Robotica (2024), 1–15
doi:10.1017/S0263574724001437

RESEARCH ARTICLE

Online robust self-learning terminal sliding mode control for balancing control of reaction wheel bicycle robots

Xianjin Zhu[1], Wenfu Xu[2], Zhang Chen[3], Yang Deng[3], Qingyuan Zheng[3], Bin Liang[3] and Yu Liu[1]

[1]School of Mechatronics Engineering, Harbin Institute of Technology, Harbin, China
[2]School of Mechatronics Engineering and Automation, Harbin Institute of Technology, Shenzhen, China
[3]Department of Automation, Tsinghua University, Beijing, China
Corresponding author: Yu Liu; Email: Lyu11@hit.edu.cn

Received: 11 June 2024; **Revised:** 16 July 2024; **Accepted:** 11 August 2024

Keywords: Control of robotic systems; adaptive dynamic programming; sliding mode control; reaction wheel bicycle robots; balancing control

Abstract

This paper proposes an online robust self-learning terminal sliding mode control (RS-TSMC) with stability guarantee for balancing control of reaction wheel bicycle robots (RWBR) under model uncertainties and disturbances, which improves the balancing control performance of RWBR by optimising the constrained output of TSMC. The TSMC is designed for a second-order mathematical model of RWBR. Then robust adaptive dynamic programming based on an actor-critic algorithm is used to optimise the TSMC only by data sampled online. The system closed-loop stability and convergence of the neural network weights are guaranteed based on the Lyapunov analysis. The effectiveness of the proposed algorithm is demonstrated through simulations and experiments.

1. Introduction

In recent years, there has been growing interest in the research of agile and high-speed mobile robots designed for rugged or narrow terrain [1–4]. Among these, bicycle robots have emerged as a promising platform due to their ability to achieve high-speed locomotion and agile manoeuvres on varied terrains. Reaction wheel bicycle robots (RWBR) are a type of bicycle robot that relies on reaction wheels as auxiliary balancing mechanisms. Compared to other auxiliary balancing mechanisms, such as control moment gyroscopes [5, 6] and mass pendulums [7, 8], reaction wheels offer advantages such as simple mechanism design and rapid response [9, 10].

Previous studies have investigated the effects of strategies on the RWBR balancing control. The proportional-integral-derivative (PID) control was designed to stabilise the roll angle [11]. Linear quadratic regulator (LQR) controller was used to achieve balancing control by approximating the linearisation around the equilibrium point [12]. The control of RWBR presents significant challenges, particularly in dealing with inherent uncertainties and disturbances. Traditional control methods often struggle to address these complexities effectively, leading to suboptimal performance and limited adaptability. To address these problems, various robust control strategies were proposed to balance the RWBR, such as robust LQR [13] and disturbance observers [14]. Sliding mode control (SMC) has an excellent ability to deal with uncertainties [15–17], which has been developed for balancing control of RWBR [18–20]. However, the robustness of the sliding mode controller to uncertainties typically comes at the cost of conservative control performance. This trade-off between robustness and control performance remains an open problem.

Many researchers have been striving to combine SMC with other methods to tackle this challenge, such as fuzzy control [18], adaptive control [21] and reinforcement learning [22–24]. A fuzzy sliding mode controller was designed to deal with impulse disturbance and system uncertainty in [18], but the determination of fuzzy rules was rather complicated. In [21], an adaptive sliding mode controller was proposed, which dynamically adjusts the parameters of the sliding mode controller to optimise the performance of the control. This work only make monotonic adjustments in certain scenarios, which may lead to excessively high system gain and more severe chattering. Our previous work has confirmed that reinforcement learning can improve the control performance of the SMC online [22, 23], while this combination cannot provide sufficient theoretical stability guarantee.

Adaptive dynamic programming (ADP) algorithm, a kind of reinforcement learning technique, has been used to address various optimal control problems [25–29]. It not only improves control performance while maintaining robustness but also provides theoretical stability guarantee. The linear controller with the offline ADP algorithm was proposed to balance a bicycle robot in [25]. The online ADP algorithm was studied to deal with the optimal control problem with known dynamics in [28]. Ref. [26] proposed a method to adjust the sliding mode controller of ADP online to optimise the trajectory tracking of mobile robots. However, its online optimisation was based on the prediction of the states of the nominal model, which greatly limits the applicability under uncertainty. In order to directly utilise online data for ADP solutions, researchers have conducted a significant amount of work, which has led to the developments of two main methods. One involves using the model obtained from online data fitting for online prediction [30]. The other directly uses online data to optimise the controller, including integral reinforcement learning [31] and robust adaptive dynamic programming (RADP) [27].

To address above problems, we introduce RADP to optimise the TSMC online for balancing control of RWBR. First, the nonlinear dynamics of RWBR with uncertainties and disturbances are established and the terminal sliding mode controller is set. Then, the problem of optimising the TSMC with stability constraints is formulated. An online actor-critic-based RADP algorithm is proposed to solve optimal control problems. The stability and convergence of the proposed control strategy are proven. The algorithm comparison in simulation demonstrates the advantages of the proposed control strategy. Prototype experiments also validate the control strategy. The main contributions of this paper are summarised as follows.

- An online robust actor-critic-based RADP algorithm with robust self-learning terminal sliding mode control
  (RS-TSMC) is proposed to optimise the control performance while maintain the robustness of balancing controller for RWBR. The optimisation process is directly based on data collected online without the need for system dynamics.
- The controller optimisation problem is transformed into solving the Hamilton–Jacobi–Bellman (HJB) equation, and the system output generated by ADP is constraint according to the range of TSMC parameters. Compared to [26], this mechanism improves the conditions for solving the constrained HJB equation, providing a more flexible and adaptable strategy for designing control strategies.
- Experimental studies conducted in simulation platform and on a prototype RWBR compared with several recently proposed control strategies show the effectiveness of the algorithm proposed in this paper.

The rest of the paper is organised as follows. The dynamics of the RWBR and the problem formulation are given in Section 1. The online self-learning sliding mode control strategy is proposed with the stability and convergence proof in Section 3. In Section 4, various simulation experiments are performed, and the experimental results for a RWBR prototype are presented. The conclusion is addressed in Section 5. The video of the simulation and the experiments of RWBR prototype are available at the following website: https://github.com/ZhuXianjinGitHub/RSTSMC. (accessed on 30 August 2024).

Throughout the paper, $\|\cdot\|$ denotes the Euclidean norm, diag $\{\cdot\}$ represents a diagonal matrix, and $\otimes$ denotes the Kronecker product.

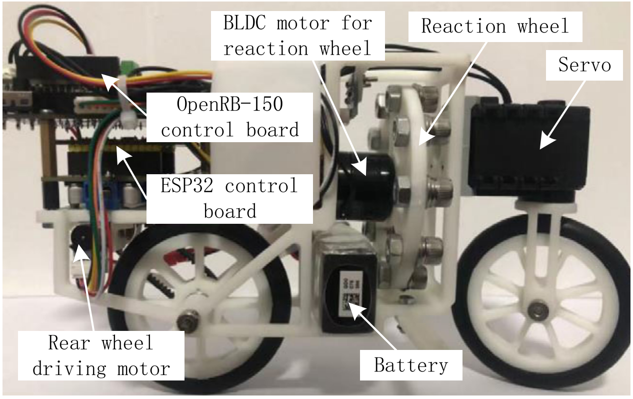

Figure 1. *Side view of the RWBR prototype.*

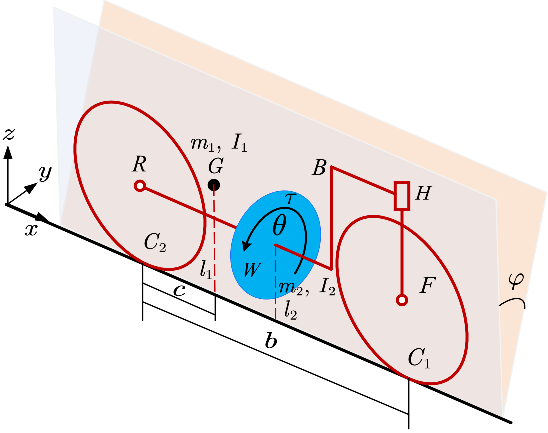

Figure 2. *Notations of the RWBR.*

2. Problem formulation

In this section, the dynamic model of RWBR with uncertainty and disturbance is derived. We also introduce the feedback transformation. In addition, a TSMC is designed. Furthermore, the online optimisation problem for this controller is presented.

2.1. Dynamics model of RWBR

Figure 1 presents the prototype of RWBR, while Figure 2 shows notations. It can be seen that the RWBR consists of five parts, including a rear wheel, body frame, reaction wheel, handlebar and a front wheel (simplified as R, B, W, H and F, respectively) in Figure 2. The details of the notation are shown in Table I.

Following [23], the roll dynamics of the RWBR is presented as follows:

$$\begin{gathered} J\ddot{\varphi} + I_2\ddot{\theta} - Mg\sin(\varphi) = d_1 \\ I_2\ddot{\varphi} + I_2\ddot{\theta} = \tau + d_2 \end{gathered} \tag{1}$$

where $J = m_1 l_1^2 + m_2 l_2^2 + I_1 + I_2$, $M = m_1 I_1 + m_2 I_2$, d_1 and d_2 represent unmodelled dynamics and uncertainty.

https://doi.org/10.1017/S0263574724001437 Published online by Cambridge University Press

Table I. *Diagram of bicycle structure.*

Symbols	Meanings
b	Wheelbase
G	Mass centre of the parts (R, B, H, and F)
c	Distance from C_2 to projection of G
l_1	Height of the mass centre G
l_2	Height of the mass centre of reaction wheel$G$
φ	Roll angle of the RWBR
θ	Angular velocity of the W
m_1, I_1	Mass and moment of inertia of the parts (R, B, H, and F)
m_2, I_2	Mass and moment of inertia of the part (W)
τ	Torque for balancing control

To make full use of the known dynamics of the system, the dynamics parameters are divided into a nominal part and an uncertainty part.

$$\begin{aligned} \Delta J &= \left|J - J_N\right| < \overline{\Delta J} \\ \Delta I_2 &= \left|I_2 - I_{2N}\right| < \overline{\Delta I_2} \\ \Delta M &= \left|M - M_N\right| < \overline{\Delta M} \end{aligned} \tag{2}$$

where J_N, I_{2N} and M_N are the nominal parameter values, $\overline{\Delta J}$, $\overline{\Delta I_2}$ and $\overline{\Delta M}$ are the upper bounds of the uncertainties ΔJ, ΔI_2 and ΔM.

Further, equation (1) can be re-written as

$$(J_N - I_{2N})\,\ddot{\varphi} - M_N g \sin(\varphi) = -\tau + d_{1N} - d_{2N} \tag{3}$$

where $d_{1N} = d_1 + \Delta M g \sin(\varphi) - \Delta J\ddot{\varphi} - \Delta I_2\ddot{\theta}$ and $d_{2N} = d_2 - \Delta I_2\ddot{\varphi} - \Delta I_2\ddot{\theta}$.

2.2. Design of TSMC controller

For the controller design, we first define φ_d as the reference roll angle. The φ_d, $\dot{\varphi}_d$ and $\ddot{\varphi}_d$ can be obtained as shown in our previous work [23]. Based on the Olfati–Saber transformation mentioned in [33], the following state variables and the feedback transformation are classified.

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u + d^* \end{aligned} \tag{4}$$

where $x_1 = \varphi - \varphi_d$, $x_2 = \dot{\varphi} - \dot{\varphi}_d$, $u = \frac{I_{2N}}{(J_N - I_{2N})} M_N g \sin(x_1) - \ddot{\varphi}_d - \frac{I_{2N}}{(J_N - I_{2N})}\tau$ and $d^* = \frac{I_{2N}}{(J_N - I_{2N})}(d_{1N} - d_{2N})$.

Assumption 1. *Assuming d_1 and d_2 are bounded, it is can be get that d_{1N} and d_{2N} are bounded. Then, it is can be easily proved that d^* is bounded. Consider that $|d^*| < L$, and note that L is an unknown constant.*

The sliding mode surface s, the equivalent control u_{eq} and the reaching control u_r of TSMC are designed according to [32]. The fractional-order terminal attractor replaces the sign item in the classical sliding mode controller, which is beneficial to attenuate chattering.

https://doi.org/10.1017/S0263574724001437 Published online by Cambridge University Press

$$
\begin{aligned}
s &= x_2 + \alpha_0 x_1 + \beta_0 x_1^{q_0/p_0} \\
u_{eq} &= -\left(\alpha_0 \dot{x}_1 + \beta_0 \frac{d}{dt} x_1^{q_0/p_0}\right) \\
u_r &= -\left(\alpha_1 s + \beta_1 s^{q_1/p_1}\right) \\
u_{tsmc} &= u_{eq} + u_r
\end{aligned}
\tag{5}
$$

where $\alpha_i > 0$, $\beta_i > 0$, q_i and p_i $(q_i < p_i)$ $(i = 0, 1)$ are positive odd integers.

By selecting appropriate gains, the system will converge to the sufficiently small neighbourhood of the system equilibrium in finite time. According to [32], $\beta_1 = \frac{L}{\left|s^{q_1/p_1}\right|} + \gamma$ and $\gamma > 0$, the sliding mode variable will reach the neighbourhood $|s| < \left(\frac{L}{\beta_1}\right)^{p_1/q_1}$ of the equilibrium in finite time t_s.

$$
t_s = \frac{p_1}{\alpha_1 (p_1 - q_1)} ln \frac{\alpha_1 s (0)^{(p_1 - q_1)/p_1} + \gamma}{\gamma}
\tag{6}
$$

Then, define $\xi_s = \left|\left(\frac{L}{\beta_1}\right)^{p_1/q_1}\right| < L\prime$,

$$
\dot{x}_1 = -\alpha_0 x_1 - \beta_0 x_1^{q_0/p_0} + L\prime
\tag{7}
$$

the system state x_1 will converge to the sufficiently small neighbourhood $|x_1| < \left(\frac{L\prime}{\beta_0}\right)^{p_0/q_0}$ of the system equilibrium in finite time t_{x_1} the system equilibrium in finite time with $\beta_0 = \frac{L\prime}{\left|x_1^{q_0/p_0}\right|} + \gamma\prime$, $\gamma\prime > 0$.

$$
t_{x_1} = \frac{p_0}{\alpha_0 (p_0 - q_0)} ln \frac{\alpha_0 x_1 (0)^{(p_0 - q_0)/p_0} + \gamma\prime}{\gamma\prime}
\tag{8}
$$

Remark 1. The parameters α_1 and β_1 influence the reaching process of sliding mode variables. The larger parameters can reduce the time required for convergence and improve the robustness of the controller to uncertainties, while the burden of the actuator is increased and the performance of the controller is more conservative. In this paper, the RADP is introduced to online tune parameters α_1 and β_1 of the TSMC controller (5) with constraints $\kappa = [\Delta\alpha_1, \Delta\beta_1]^T$. The main motivation is to improve the control performance while maintain stability and robustness.

Assumption 2. *Assuming $\kappa \in \mathcal{K} = \{\kappa_{i\min} \leqslant \kappa_i \leqslant \kappa_{i\max}\}$, $(i = 1, 2)$. $\mathcal{K}$ is set to guarantee the finite-time convergence. $\mathcal{K}$ and L generally can be obtained through experiments. And the the stability proof is given in [26].*

3. Online robust self-learning TSMC

In this section, an online robust self-learning TSMC for RWBR is proposed to improve the control performance and retain the robustness. First, the optimal control problems with stability constraints are formulated. Then, an online actor-critic-based RADP algorithm is designed to approximate the HJB solutions.

Define u_{adp} as the self-learning part of the control, the output of the controller as follows:

$$
\begin{aligned}
u &= u_{tsmc} + u_{adp} \\
[\kappa_{1\,\min}, \kappa_{2\,\min}]\,\zeta < \left|u_{adp}\right| &< [\kappa_{1\,\max}, \kappa_{2\,\max}]\,\zeta
\end{aligned}
\tag{9}
$$

where $\zeta = \left[s, s^{q_1/p_1}\right]^T$.

Taking (9) into (4), the system can be written as

$$
\dot{X} = AX + Bu + D
\tag{10}
$$

where $X=\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $A=\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $B=\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and $D=\begin{bmatrix} 0 \\ d^* \end{bmatrix}$.

The optimal problem is considered to be solved by minimising the value function V_c to obtain the optimal policy function u. V_c is defined as

$$V_c=\int_0^\infty \left(X^TQX+r\left(u_{tsmc}+u_{adp}\right)^2\right)dt, X\left(0\right)=X_0 \tag{11}$$

where Q is symmetric positive definite matrices and r is a positive constant. Taking the derivative of (11) along the trajectory of (10), the following Hamiltonian function can be obtained

$$H=V_{cX}^T\dot{X}+X^TQX+r\left(u_{tsmc}+u_{adp}\right)^2 \tag{12}$$

where $V_{cX}=\frac{\partial V_c}{\partial X}$. Define $V_c^*=\min_{U\prime}\left(V_c\right)$ to denote the optimal value function, which satisfies

$$0=H^*=\min_{u_{adp}}\left\{H\right\}=V_{cX}^{*T}\dot{X}+X^TQX+r\left(u_{tsmc}+u_{adp}\right)^2 \tag{13}$$

where $V_{cX}^*=\frac{\partial V_c^*}{\partial X}$. Assuming the minimum of (13) exists and is unique, then we can obtain the optimal control policy $u_{adp}^*=arg\min_{u_{adp}}\left\{H\right\}$ by $\frac{\partial H}{\partial u_{adp}}=0$, which is described as

$$u_{adp}^*=-\frac{1}{2r}V_{cX}^{*T}B-u_{tsmc} \tag{14}$$

Taking (14) into (13),

$$0=V_{cX}^{*T}\dot{X}+X^TQX+r\left(-\frac{1}{2r}V_{cX}^{*T}B-u_{tsmc}\right)^2 \tag{15}$$

Traditionally, (15) is difficult to get the solution directly. The policy iteration algorithm [34] is adopted to iteratively solve in traditional ADP by the following two steps:

a) given $u^{(i)}$, solve for the $V_c^{(i)}$ using

$$\begin{aligned} 0=V_{cX}^{(i)T}\dot{X}+X^TQX+r\left(u_{tsmc}+u_{adp}{}^{(i)}\right)^2 \\ V_c^{(i)}\left(0\right)=0 \end{aligned} \tag{16}$$

b) update the control policy using

$$u_{adp}^{(i+1)}=-\frac{1}{2r}V_{cX}^{(i)T}B-u_{tsmc} \tag{17}$$

where $i=1,2,\cdots$ denotes the iterations. When $i\to\infty$, then $V_c\to V_c^*$, $u_{adp}\to u_{adp}^*$.

It can be seen that the system dynamic is needed in (16) to get $\dot{X}$. When there is a certain deviation between the nominal model of the system and the actual scene, the optimisation effect based on the nominal model of the system may be affected. In this paper, RADP [27] is used to solve the optimal control problem only by data sampled online.

Consider an arbitrary control input $u=u_{tsmc}+u_s$ and differentiate the value function $V_c^{(i)}$.

$$\begin{aligned} \dot{V}_c^{(i)}=V_{cX}^{(i)T}\left(AX+B\left(u_{tsmc}+u_{adp}{}^{(i)}\right)+B\left(u_s-u_{adp}{}^{(i)}\right)\right) \\ =-2r\left(u_{tsmc}+u_{adp}^{(i+1)}\right)\left(u_s-u_{adp}{}^{(i)}\right)-X^TQX-r\left(u_{tsmc}+u_{adp}{}^{(i)}\right)^2 \end{aligned} \tag{18}$$

Integral (18) over an arbitrary interval as follows,

$$\begin{aligned} V_c^{(i)}\left(X_t\right)-V_c^{(i)}\left(X_{t-T}\right)= \\ -\int_{t-T}^t\left(2r\left(u_{tsmc}+u_{adp}^{(i+1)}\right)\left(u_s-u_{adp}{}^{(i)}\right)+X^TQX+r\left(u_{tsmc}+u_{adp}{}^{(i)}\right)^2\right)d\tau \end{aligned} \tag{19}$$

https://doi.org/10.1017/S0263574724001437 Published online by Cambridge University Press

The closed-loop stability of the system is ensured by (9). $V_c^{(i)}$ and the improved policy $u_{adp}^{(i+1)}$ can be obtained in one calculation, and it does not need knowledge of the system dynamics.

The value function and the policy function are defined as neural network (NN),

$$\hat{V}_c^* (X) = \hat{W}_c^T \phi (X) \tag{20}$$

$$\hat{u}_{adp}^* (X) = \hat{W}_a^T \varphi (X) \tag{21}$$

After inserting into (19),

$$\begin{aligned} \epsilon (t) = \hat{W}_c^T (\phi (x_t) - \phi (x_{t-T})) + \\ \int_{t-T}^{t} \left(2r \left(u_{tsmc} + \hat{W}_a^T \varphi (X)\right) \left(u_s - \hat{W}_a^T \varphi (X)\right) + X^T Q X + r \left(u_{tsmc} + \hat{W}_a^T \varphi (X)\right)^2\right) d\tau \end{aligned} \tag{22}$$

Under the gradient descent method, the updating laws for the weights of the critic NN and the actor NN as follows,

$$\dot{\hat{W}}_c = -\lambda_1 \frac{\phi (x_t) - \phi (x_{t-T})}{m_s^2 (t)} \epsilon (t) \tag{23}$$

$$\dot{\hat{W}}_a = -\lambda_2 \frac{\eta (t)}{m_s^2 (t)} \epsilon (t) \tag{24}$$

where $\eta (t) = 2 \int_{t-T}^{t} ((r u_s) \otimes \varphi (x)) \, d\tau - \int_{t-T}^{t} (\varphi (x) \otimes r) \otimes \varphi (x) \, d\tau \, \mathbf{vec} \left(\hat{W}_a^T\right)$, $m_s = (\phi(x_t) - \phi(x_{t-T}))^T (\phi (x_t) - \phi (x_{t-T})) + \eta^T \eta + 1$ and m_s is used for normalization.

Remark 2. The differences between RS-TSMC proposed in this paper and self-TSMC (S-TSMC) in [26] and R-TSMC (rosbust-TSMC) in [27] are listed as follows. First, the optimisation process in S-TSMC is based on the state prediction of the nominal model, which is not conducive to the online application of the algorithm. To address this problem, this paper employs an iterative form of RADP to optimise TSMC using online data. Second, the optimisation in S-TSMC is performed directly for the state variable s, which is not exactly equivalent to the optimisation for the state X. The optimisation objective in R-TSMC considers only the part of the u_{adp} and not the overall output of the controller. However, the optimisation is directly based on the system state and controller output in RS-TSMC. Third, the optimisation solution in S-TSMC is performed provided that the constraints of the HJB equations have a solution, whereas R-TSMC does not consider the constraints, but RS-TSMC first solves the unconstrained problem of the HJB and subsequently constrain the controller outputs.

The proposed control strategy schemes are illustrated in Algorithm 1 and Figure 3. The stability and the convergence of the proposed control strategy are given in the Appendix.

Algorithm 1. Online robust self-learning TSMC for RWBR

```
1: Initialize state $\hat{W}_c$ (0), and $\hat{W}_a$ (0), compute $X_0$ using (4).
2: Compute $V_c (t)$ using (11).
3: Update $\hat{W}_c (t)$ and $\hat{W}_a (t)$ using (23)(24).
4: Compute $u_{tsmc}$ using (5).
5: Compute $\hat{u}_{adp}$ using (21).
6: Compute $\xi_{\max}$ and $\xi_{\min}$ using (5), saturate $\hat{u}_{adp}$ satisfies (9).
7: Compute $\tau$ using (4) and propagate $t, X (t)$.
8: Repeat 2-7.
```

https://doi.org/10.1017/S0263574724001437 Published online by Cambridge University Press

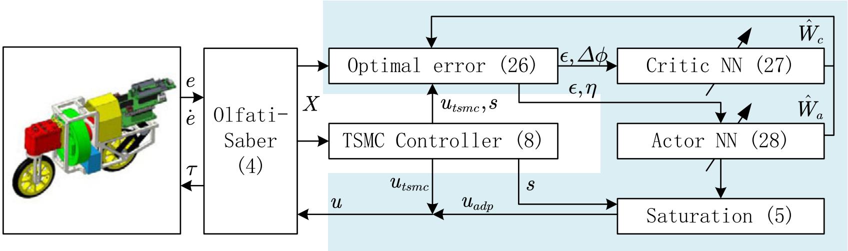

Figure 3. *Schematic of control system.*

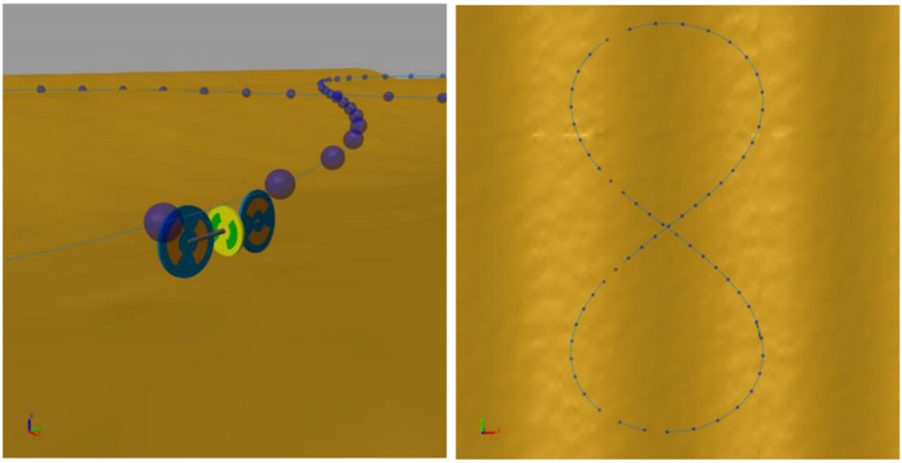

Figure 4. *Simulation environment of RWBR in Matlab Simscape.*

4. Simulations and experiments

4.1. Simulations

In order to demonstrate the effectiveness of the RS-TSMC controllers proposed in this paper, two cases built in a simulation platform shown in Figure 4 as one of our previous works [23]. And two recently developed methods: S-TSMC [26] and R-TSMC [27] are used for comparison. The other simulation factors are the same except for the distinction mentioned in Remark 2. The RWBR is placed on a curved pavement with white noise. The nominal parameters of RWBR are $J_N = 0.0368$, $I_{2N} = 0.0035$ and $M_N = 0.2544$. The true parameters of RWBR used for simulation are $J = 0.033$, $I_2 = 0.0040$ and $M = 0.2742$. The control period of the controllers is 0.01s. The other parameters are given as follows:

$$\begin{aligned} Q &= \text{diag}\left\{1, 1\right\}, r = 1 \\ \alpha_0 = 3, \alpha_1 = 2, \beta_0 = 1, \beta_1 = 1, q_0 = q_1 &= 17 \textit{ and } p_0 = p_1 = 19 \\ \kappa_{1\max} = \kappa_{2\max} = 0.8, \kappa_{1\min} &= \kappa_{2\min} = -0.8 \\ \lambda_1 = 0.2, \lambda_2 &= 0.1 \end{aligned} \tag{25}$$

The activation functions of the critic NN and the actor NN are considered as

$$\begin{aligned} \phi\left(X\right) &= \left[x_1^2, x_2^2, x_1x_2, x_1^4, x_2^4, x_1^2x_2^2\right]^T \\ \varphi\left(X\right) &= \left[x_1, x_2, x_1^2, x_2^2, x_1x_2, x_1^4, x_2^4, x_1^2x_2^2\right]^T \end{aligned} \tag{26}$$

https://doi.org/10.1017/S0263574724001437 Published online by Cambridge University Press

***Table II.** Assessment of control performance under different cases.*

	RS-TSMC	TSMC	R-TSMC [27]	S-TSMC [26]
Case 1	9.11	15.13	13.96	14.16
Improvement 1	39.79%	–	7.73%	6.41%
Case 2	48.99	58.26	51.51	56.85
Improvement 2	15.91%	–	11.59%	2.42%

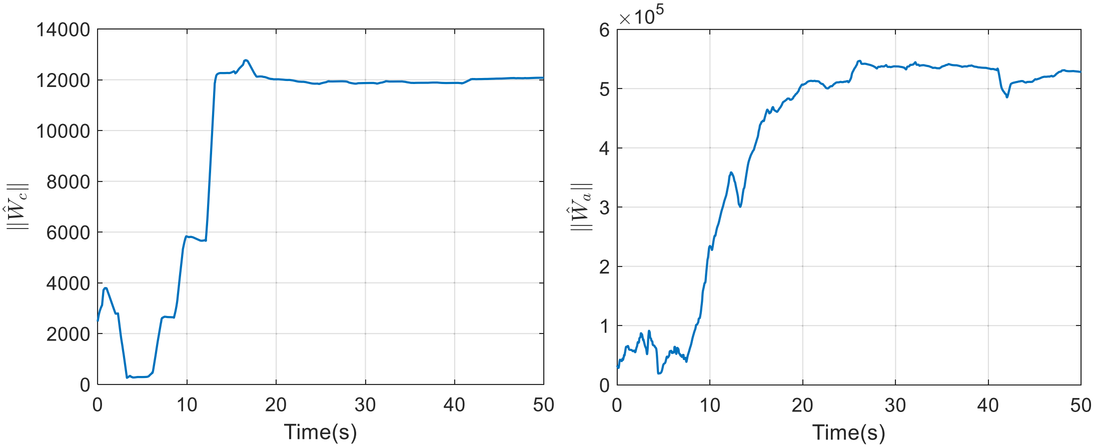

***Figure 5.** $\|\hat{W}_c\|$ and $\|\hat{W}_a\|$ with respect to the time under RS-TSMC in case 1.*

An overturning moment d_2 is added to the system of RWBR. In case 1,

$$d_2 = 0.02 \sum_j \sin(jt) \tag{27}$$

where $j = \begin{bmatrix} 1, 3, 7, 11, 13, 15 \end{bmatrix}$. In case 2,

$$d_2 = 0.02 \sum_j \sin(jt) + \begin{cases} 0.2, t \in [10, 12) \cup t \in [30, 32) \\ -0.2, t \in [20, 22) \cup t \in [40, 42) \\ 0, else \end{cases} \tag{28}$$

To clearly demonstrate the superiority of the proposed method, V_c defined in (11) are used to quantitatively estimate the performance, which are shown in Table II. As seen in this table, RS-TSMC reduced the criteria by 39.79% in case 1 and by 15.91% in case 2 to TSMC. It is less than the other two recently developed methods (R-TSMC, S-TSMC), which implies that the proposed method can achieve better control performance with less control effort. Then, details of the simulations of the two cases are discussed.

The simulation results of Case 1 are demonstrated in Figure 5 and Figure 6. Figure 5 gives the norms $\left\|\hat{W}_c\right\|$ and $\left\|\hat{W}_a\right\|$ with respect to the time under RS-TSMC. As shown in Figure 5, $\left\|\hat{W}_c\right\|$ converges after 12 s, and $\left\|\hat{W}_a\right\|$ converges after 20 s. Figure 6 gives the states, the control output, and V_c of four methods. As can be seen, the proposed method has the smallest value of V_c among the four controllers. In sum, it can be concluded that the control performance of the proposed method (RS-TSMC) outperforms the other three methods, which illustrates the superiority of the proposed method.

Figure 7 gives the norm $\left\|\hat{W}_c\right\|$ and $\left\|\hat{W}_a\right\|$ with respect to the time under RS-TSMC. The pulse perturbation has a significant effect on $\left\|\hat{W}_c\right\|$ at 10 s. The $\left\|\hat{W}_a\right\|$ shows regular changes with the pulse

https://doi.org/10.1017/S0263574724001437 Published online by Cambridge University Press

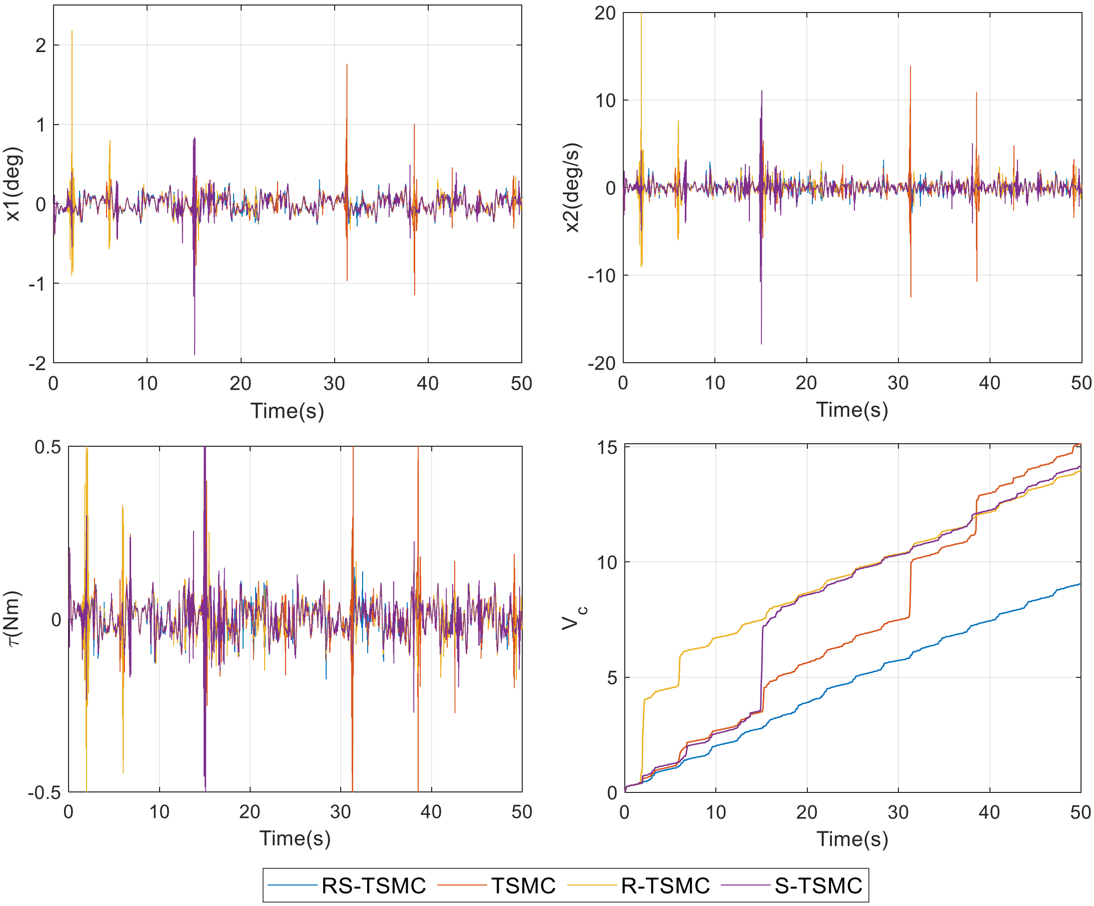

Figure 6. *The states, output and V_c with respect to the time under four different algorithms in case 1.*

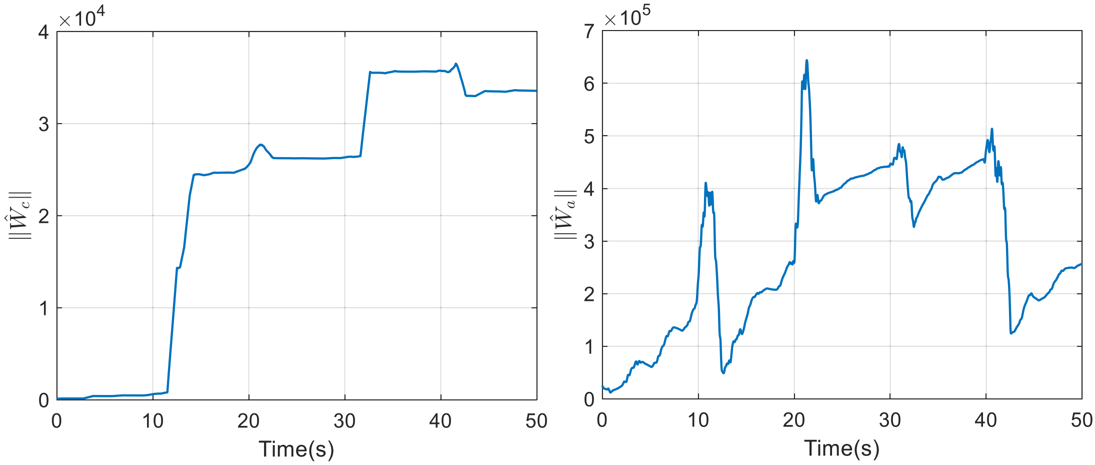

Figure 7. *$\|\hat{W}_c\|$ and $\|\hat{W}_a\|$ with respect to the time under RS-TSMC in case 2.*

https://doi.org/10.1017/S0263574724001437 Published online by Cambridge University Press

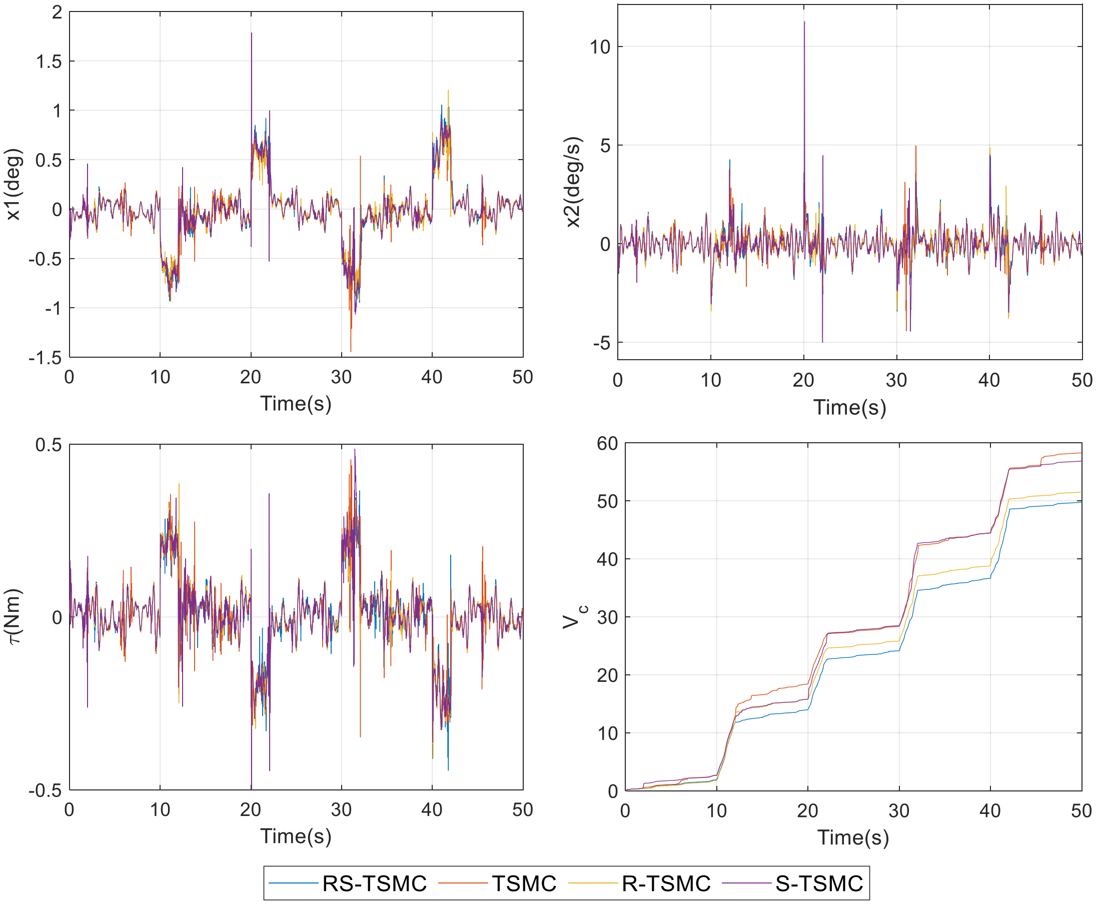

Figure 8. *The states, output and V_c with respect to the time under four different algorithms in case 2.*

disturbance, indicating the regulation effect of the online learning algorithm on the controller output. Figure 8 illustrates the simulation results in Case 2. Similarly, we can conclude that the better control performance is reached and the less control effort is needed with the proposed method in this case.

4.2. Experiments

The RWBR prototype is used to verify the effectiveness of the proposed controller in this subsection. We presented the experiment results of the proposed RS-TSMC controller. We also performed TSMC, R-TSMC and S-TSMC for performance comparisons, which can be found in Figure 9. In the experimental studies, the TSMC algorithm works on ESP32 control board at 50 Hz and the optimising algorithm works on a PC at 25 Hz. Wireless data transmission between ESP32 and PC is achieved via UDP communication protocol. We consider the swing of the handlebars to generate disturbances for the control of the roll angel. The other settings are the same as in the simulations.

Figure 9 demonstrates the experimental results. Within the first 10 s, it can be seen that the V_c of the three optimisation algorithms is slightly higher than that in TSMC, which can also be seen from the curves of x_1, x_2 and τ. The reasons may be as follows: 1) The experimental factors such as initial roll Angle and initial roll angular velocity of RWBR are not completely consistent in different experiments. 2) The processing power of RWBR and PC is limited. With the iterative optimisation of the controller, it is only after 15 s that the three optimisation algorithms gradually outperform TSMC.

https://doi.org/10.1017/S0263574724001437 Published online by Cambridge University Press

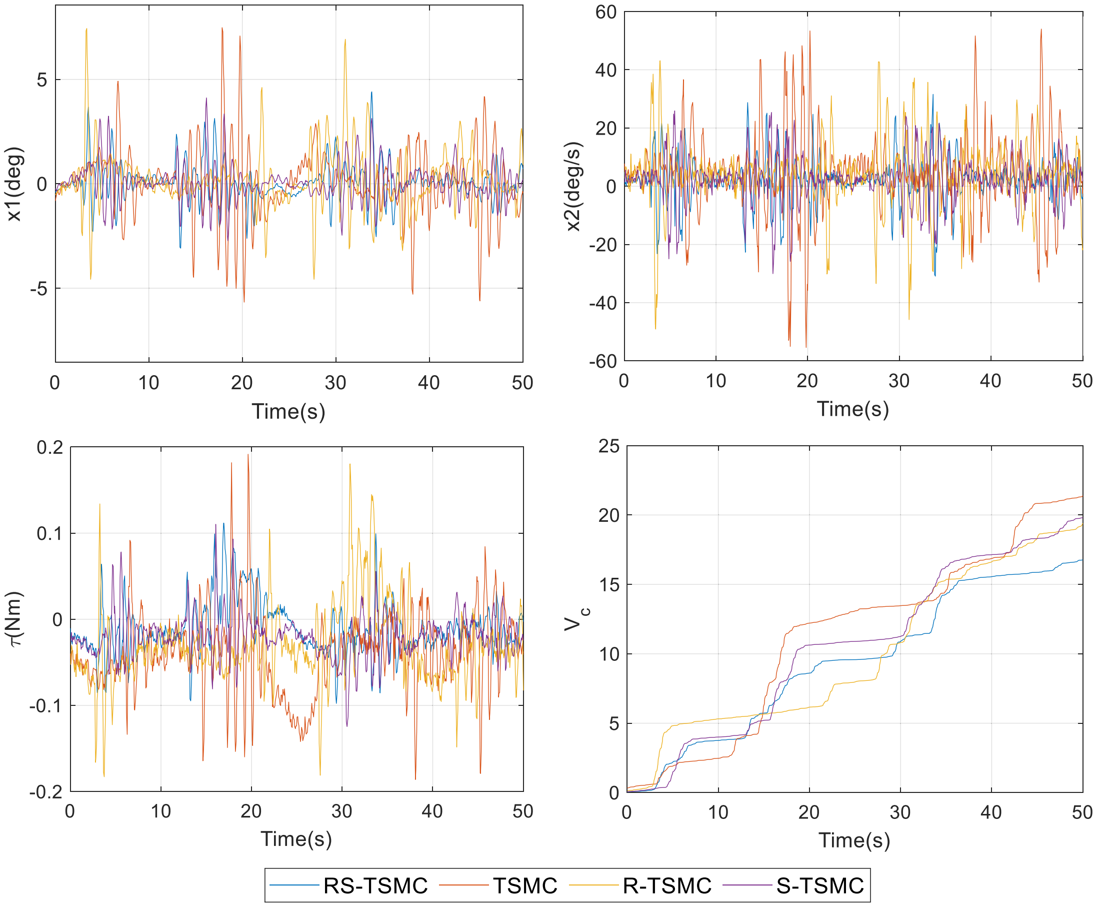

Figure 9. *The states, output and V_c with respect to the time under four different algorithms of RWBR prototype.*

The main reason lies in the fact that the control period in the RWBR prototype is much lower than that in the simulation experiment. In addition, it is not difficult to find that RS-TSMC almost outperforms the other two optimisation algorithms throughout the experiment. The proposed controller (RS-TSMC) reduced the criteria by 21.79%, while R-TSMC and S-TSMC reduced by about 10% to TSMC. The experimental results also validate the effectiveness and feasibility of the proposed control strategy.

5. Conclusions

This paper proposes an online RS-TSMC with stability guarantee for balancing control of RWBR under uncertainties, which improves the balancing control performance of RWBR by optimising the constrained output of TSMC. The robust adaptive dynamic programming (RADP) is used to optimise the TSMC only based on data sampled online without system dynamic. The constraint on the parameters of the sliding mode controller is utilised to derive the constraint on the control output at each time step to maintain the stability of the closed-loop system. Experimental studies conduct a simulate platform and on a prototype RWBR compared with several recently proposed control strategies show the effectiveness of the algorithm proposed in this paper.

https://doi.org/10.1017/S0263574724001437 Published online by Cambridge University Press

Author contributions. Conceptualization and methodology, X.Z. (Xianjin Zhu); software, X.Z. (Xianjin Zhu); validation, W.X., Q.Z., Y.D.; writing – original draft preparation, X.Z. (Xianjin Zhu); writing – review and editing, W.X. and Z.C.; visualisation, Q.Z.; supervision, Y.L.; project administration, Y.L. and B.L.; funding acquisition, Z.C. and Y.D. All authors have read and agreed to the published version of the manuscript.

Financial support. This research was funded by the National Natural Science Foundation of China (62203252, 52205008).

Competing interests. The authors declare no conflicts of interest exist.

Ethical approval. Not applicable.

References

[1] F. Rubio, F. Valero and C. Llopis-Albert, “A review of mobile robots: Concepts, methods, theoretical framework, and applications,” *Int J Adv Robot Syst.* **16**(2), 1–22 (2019).

[2] G. Fadini, S. Kumar, R. Kumar, T. Flayols, A. Del Prete, J. Carpentier and P. Souères, “Co-designing versatile quadruped robots for dynamic and energy-efficient motions,” *Robotica* **42**(6), 2004–2025 (2024).

[3] Y. Huang, Q. Liao, L. Guo and S. Wei, “Simple realization of balanced motions under different speeds for a mechanical regulator-free bicycle robot,” *Robotica* **33**(9), 1958–1972 (2015).

[4] J. Huang, M. Zhang, S. Ri, C. Xiong, Z. Li and Y. Kang, “High-order disturbance-observer-based sliding mode control for mobile wheeled inverted pendulum systems,” *IEEE T Ind Electron.* **67**(3), 2030–2041 (2020).

[5] A. Beznos, A. Formal’sky, E. Gurfinkel, D. Jicharev, A. Lensky, K. Savitsky and L. Tchesalin, “Control of autonomous motion of two-wheel bicycle with gyroscopic stabilisation,” **In:** *IEEE International Conference on Robotics and Automation*, Leuven, Belgium, (1998) pp. 2670–2675.

[6] C.-K. Chen, T.-D. Chu and X.-D. Zhang, “Modeling and control of an active stabilizing assistant system for a bicycle,” *Sensors* **19**(2), 248 (2019).

[7] L. Keo and M. Yamakita, “Controller design of an autonomous bicycle with both steering and balancer controls,” **In:** *IEEE International Conference on Control Applications/International Symposium on Intelligent Control*, St Petersburg, Russia (2009) pp. 1294–1299.

[8] K. He, Y. Deng, G. Wang, X. Sun, Y. Sun and Z. Chen, “Learning-based trajectory tracking and balance control for bicycle robots with a pendulum: A gaussian process approach,” *IEEE-ASME T Mech.* **27**(2), 634–644 (2022).

[9] K. Kanjanawanishkul, “LQR and MPC controller design and comparison for a stationary self-balancing bicycle robot with a reaction wheel,” *Kybernetika* **51**(1), 173–191 (2015).

[10] S. Wang, L. Cui, J. Lai, S. Yang, X. Chen, Y. Zheng, Z. Zhang and Z.-P. Jiang, “Gain scheduled controller design for balancing an autonomous bicycle,” **In:** *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Electr Network (2020-2021) pp. 7595–7600.

[11] H.-W. Kim, J.-W. An, H.D Yoo and J.-M. Lee, “Balancing control of bicycle robot using pid control,” **In:** *13th International Conference on Control, Automation and Systems (ICCAS)*, Gwangju, South Korea (2013) pp. 145–147.

[12] C. Xiong, Z. Huang, W. Gu, Q. Pan, Y. Liu, X. Li and E. X. Wang, “Static balancing of robotic bicycle through nonlinear modeling and control,” **In:** *3rd International Conference on Robotics and Automation Engineering (ICRAE)*, Guangzhou, China (2018) pp. 24–28.

[13] A. Owczarkowski, D. Horla and J. Zietkiewicz, “Introduction of feedback linearization to robust lqr and lqi control - analysis of results from an unmanned bicycle robot with reaction wheel,” *Asian J Control* **21**(2), 1028–1040 (2019).

[14] S. Jeong and D. Chwa, “Sliding-mode-disturbance-observer-based robust tracking control for omnidirectional mobile robots with kinematic and dynamic uncertainties,” *IEEE-ASME T Mech* **26**(2), 741–752 (2021).

[15] L. A. Tuan and Q. P. Ha, “Adaptive fractional-order integral fast terminal sliding mode and fault-tolerant control of dual-arm robots,” *Robotica* **42**(5), 1476–1499 (2024).

[16] J. Song, D. W. C. Ho and Y. Niu, “Model-based event-triggered sliding-mode control for multi-input systems: Performance analysis and optimisation,” *IEEE T Cybernetics* **52**(5), 3902–3913 (2022).

[17] A. Behera, B. Bandyopadhyay, M. Cucuzzella, A. Ferrara and X. Yu, “A survey on event-triggered sliding mode control,” *IEEE Journal of Emerging and Selected Topics in Industrial Electronics* **2**(3), 206–217 (2021).

[18] L. Guo, Q. Liao and S. Wei, “Design of fuzzy sliding-mode controller for bicycle robot nonlinear system,” **In:** *IEEE International Conference on Robotics and Biomimetics (ROBIO 2006)*, Kunming, China (2006) pp. 176–180.

[19] M. Alizadeh, A. Ramezani and H. Saadatinezhad, “Fault tolerant control in an unmanned bicycle robot via sliding mode theory,” *IET Cyber-syst Robot.* **4**(2), 139–152 (2022).

[20] L. Chen, B. Yan, H. Wang, K. Shao, E. Kurniawan and G. Wang, “Extreme-learning-machine-based robust integral terminal sliding mode control of bicycle robot,” *Control Eng Pract.* **121**, 105064 (2022).

[21] L. Chen, J. Liu, H. Wang, Y. Hu, X. Zheng, M. Ye and J. Zhang, “Robust control of reaction wheel bicycle robot via adaptive integral terminal sliding mode,” *Nonlinear Dynam.* **104**(3), 2291–2302 (2021).

[22] X. Zhu, Y. Deng, X. Zheng, Q. Zheng, B. Liang and Y. Liu, “Online reinforcement-learning-based adaptive terminal sliding mode control for disturbed bicycle robots on a curved pavement,” *Electronics* **11**(21), 3495 (2022).

https://doi.org/10.1017/S0263574724001437 Published online by Cambridge University Press

[23] X. Zhu, Y. Deng, X. Zheng, Q. Zheng, Z. Chen, B. Liang and Y. Liu, “Online series-parallel reinforcement-learning- based balancing control for reaction wheel bicycle robots on a curved pavement,” *IEEE Access* **11**, 66756–66766 (2023).

[24] B. Huo, L. Yu, Y. Liu and S. Sha, “Reinforcement learning based path tracking control method for unmanned bicycle on complex terrain,” **In:** *IECON. 2023- 49th Annual Conference of the IEEE Industrial Electronics Society*, Singapore, Singapore (2023) pp. 1–6.

[25] L. Guo, H. Lin, J. Jiang, Y. Song and D. Gan, “Combined control algorithm based on synchronous reinforcement learning for a self-balancing bicycle robot,” *ISA T.* **145**, 479–492 (2024).

[26] Q. Ma, X. Zhang, X. Xu, Y. Yang and E. Q. Wu, “Self-learning sliding mode control based on adaptive dynamic programming for nonholonomic mobile robots,” *ISA T.* **142**, 136–147 (2023).

[27] Y. Zhu and D. Zhao, “Comprehensive comparison of online adp algorithms for continuous-time optimal control,” *Artif Intell Rev.* **49**(4), 531–547 (2018).

[28] K. G. Vamvoudakis and F. L. Lewis, “Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem,” *Automatica* **46**(5), 878–888 (2010).

[29] D. Liu, S. Xue, B. Zhao, B. Luo and Q. Wei, “Adaptive dynamic programming for control: A survey and recent advances,” *IEEE T Syst Man Cy-S.* **51**(1), 142–160 (2021).

[30] S. Bhasin, R. Kamalapurkar, M. Johnson, K. G. Vamvoudakis, F. L. Lewis and W. E. Dixon, “A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems,” *Automatica* **49**(1), 82–92 (2013).

[31] K. G. Vamvoudakis, D. Vrabie and F. L. Lewis, “Online adaptive algorithm for optimal control with integral reinforcement learning,” *Int J Robust Nonlin.* **24**(17), 2686–2710 (2014).

[32] S. Yu, X. Yu and M. Zhihong, “Robust global terminal sliding mode control of SISO nonlinear uncertain systems,” **In:** *Proceedings of the 39th IEEE Conference on Decision and Control (Cat 00CH37187)*, vol. 3 (2000) pp. 2198–2203.

[33] M. Spong, P. Corke and R. Lozano, “Nonlinear control of the reaction wheel pendulum,” *Automatica* **37**(11), 1845–1851 (2001).

[34] B. A. Sutton RS. *Reinforcement Learning: An Introduction* (MIT Press, United States, 2018).

Appendix

Define the errors $\tilde{W}_c = W_c - \hat{W}_c$ and $\tilde{W}_a = W_a - \hat{W}_a$, $\tilde{W}_c$, where W_c and W_a represent the ideal coefficients of V_c^* and u_{adp}^*, ε_c and ε_a are the approximation errors.

$$\begin{aligned} V_c^*(X) &= W_c^T \phi(X) + \varepsilon_c \\ u_{adp}^*(X) &= W_a^T \varphi(X) + \varepsilon_a \end{aligned} \tag{29}$$

According to (19),

$$V_c^*(X_t) - V_c^*(X_{t-T}) = -\int_{t-T}^{t} \left(2r\left(u_{tsmc} + u_{adp}^*\right)\left(u_s - u_{adp}{}^*\right) + X^T QX + r\left(u_{tsmc} + u_{adp}{}^*\right)^2\right) d\tau \tag{30}$$

Inserting (29) to (30):

$$\begin{aligned} \left(W_c^T \phi\left(X_t\right) + \varepsilon_c\left(t\right)\right) - \left(W_c^T \phi\left(X_{t-T}\right) + \varepsilon_c\left(t-T\right)\right) = \\ -\int_{t-T}^{t} \left(2r\left(u_{tsmc} + W_a^T \varphi\left(X\right) + \varepsilon_a\right)\left(u_s - W_a^T \varphi\left(X\right) - \varepsilon_a\right) + X^T QX + r\left(u_{tsmc} + W_a^T \varphi\left(X\right) + \varepsilon_a\right)^2\right) d\tau \end{aligned} \tag{31}$$

Then substitude $\tilde{W}_c = W_c - \hat{W}_c$ and $\tilde{W}_a = W_a - \hat{W}_a$ to (22),

$$\begin{aligned} \epsilon\left(t\right) = \left(W_c - \tilde{W}_c\right)\left(\phi\left(x_t\right) - \phi\left(x_{t-T}\right)\right) + \\ \int_{t-T}^{t}\left(2r\left(u_{tsmc} + \left(W_a - \tilde{W}_a\right)\varphi(X)\right)\left(u_s - \left(W_a - \tilde{W}_a\right)\varphi(X)\right) + X^T QX + r\left(u_{tsmc} + \left(W_a - \tilde{W}_a\right)\varphi(X)\right)^2\right) d\tau \end{aligned} \tag{32}$$

Substract (32) from (31),

$$\epsilon\left(t\right) = -\left(\tilde{W}_c\left(\phi\left(x_t\right) - \phi\left(x_{t-T}\right)\right) + \tilde{W}_a \eta\left(t\right) - \int_{t-T}^{t} rW_a \varphi\left(X\right) \tilde{W}_a \varphi\left(X\right) d\tau - \varepsilon_{HJB}\right) \tag{33}$$

where $\varepsilon_{HJB} = -\left[\varepsilon_c\left(t\right) - \varepsilon_c\left(t-T\right)\right] - \int_{t-T}^{t}\left(2r\varepsilon_a\left(u_s - W_a^T \varphi\left(X\right)\right) - r\varepsilon_a^2\right) d\tau$.

https://doi.org/10.1017/S0263574724001437 Published online by Cambridge University Press

Define the Lyapunov candidata $L_y = \frac{1}{2\lambda_1}\tilde{W}_c^T\tilde{W}_c + \frac{1}{2\lambda_2}\tilde{W}_a^T\tilde{W}_a$, its time derivative has,

$$\begin{aligned}\dot{L}_y &= \frac{1}{\lambda_1}\tilde{W}_c^T\dot{\tilde{W}}_c + \frac{1}{\lambda_2}\tilde{W}_a^T\dot{\tilde{W}}_a \\ &= \frac{\epsilon\left(t\right)}{m_s^2\left(t\right)}\left(\tilde{W}_c\left(\phi\left(x_t\right) - \phi\left(x_{t-T}\right)\right) + \tilde{W}_a\eta\left(t\right)\right) \\ &\leqslant - \left\|\frac{\rho\left(t\right)}{m_s\left(t\right)}\tilde{W}\right\|\left[\left\|\frac{\rho\left(t\right)}{m_s\left(t\right)}\tilde{W}\right\| - \left\|\frac{\varepsilon_H}{m_s\left(t\right)}\right\|\right]\end{aligned} \tag{34}$$

where $\rho\left(t\right) = \left[\phi^T\left(x_t\right) - \phi^T\left(x_{t-T}\right), \eta^T\left(t\right)\right]^T$ and $\tilde{W} = \left[\tilde{W}_c^T, \tilde{W}_a^T\right]^T$.

Therefore $\dot{L}_y \leqslant 0$, if $\left\|\frac{\rho(t)}{m_s(t)}\tilde{W}\right\| > \left\|\frac{\varepsilon_H}{m_s(t)}\right\|$, since $\|m_s\left(t\right)\| > 1$. This provides an effective practical bound for$\left\|\rho\left(t\right)\tilde{W}\right\|$, since L decreases. According to the lemma 2 in [28], $\tilde{W}_c$ and $\tilde{W}_a$ are ultimately uniformly bounded.

Cite this article: X. Zhu, W. Xu, Z. Chen, Y. Deng, Q. Zheng, B. Liang and Y. Liu, "Online robust self-learning terminal sliding mode control for balancing control of reaction wheel bicycle robots", Robotica. https://doi.org/10.1017/S0263574724001437