T B_v \tau_v \\ 0 \end{bmatrix}$$

The control objective in this study can be built according to the following formulas. The design of the controller for a self-balancing two-wheeled robot ensures that $|\zeta_1(t) - \zeta_{1d}(t)| \le \varepsilon_1$, $|\zeta_2(t) - \zeta_{2d}(t)| \le \varepsilon_2$, $|\zeta_3(t) - \zeta_{3d}(t)| \le \varepsilon_3$, $t \to \infty$, where $\zeta_{1d}(t)$, $\zeta_{2d}(t)$ and $\zeta_{3d}(t)$ are the desired orbits for $\zeta_1(t)$, $\zeta_2(t)$ and $\zeta_3(t)$ respectively, and $\varepsilon_i > 0$, $i = 1,2,3$ are small constants.

Considering the physical properties of the self-balancing two-wheeled robot presented in equation (6), we have

$$M_1(\zeta) = \begin{bmatrix} m_{11}(\zeta_3) & 0 & 0 \\ 0 & m_{22} & m_{23}(\zeta_3) \\ 0 & m_{23}(\zeta_3) & m_{33} \end{bmatrix}$$

$$C_1(\zeta,\dot{\zeta}) = \begin{bmatrix} v_{11} & 0 & v_{13} \\ 0 & 0 & v_{23} \\ v_{31} & 0 & 0 \end{bmatrix}$$

$$G_1(\zeta) = \begin{bmatrix} 0 \\ 0 \\ g_3(\zeta_3) \end{bmatrix}$$

$$F_1(\dot{\zeta}) = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

$$D_1 = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}$$

$$B_1(\zeta)\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ 0 \end{bmatrix}$$

In which m_{22}, m_{33} are unknown constants, $m_{11}(\zeta_3)$, $m_{23}(\zeta_3)$, v_{11}, v_{13}, v_{23}, v_{31}, f_1, f_2, f_3 , d_1, d_2, d_3 , are unknown continuous functions.

3. Controller Design

Define position tracking error as [30]

$$e(t) = \zeta_d(t) - \zeta(t) \tag{7}$$

where $\zeta_d(t)$ is ideal position signal, and $\zeta(t)$ is the practical position signal.

Define the sliding mode function as

$$r = \dot{e} + \Lambda e \tag{8}$$

where $\Lambda > 0$

Define $\dot{\zeta}_r = r(t) + \dot{\zeta}(t)$, then $\ddot{\zeta}_r = \dot{r}(t) + \ddot{\zeta}(t)$, $\dot{\zeta}_r = \dot{\zeta}_d + \Lambda e$, and $\ddot{\zeta}_r = \ddot{\zeta}_d + \Lambda\dot{e}$. From (6), we have

$$\begin{aligned} \tau &= M_1(\zeta)\ddot{\zeta} + C_1(\zeta,\dot{\zeta})\dot{\zeta} + G_1(\zeta) + F_1(\dot{\zeta}) + D_1 \\ &= M_1(\zeta)(\ddot{\zeta}_r - \dot{r}) + C_1(\zeta,\dot{\zeta})(\dot{\zeta}_r - r) + G_1(\zeta) + F_1(\dot{\zeta}) + D_1 \\ &= M_1(\zeta)\ddot{\zeta}_r + C_1(\zeta,\dot{\zeta})\dot{\zeta}_r + G_1(\zeta) + F_1(\dot{\zeta}) - M_1(\zeta)\dot{r} \\ &\quad - C_1(\zeta,\dot{\zeta})r + D_1 \\ &= M_0(\zeta)\ddot{\zeta}_r + C_0(\zeta,\dot{\zeta})\dot{\zeta}_r + G_0(\zeta) + F_0(\dot{\zeta}) + E' - M_1(\zeta)\dot{r} \\ &\quad - C_1(\zeta,\dot{\zeta})r + D_1 \end{aligned} \tag{9}$$

where $E' = E_M\ddot{\zeta}_r + E_M\dot{\zeta}_r + E_G + E_F$

For the system, a controller was proposed as [29]

$$\tau = \tau_m + K_p r + K_i \int_0^t r dt + \tau_r \tag{10}$$

where $K_p > 0$; $K_i > 0$; τ_m denotes control term based on the nominal model, τ_r denotes robust term, and

$$\tau_m = M_0(\zeta)\ddot{\zeta}_r + C_0(\zeta,\dot{\zeta})\dot{\zeta}_r + G_0(\zeta) + F_0(\dot{\zeta}) \tag{11}$$

$$\tau_r = K_r sgn(r) \tag{12}$$

where $K_r = diag[k_{rii}]$, $k_{rii} \ge |E_i|$, $i = 1, \dots. n$, and $E = E' + D_1$

From (9) - (12), we have

$$\begin{aligned} &M_0(\zeta)\ddot{\zeta}_r + C_0(\zeta,\dot{\zeta})\dot{\zeta}_r + G_0(\zeta) + F_0(\dot{\zeta}) - M_1(\zeta)\dot{r} \\ &\quad - C_1(\zeta,\dot{\zeta})r + E' + \tau_r \\ &= M_0(\zeta)\ddot{\zeta}_r + C_0(\zeta,\dot{\zeta})\dot{\zeta}_r + G_0(\zeta) + F_0(\dot{\zeta}) + K_p r \\ &\quad + K_i \int_0^t r dt + K_r sgn(r) \end{aligned} \tag{13}$$

Thus,

$$\begin{aligned} &M_1(\zeta)\dot{r} + C_1(\zeta,\dot{\zeta})r + K_i \int_0^t r dt \\ &= -K_p r - K_r sgn(r) + E \end{aligned} \tag{14}$$

3.1. Stability Analysis

An integration-type Lyapunov function is designed as [30,31]:

$$V = \frac{1}{2} r^T M r + \frac{1}{2}\left(\int_0^t r d\tau\right)^T K_i \left(\int_0^t r d\tau\right) \tag{15}$$

Then

$$\dot{V} = r^T \left[M\dot{r} + \frac{1}{2}\dot{M}r + K_i\left(\int_0^t r d\tau\right)\right] \tag{16}$$

Considering the skew-symmetric characteristics of the manipulator dynamic equation, $x^T\left(\dot{M}_1(\zeta) - 2C_1(\zeta,\dot{\zeta})\right)x = 0$, we have

$$\dot{V} = r^T \left[M\dot{r} + Cr + K_i\left(\int_0^t r d\tau\right)\right] \tag{17}$$

Substituting (7.7) into the above, we have

$$\begin{aligned}\dot{V} &= -r^T K_p r - r^T K_r sgn(r) + r^T E \\ &= -r^T K_p r - \sum_{i=1}^{n} K_{rii} \, |r|_i + r^T E \end{aligned} \tag{18}$$

Considering $K_{rii} \geq |E_i|$; then

$$\dot{V} \leq -r^T K_p r \leq 0 \tag{19}$$

4. Simulation Results

In the simulation, we consider a self-balancing two-wheeled robot system with the parameters defined as follows. M_p, m_w are the masses of the body and the pendulum, respectively, d is the distance between two wheels, l is half the length of the pendulum (assume that the center of gravity is in the middle of the pendulum). r is the radius of each wheel. g is the gravitational acceleration. I_m, I_w are the moment of inertia of the robot body and each wheel. θ is the direction angle of the robot body. τ_1, τ_2 are the control torque corresponding to the left and right motors of the wheels. For numerical calculations, the values of the system parameters are given in Table 2.

With the data given above, after calculation, we get some simulation results, as shown in Figures 2, 3, 4, and 5. From Figures 2 and 3, we see that when TWSBR moves on the horizontal plane, we can be controlled so that the robot moves at a constant speed and the robot body is stable in a vertical position ($\alpha = 0$).

Table 2. Parameters of TWSBR in simulation

Parameter	Value	Unit
M_p	2.5	kg
m_w	0.3	kg
d	0.12	m
l	0.35	m
r	0.048	m
g	9.81	m/s^2
I_m	0.055	kgm^2
I_w	0.051	kgm^2

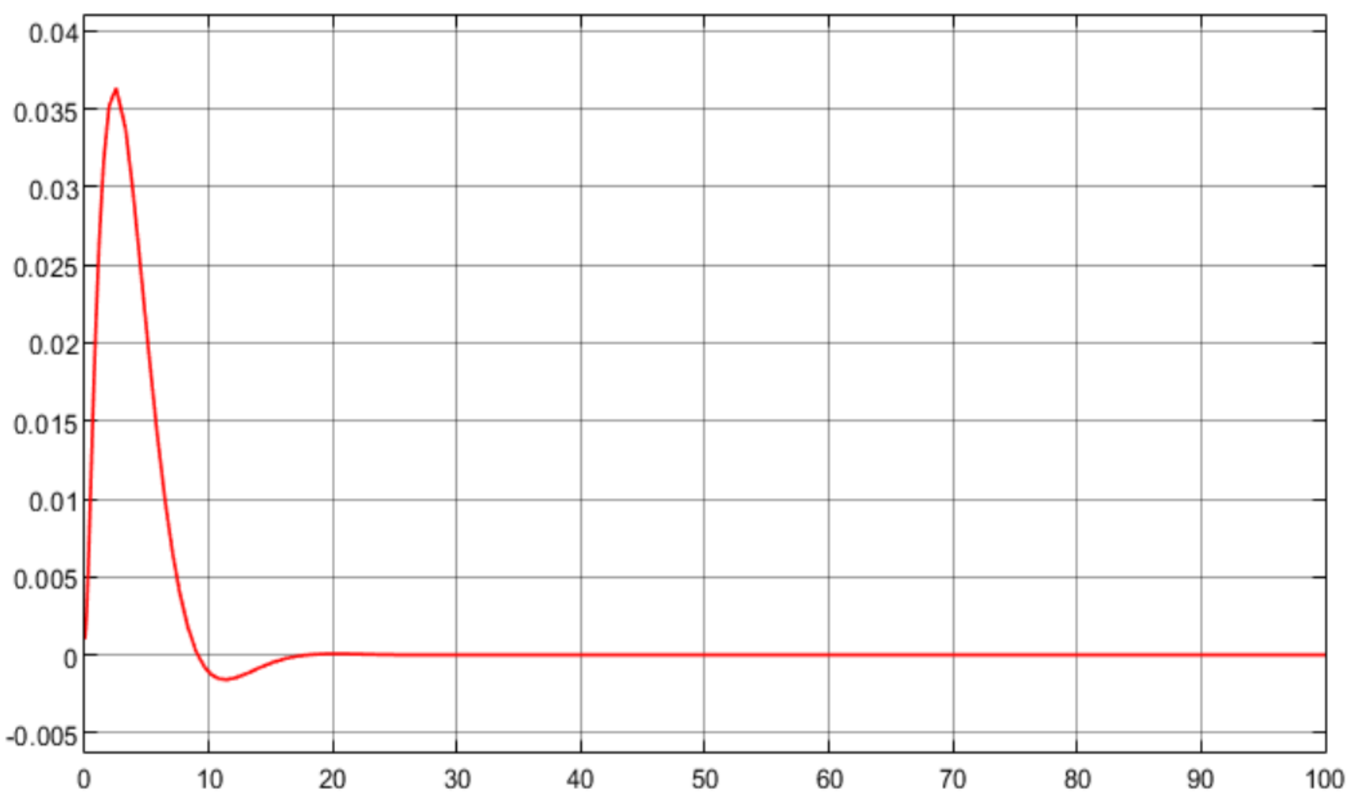

Fig. 2 Wheel angle of TWSBR in Scenario 1

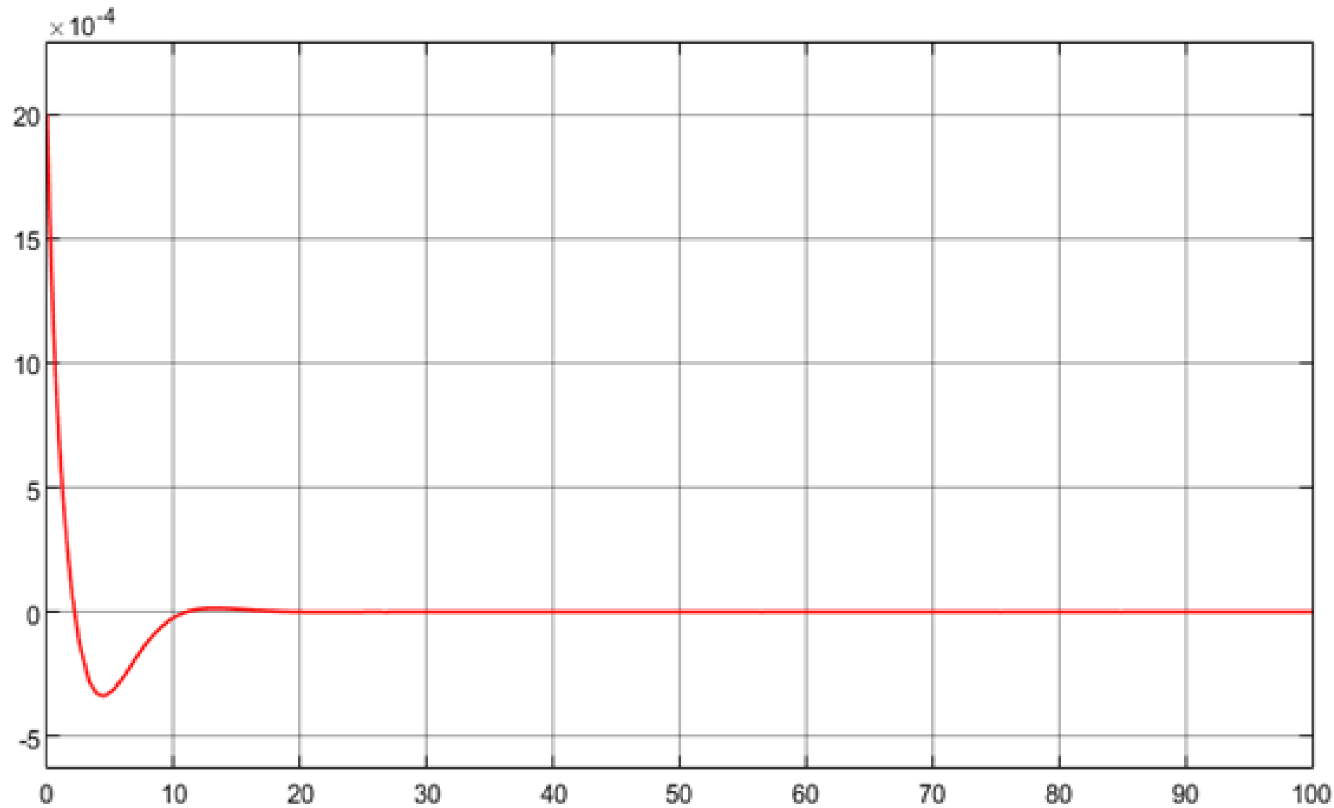

Fig. 3 Tilt angle of TWSBR in Scenario 1

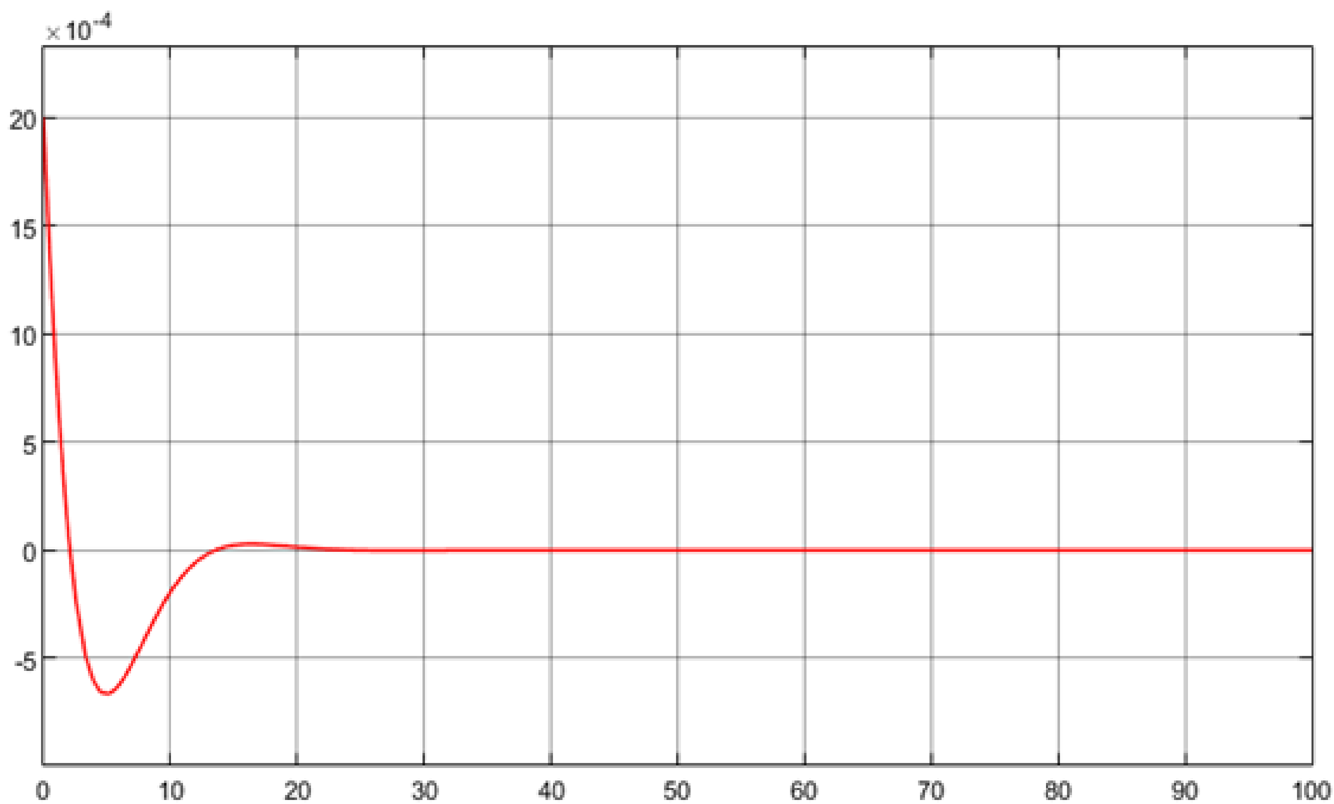

Fig. 4 Rotation angle of TWSBR in Scenario 1

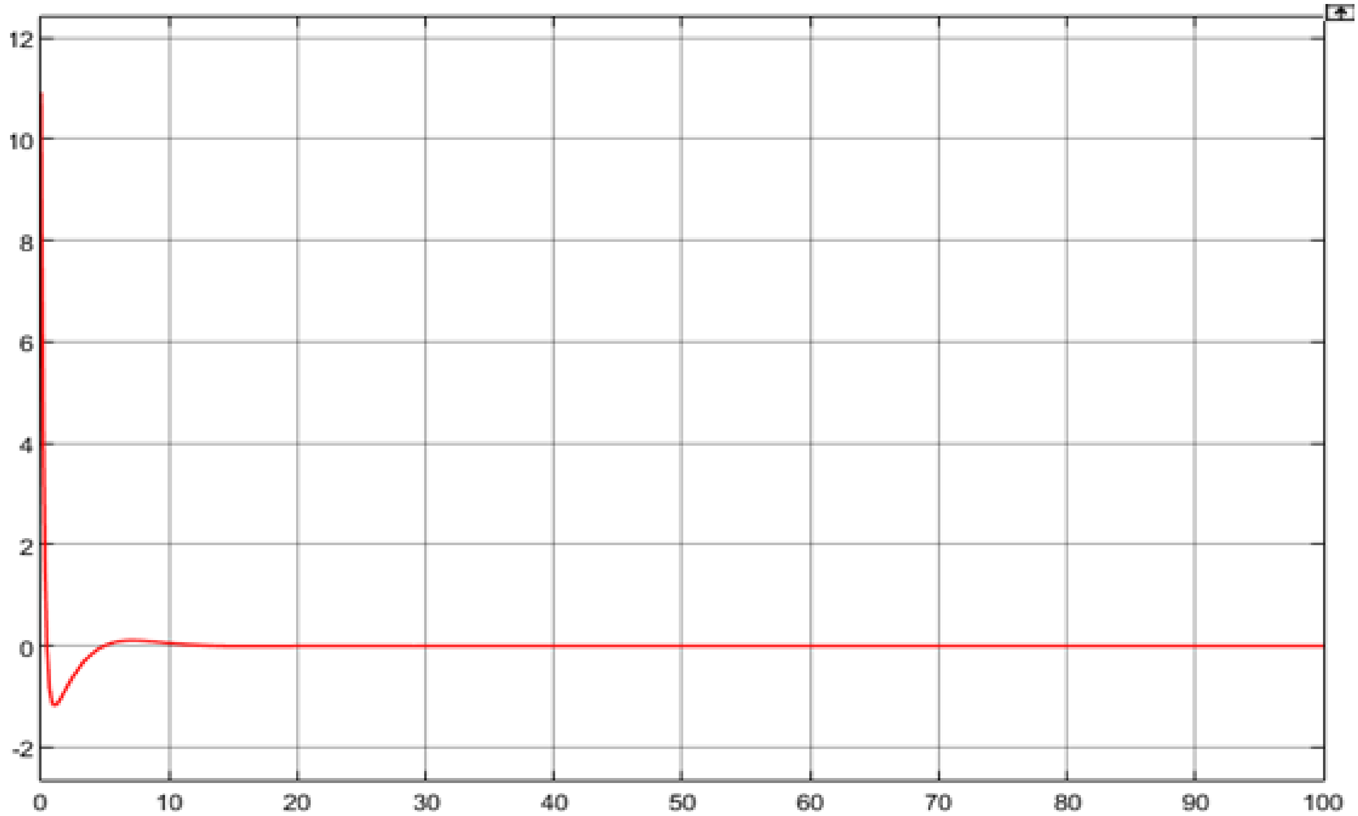

Fig. 5 Voltage supplied to the left motor of TWSBR in Scenario 1

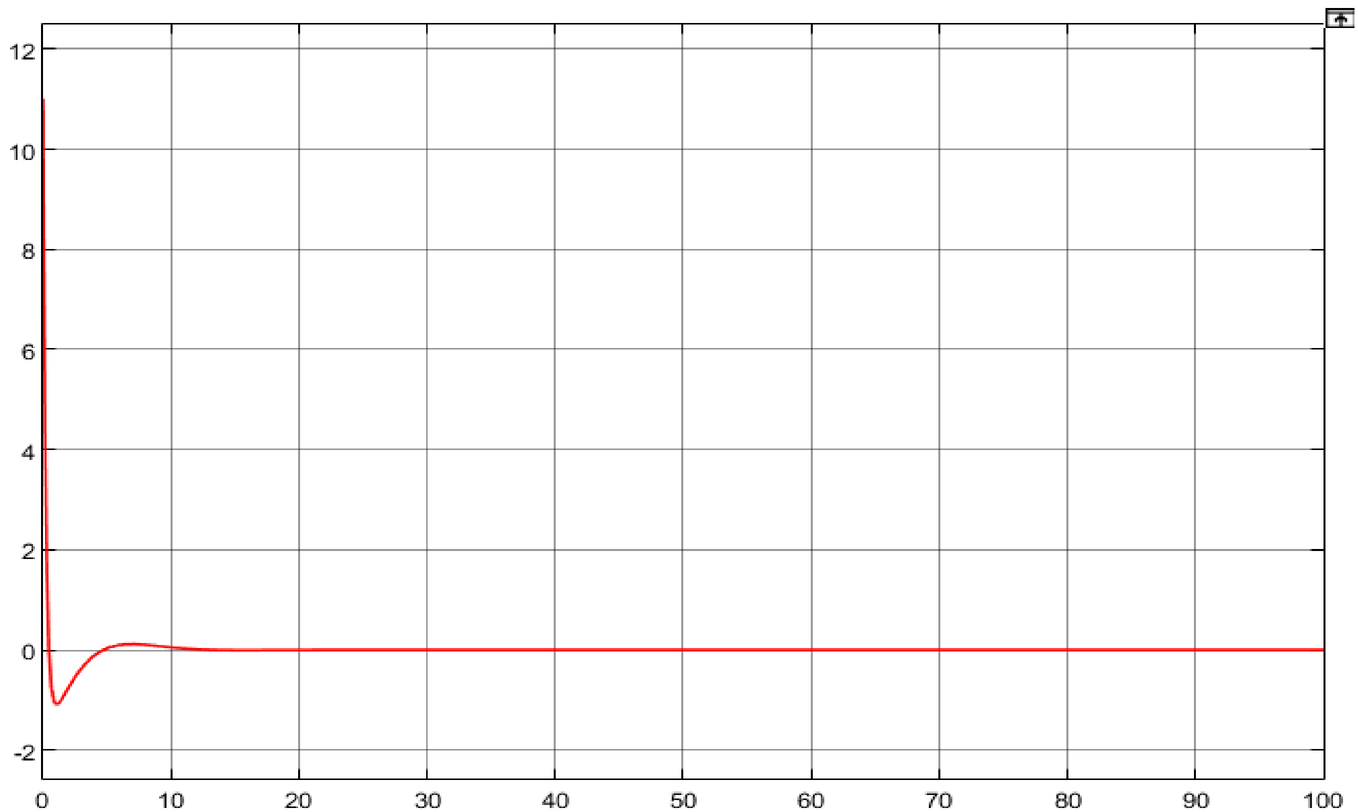

Fig. 6 Voltage supplied to the right motor of TWSBR in Scenario 1

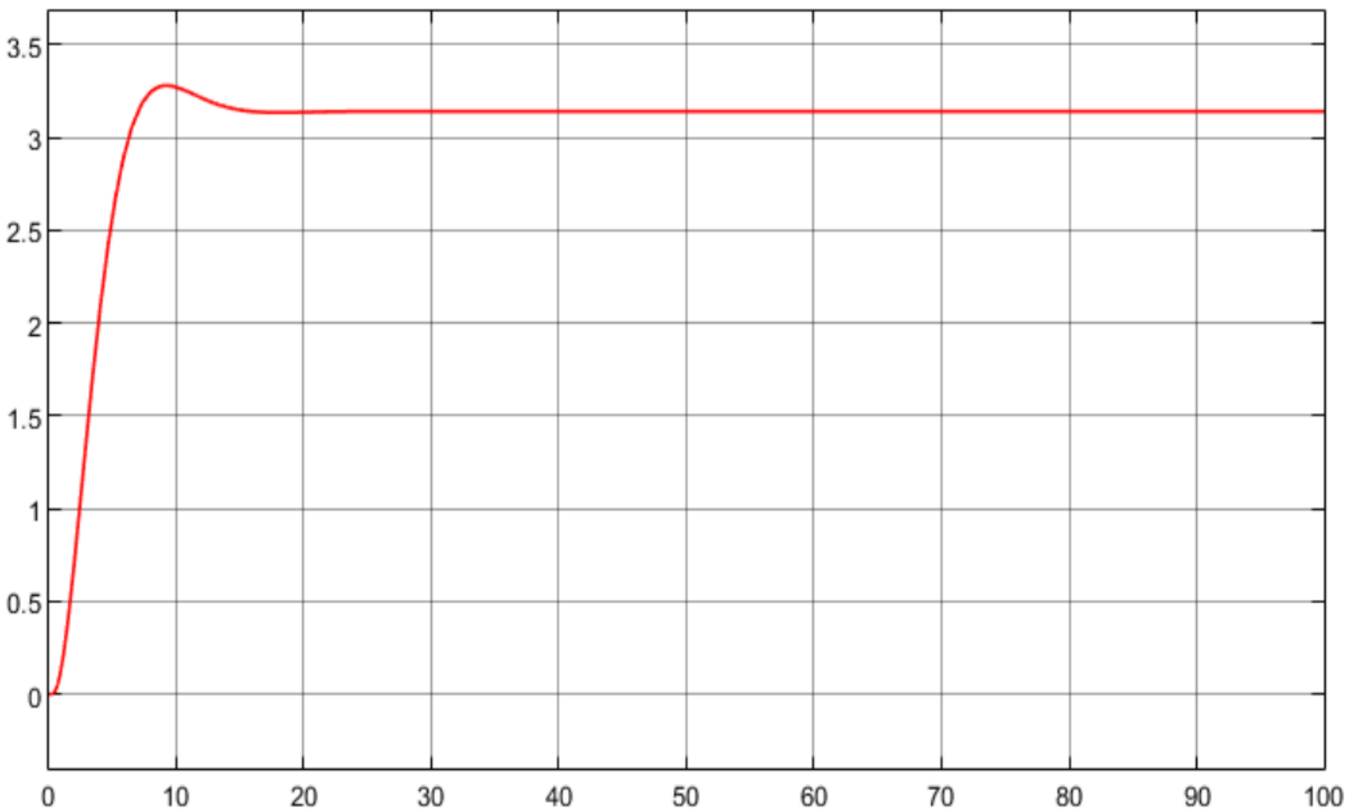

Fig. 7 Wheel angle of TWSBR in Scenario 2

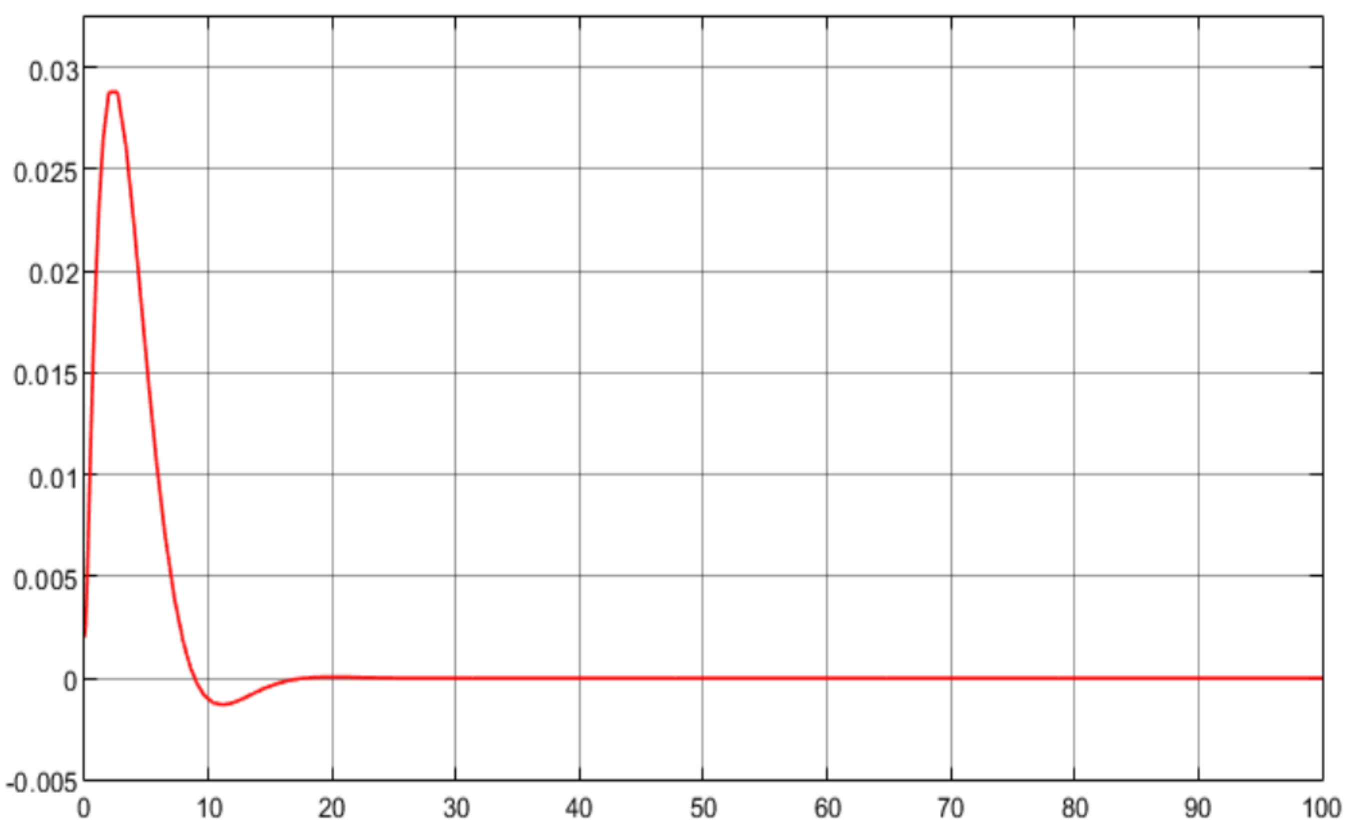

Fig. 8 Tilt angle of TWSBR in Scenario 2

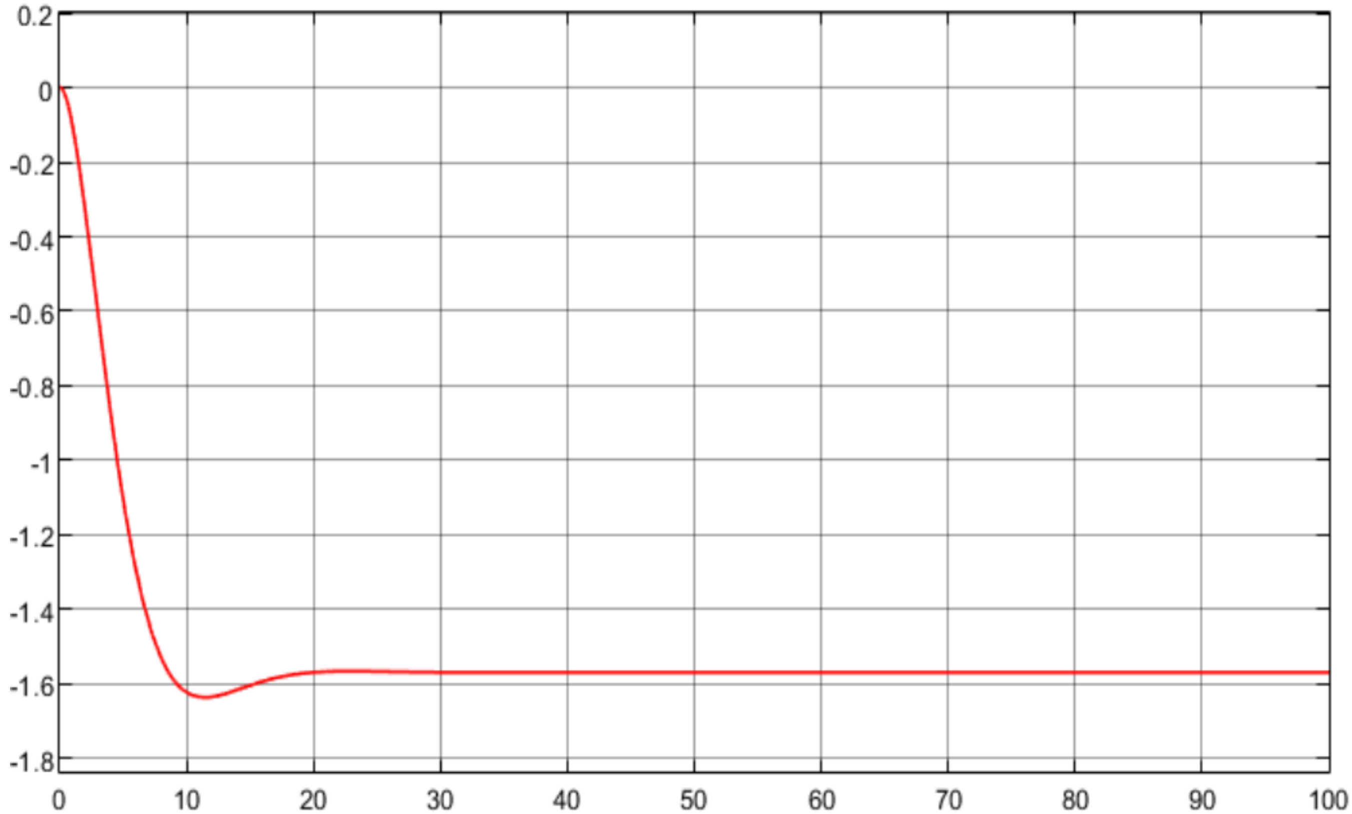

Fig. 9 Rotation angle of TWSBR in Scenario 2

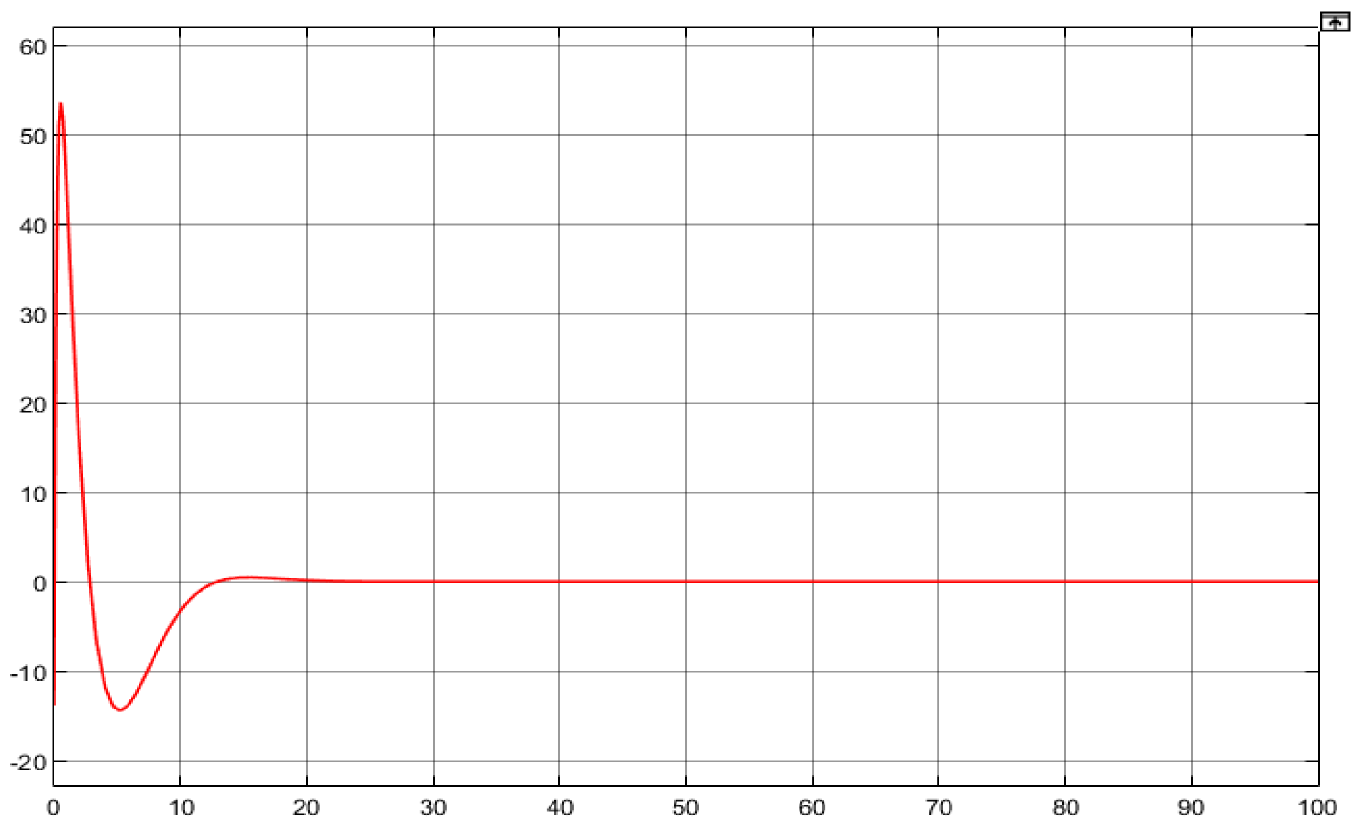

Fig. 10 Voltage supplied to the left motor of TWSBR in Scenario 2

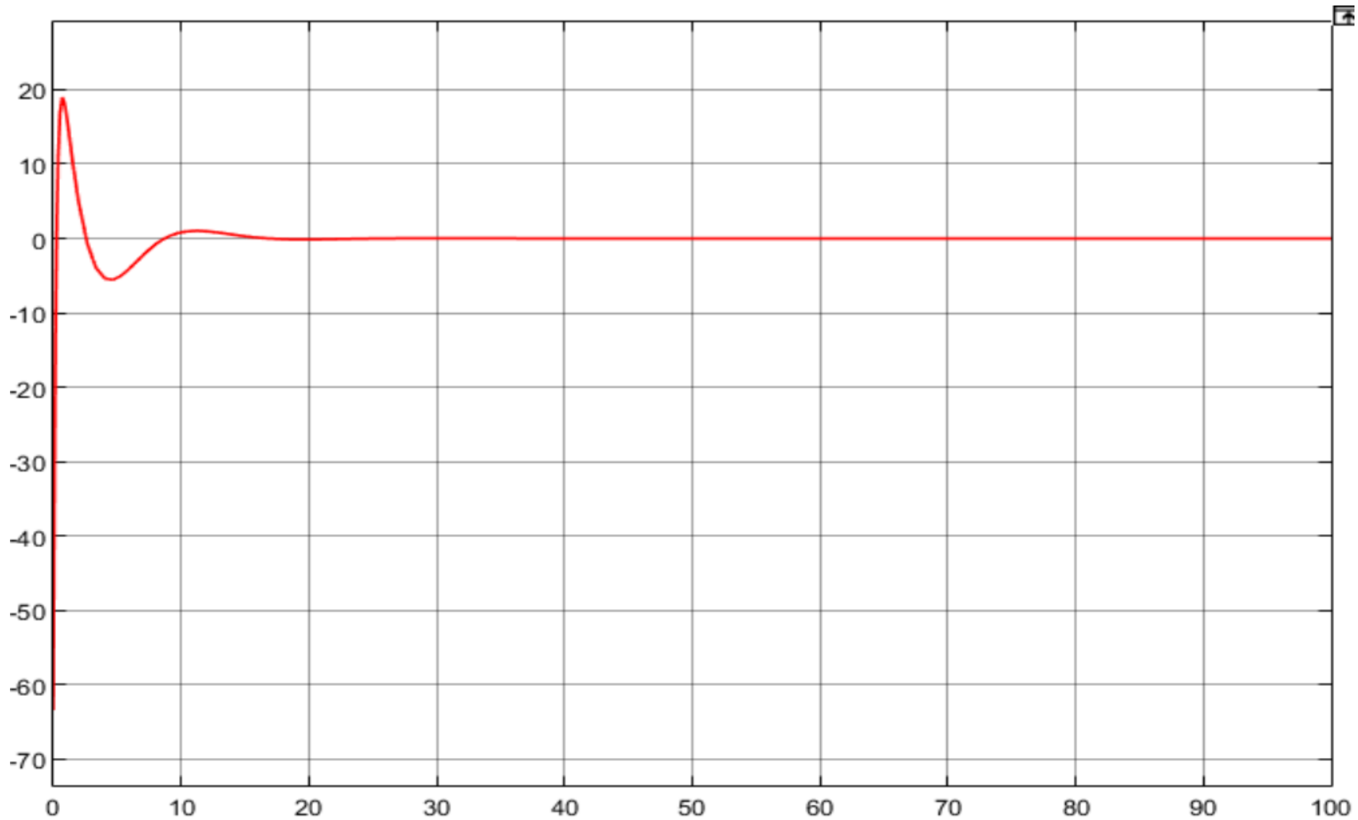

Fig. 11 Voltage supplied to the right motor of TWSBR in Scenario 2

We simulate in 2 scenarios:
Scenario 1: Set value is zero
Scenario 2: Set value is non-zero

Select the set value of the direction angle to be 3.14 rad and the set value of the tilt angle to be 1.57 rad. The robot moves forward at a distance equal to the direction of rotation of two wheels and rotates 90 degrees around the vehicle's axis.

Simulation results from Figures 2 to 11 show that the controller returns the state variables to the reference value, giving a good response. The controller's stability is high even when the system has external disturbances or model parameter uncertainties. This proves that TWSBR can move forward, backwards, turn right, and turn left depending on the set value.

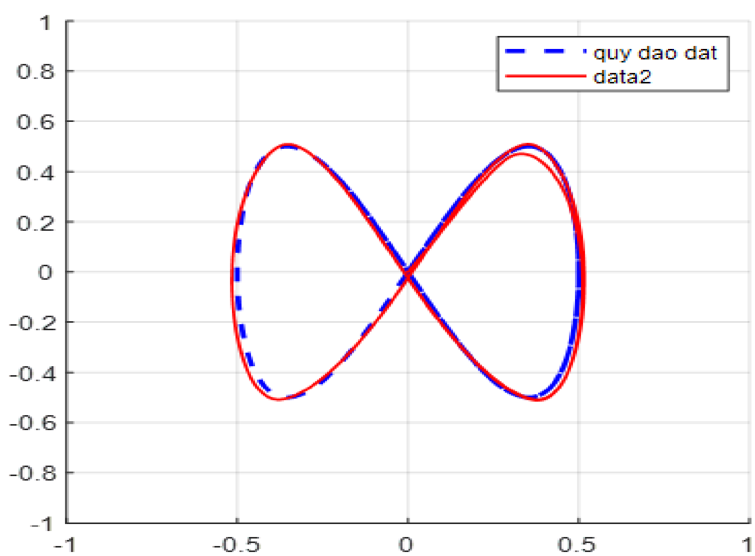

Fig. 12 TWSBR moves along a figure-8 trajectory

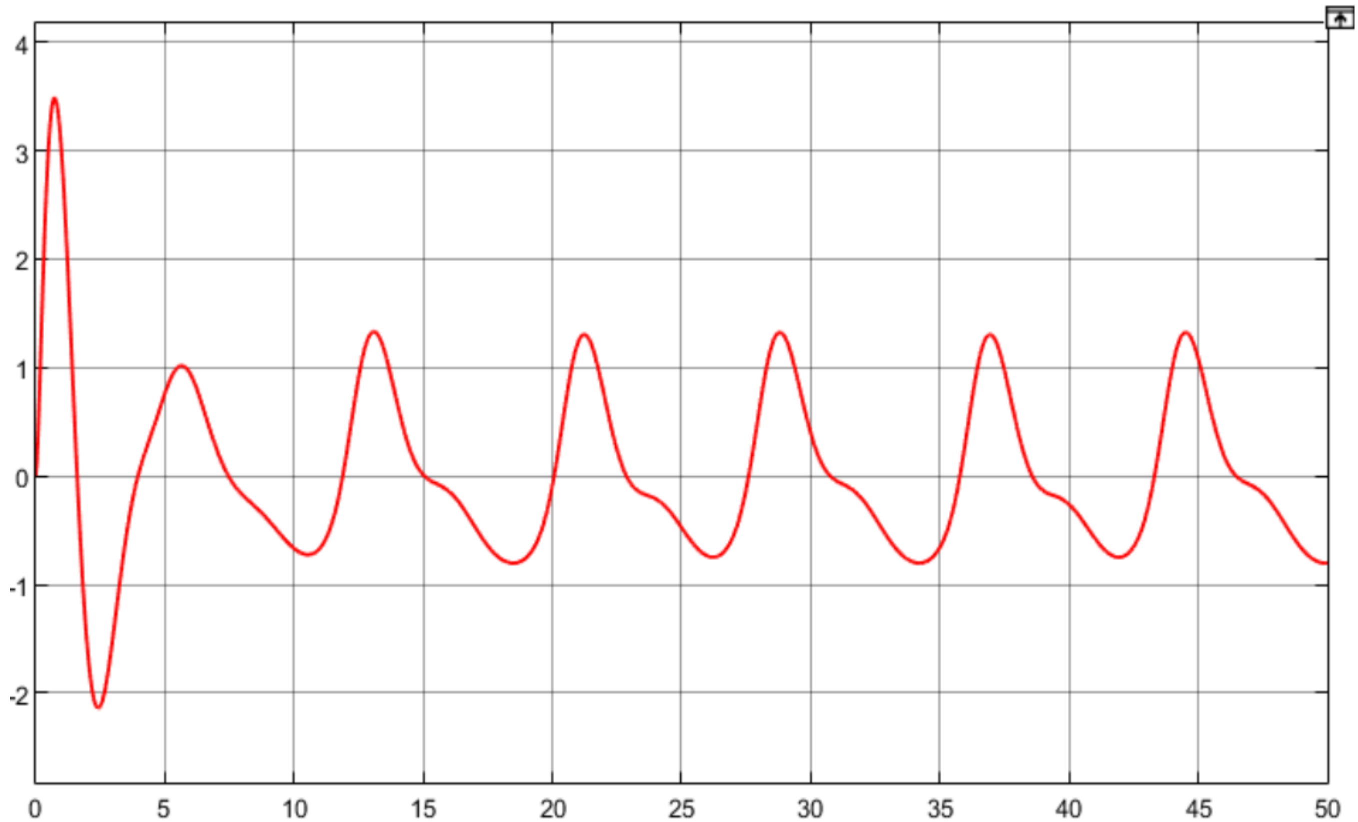

Fig. 13 TWSBR's wheel angle when tracking a figure-8 trajectory

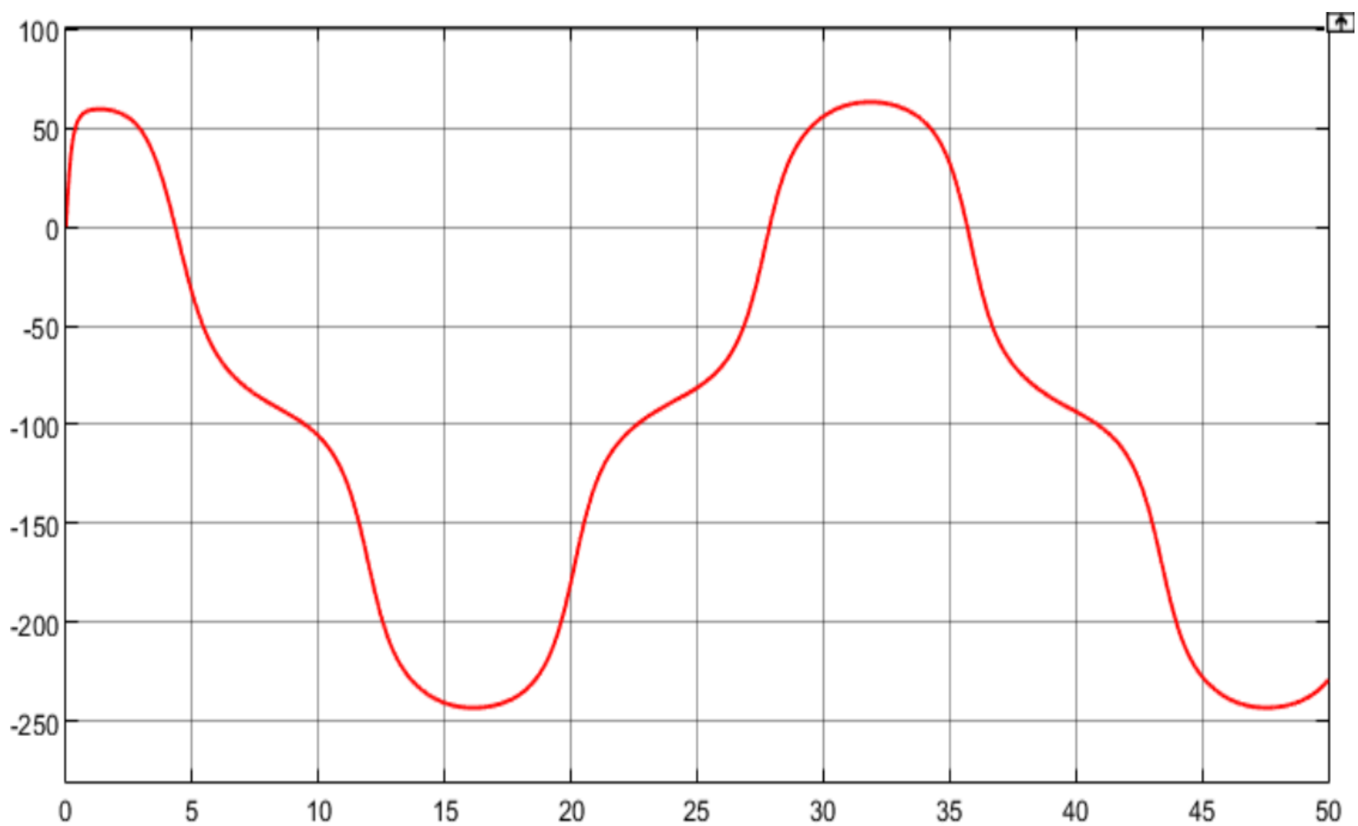

Fig. 14 Tilt angle of TWSBR when tracking a figure-8 trajectory

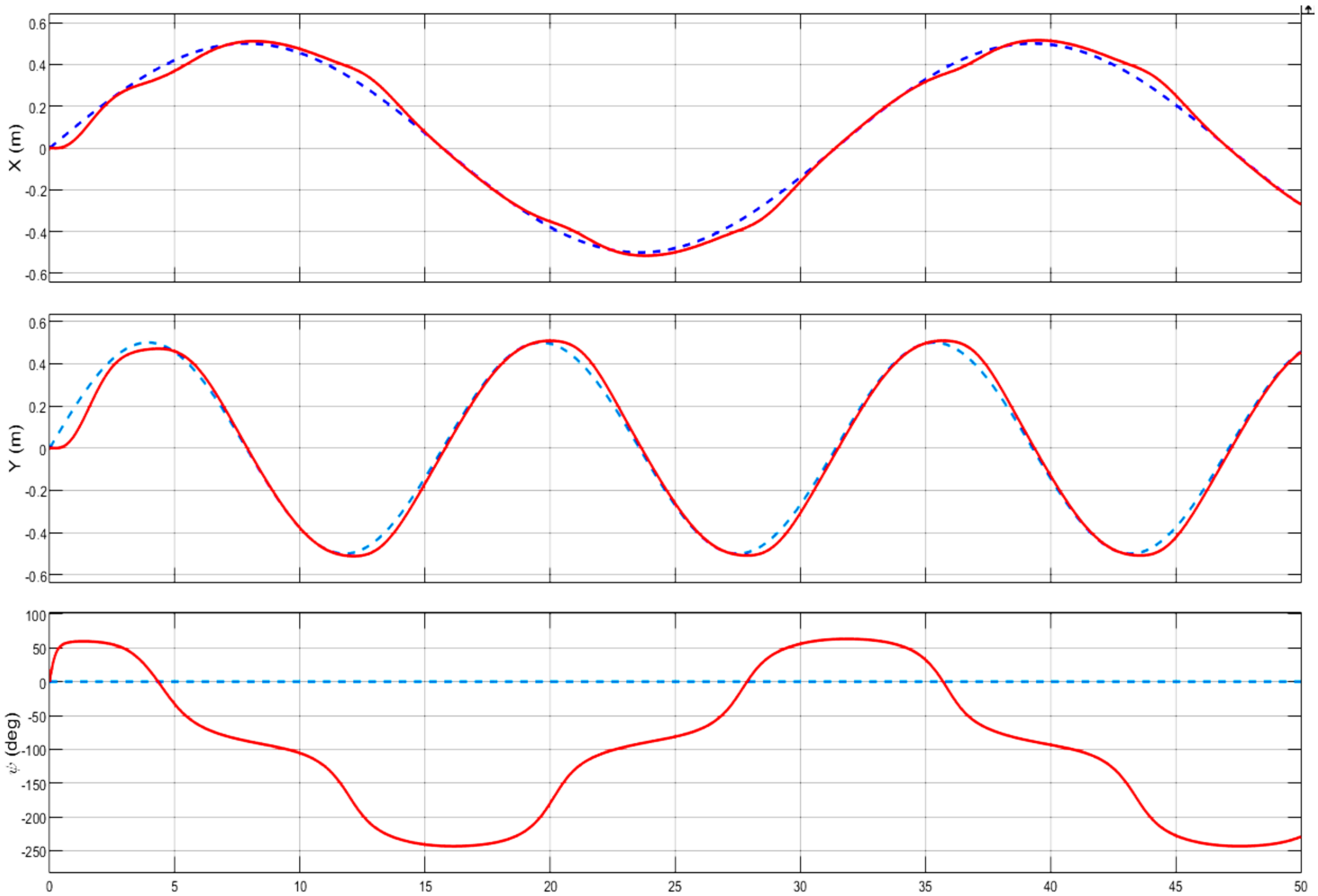

Fig. 15 Error in tracking x, y-axis trajectory and robot tilt angle

According to the simulation results from Figures 12 to 15 above, TWSBR has moved to track the Figure 8 orbit. The orbit tracking error of TWSBR compared to the reference orbit is relatively small in the range of $\pm 0.002m$. The parameters of the tilt angle, wheel angle, and rotation angle have been tracked to a set position of 0.

5. Conclusion

The article has successfully built a simulation program on Matlab/Simulink software for a self-balancing two-wheeled robot system. TWSBR control is stable around the balance position and follows the reference orbit with good quality and small errors. The stability of the system is rigorously proven mathematically using Lyapunov stability theory. The simulation results are performed on a mathematical model of TWSBR with the impact of external disturbances and with uncertain nonlinear model parameters. The results show the possibility of applying these controllers to real application systems.

Acknowledgements

This study was supported by the University of Economics - Technology for Industries, Ha Noi - Vietnam; http://www.uneti.edu.vn/.

References

[1] Suhardi Azliy Bin Junoh, "*Two-Wheeled Balancing Robot Controller Designed Using PID*," Universiti Tun Hussein Onn Malaysia, 2015. [Google Scholar] [Publisher Link]

[2] Jorge Solis et al., "Development of the Two-Wheeled Inverted Pendulum Type Mobile Robot WV-2R for Educational Purposes," *2009 IEEE/RSJ International Conference on Intelligent Robots and* Systems, St. Louis, MO, USA, pp. 2347-2352, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[3] Vicky Mudeng et al., "Design and Simulation of Two-Wheeled Balancing Mobile Robot with PID Controller," *International Journal of Sustainable Transportation Technology*, vol. 3, no. 1, pp. 12-19, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[4] Junfeng Wu, and Wanying Zhang, "Design of Fuzzy Logic Controller for Two-Wheeled Self-Balancing Robot," *Proceedings of 2011 6th International Forum on Strategic Technology*, Harbin, China, pp. 1266-1270, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[5] Yan-Hou Wen, Yu-Sheng Lin, and Yih-Guang Leu, "Design and Implementation of the Balance of Two-Wheeled Robots," *2013 International Conference on Advanced Robotics and Intelligent Systems*, Tainan, Taiwan, pp. 130-134, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[6] Junfeng Wu, Wanying Zhang, and Shengda Wang, "A Two-Wheeled Self-Balancing Robot with the Fuzzy PD Control Method," *Mathematical Problems in Engineering*, pp. 1-13, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[7] M.A. Akmal, N.F. Jamin, and N.M. Abdul Ghani, "Fuzzy Logic Controller for Two Wheeled EV3 LEGO Robot," *2017 IEEE Conference on Systems, Process and Control*, Meleka, Malaysia, pp. 134-139, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[8] Shui-Chun Lin, Ching-Chih Tsai, and Wen-Lung Luo, "Adaptive Neural Network Control of a Self-balancing Two-wheeled Scooter," *IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society*, Taipei, Taiwan, pp. 868-873, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[9] Chenxi Sun, Tao Lu, and Kui Yuan, "Balance Control of Two-Wheeled Self-Balancing Robot Based on Linear Quadratic Regulator and Neural Network," *2013 Fourth International Conference on Intelligent Control and Information Processing*, Beijing, China, pp. 862-867, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[10] Ali Unluturk, and Omer 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, pp. 1-9, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[11] Arunit Maity, and Sarthak Bhargava, "Design and Implementation of a Self-Balancing Two-Wheeled Robot Driven by a Feed-Forward Backpropagation Neural Network," *International Research Journal of Engineering and Technology*, vol. 7, no. 9, pp. 3876-3881, 2020. [Google Scholar] [Publisher Link]

[12] Fuquan Dai et al., "A Two-Wheeled Inverted Pendulum Robot with Friction Compensation," *Mechatronics*, vol. 30, pp. 116-125, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[13] Carlos Aguilar Ibañez, O. Gutiérrez Frias, and M. Suárez Castañón, "Lyapunov-Based Controller for the Inverted Pendulum Cart System," *Nonlinear Dynamics*, vol. 40, pp. 367-374, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[14] O. Octavio Gutiérrez Frías, "Lyapunov Method for the Controlling of the Two Wheels Inverted Pendulum," *2011 8th International Conference on Electrical Engineering, Computing Science and Automatic Control*, Merida City, Mexico, pp. 1-5, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[15] Chong Chee Soon et al., "Sliding Mode Controller Design with Optimized PID Sliding Surface Using Particle Swarm Algorithm," *Procedia Computer Science*, vol. 105, pp. 235-239, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[16] İlyas Eker, "Second-Order Sliding Mode Control with Experimental Application," *ISA Transactions*, vol. 49, no. 3, pp. 394-405, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[17] Ming Yue et al., "Indirect Adaptive Fuzzy Control for a Nonholonomic/Underactuated Wheeled Inverted Pendulum Vehicle Based on a Data-Driven Trajectory Planner," *Fuzzy Sets and Systems*, vol. 290, pp. 158-177, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[18] Mou Chen, "Robust Tracking Control for Self-Balancing Mobile Robots Using Disturbance Observer," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 3, pp. 458-465, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[19] Nguyen Gia Minh Thao, Duong Hoai Nghia, and Nguyen Huu Phuc, "A PID Backstepping Controller for Two-Wheeled Self-Balancing Robot," *International Forum on Strategic Technology*, Ulsan, Korea (South), pp. 76-81, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[20] Zhijun Li, and Chunquan Xu, "Adaptive Fuzzy Logic Control of Dynamic Balance and Motion for Wheeled Inverted Pendulums," *Fuzzy Sets and Systems*, vol. 160, no. 12, pp. 1787-1803, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[21] Ching-Chih Tsai, Hsu-Chih Huang, and Shui-Chun Lin, "Adaptive Neural Network Control of a Self-Balancing Two-Wheeled Scooter," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 4, pp. 1420-1428, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[22] Liangyong Wang, Tianyou Chai, and Lianfei Zh, "Neural-Network-Based Terminal Sliding-Mode Control of Robotic Manipulators Including Actuator Dynamics," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 9, pp. 3296-3304, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[23] J. Park, and I.W. Sandberg, "Universal Approximation Using Radial-Basis-Function Networks," *Neural Computation*, vol. 3, no. 2, pp. 246–257, 1991. [CrossRef] [Google Scholar] [Publisher Link]

[24] S.S. Ge, and C. Wang, "Adaptive Neural Control of Uncertain MIMO Nonlinear Systems," *IEEE Transactions on Neural Networks*, vol. 15, no. 3, pp. 674–692, 2004. [CrossRef] [Google Scholar] [Publisher Link]

[25] Chenguang Yang et al., "Neural-Learning-Based Telerobot Control with Guaranteed Performance," *IEEE Transactions on Cybernetics*, vol. 47, no. 10, pp. 3148–3159, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[26] Zhijun Li, Chenguang Yang, and Liping Fan, *Advanced Control of Wheeled Inverted Pendulum Systems*, Springer London, pp. 1-218, 2012. [Google Scholar] [Publisher Link]

[27] Rongxin Cui, Ji Guo, and Zhaoyong Mao, "Adaptive Backstepping Control of Wheeled Inverted Pendulums Models," *Nonlinear Dynamics*, vol. 79, no. 1, pp. 501–511, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[28] Vo Ba Viet Nghia et al., "Adaptive Neural Sliding Mode Control for Two Wheel Self Balancing Robot," *International Journal of Dynamics and Control*, vol. 10, no. 771-784, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[29] F. Grasser et al., "JOE: A Mobile, Inverted Pendulum," *IEEE Transactions on Industrial Electronics*, vol. 49, no. 1, pp. 107-114, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[30] Jinkun Liu, *Radial Basis Function (RBF) Neural Network Control for Mechanical Systems: Design, Analysis and Matlab Simulation*, Springer Berlin Heidelberg, pp. 1-365, 2013. [Google Scholar] [Publisher Link]

[31] Shuzhi S. Ge, C.C. Hang, and L.C. Woon, "Adaptive Neural Network Control of Robot Manipulators in Task Space," *IEEE Transactions on Industrial Electronics*, vol. 44, no. 6, pp. 746-752, 1997. [CrossRef] [Google Scholar] [Publisher Link]