IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 6, 2021, accepted January 19, 2021, date of publication January 21, 2021, date of current version January 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3053396

A Hybrid Tracking Control Strategy for Nonholonomic Wheeled Mobile Robot Incorporating Deep Reinforcement Learning Approach

XUESHAN GAO, RUI GAO, PENG LIANG, QINGFANG ZHANG, RUI DENG, AND WEI ZHU
School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China

Corresponding author: Xueshan Gao (xueshan.gao@bit.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2019YFB1309600.

ABSTRACT Tracking control is an essential capability for nonholonomic wheeled mobile robots (NWMR) to achieve autonomous navigation. This paper presents a novel hybrid control strategy combined mode-based control and actor-critic based deep reinforcement learning method. Based on the Lyapunov method, a kinematics control law named given control is obtained with pose errors. Then, the tracking control problem is converted to a finite Markov decision process, in which the defined state contains current tracking errors, given control inputs and one-step errors. After training with deep deterministic policy gradient method, the action named acquired control inputs is capable of compensating the existing errors. Thus, the hybrid control strategy is obtained under velocity constraint, acceleration constraint and bounded uncertainty. A cumulative error is also defined as a criteria to evaluate tracking performance. The comparison results in simulation demonstrate that our proposed method have an obviously advantage on both tracking accuracy and training efficiency.

INDEX TERMS Deep reinforcement learning, tracking control, kinematics control, hybrid control strategy, nonholonomic wheeled mobile robot.

I. INTRODUCTION

A. RELATED WORKS

The tracking control of wheeled mobile robot is one of the fundamental functions of robot autonomous navigation, which has been widely used in inspection, security, cleaning, planet exploration, military application and so on. It could be classified as a nonlinear system with multi-inputs and multi-outputs, it is also a underactuated system with uncertainty actually. In addition, wheeled mobile robot must be subjected to nonholonomic constraints, which make it challenging to construct a controller with desired performance. As a result, tracking control of nonholonomic wheeled mobile robot (NWMR) has always been a research focus for past decades.

Due to the existence of nonholonomic constraints of WMR, the approaches to tracking control including both kinematics control and dynamics control, which has been a basic pipeline of tracking control for NWMR. As for kinematics control, it is used to tracking desired pose with WMR's speed commands, thus a time-varying controller based on Lyapunov theory was proposed in [1]. Kinematics mode with chained-form [2], [3] was used to transform this complex system to a convenient one. Besides, the model described in polar coordinates [4] was also reported to design a robust system. In [5], mode prediction control algorithm combined with neural-dynamics optimization was proposed by using the derived tracking-error kinematics, which effectively achieved tracking control under the velocity constraints and velocity increment constraints. More recently, in [6], nonlinear controllers using synthetic-analytic behavior-based control framework was presented to track with velocity constraints. A PID-based kinematic controller is proposed as a non-model based controller to navigate the tractor-trailer wheeled robot follows desired trajectories in [7].

As for a real WMR, it is obvious that separate kinematics controller can not perform trajectory tracking well. So various advanced dynamics controller is adopted to track the desired

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

velocity, which is the output of kinematics controller exactly. These researches have been mainly concentrated on overcoming system uncertainties and external disturbances. Instead of classical torque-based control, a robust control approach [8] was developed based on the voltage control strategy. In [9], a robust adaptive controller is proposed with the utilization of adaptive control, backstepping and fuzzy logic techniques. Considering the robust performance of sliding mode control, in [10], a controller with finite-time convergence of the tracking errors was provided, the disturbance observer and adaptive compensator were used to enhanced the robustness of system. Similarly, an integral terminal sliding mode controller [11] was adopted in the presence of parameter uncertainties and external disturbances, and an adaptive fuzzy observer was introduced to compensate the non-measurement of velocity. In [12] a fast terminal sliding mode control scheme was proposed under known or unknown upper bound of the system uncertainty and external disturbances.

From the perspective of optimization, a controller [13] based on model prediction control was proposed to prevent sideslip and improve the performance of path tracking control. A nonlinear model predictive controller [14] was introduced by using a set of modifications to track a given trajectory. With the consideration of the uncertainties to be time varying and dynamic, a robust control strategy [15] was proposed with time delay control. In [16], the optimization-based nonlinear control laws were analytically developed using the prediction of WMR responses, the tracking precision is more increased with the integral feedback technique appending. Because neural networks(NNs) can approximate nonlinear functions well, the NNs-based method [17] was provided to approximate the unknown modeling item, the skidding and the slipping item, though it is not common in the low-level driver. Therefore, it could be concluded that kinematics control and dynamics control are two different ways to address the problem of tracking control, in which a variety of nonlinear control approaches could be employed. And both methods highly dependent on a system model, an acceptable algorithm with more accurate model may lead to more precise control accuracy. However, it is hard to describe with nonlinear formulations exactly, especially the mode uncertainties and disturbances.

Except the model-based control method mentioned above, the learning-based (reinforcement learning) methods have been become a new research focus [18], because there is no need to consider a system model. In [19], with the candidate parameters of the PD controller defined as the action space, a hierarchical reinforcement learning approach for optimal path tracking of WMR was proposed, but the state space and action space was decomposed into several subspaces, which is not amenable to the continuous control problem. Thus, the RL method with continuous space has been studied, an actor-critic goal-oriented deep RL architecture [20] was developed to achieve adaptive low-level control strategy in continuous space. In [21], an RL algorithm is designed to generate an optimal control signal for uncertain nonlinear MIMO systems. In [22], A RL-based adaptive tracking control algorithm is proposed for a time-delayed WMR system with slipping and skidding. In [23] a layered depth reinforcement learning algorithm for robot composite tasks is proposed, which is superior to common deep reinforcement learning algorithm among discrete state space. A solution for the path following problem of a quadrotor vehicle based on deep reinforcement learning theory is proposed in three different conditions [24].

Although the excellent performance with RL algorithms, it has been suffered from the disadvantages of time-consuming training and ineffective sampling with interaction between agent and environment [25]. Thus, In [26], a model-based reinforcement learning algorithm with excellent sample complexity was achieved by combining neural network dynamics models with model predictive control (MPC), which produce stable and plausible gaits that accomplish various complex locomotion tasks. And, a kernel-based dynamic model for reinforcement learning was proposed to fulfill the robotic tracking tasks [27], and the optimal control policy is searched by the model-based RL method. In [28] multi pseudo Q-learning-based deterministic policy gradient algorithm was proposed to achieve high-level tracking control accuracy of AUVs, which validated that increasing the number of the actors and critics could further improve the performance. Recently, a data-based approach for analyzing the stability of discrete-time nonlinear stochastic systems modeled by Markov decision process, by using the classic Lyapunov's method in control theory [29]. Due to the limited exploration ability caused deterministic policy, high-speed autonomous drifting is addressed, using a closed-loop controller based on the deep RL algorithm soft actor critic (SAC) to control the steering angle and throttle of simulated vehicles in [30]. We should notice a fact that deep reinforcement learning algorithms always require time-consuming training episodes. This may be acceptable to a certain extent for simulated robots, but it is not feasible for a actual environment. So the effort should be concentrated on improving the efficiency of deep reinforcement learning algorithms.

B. MOTIVATION OF OUR APPROACH

In general, the model-based control methods have always been preferred to develop a controller, and the performance will depend largely on the accuracy of the model. However, model uncertainty and external disturbances are objective and have to be addressed. Thus, a number of robust strategies should be adopted to obtain a controller with more precise control accuracy. Furthermore, once the control algorithm is determined, the accuracy of the controller remains unchanged. It may lose the possibility of improving itself by learning, just like what our humans doing. While the RL-based method do not need a system mode at all, and the human-level performance could be obtained with a reasonable end-to-end training process. Naturally, the synthesis of mode-based control and learning-based control could be a pretty alternative for autonomous WMR.

Considering the great tracking performance of dynamics controller at present, we prefer to control the velocity based on kinematics mode. And existing kinematics controllers are used for solving a complex nonlinear control problem. Thus, it is suboptimal and difficult to improved with model-based methods. So, the learning method can be used to optimize the existing kinematics controller to obtain a better tracking performance.

Thus, in our effort of tracking control for NWMR, the kinematics control is chose as mode-based method, just like ''given talent'' of human. And the actor-critic based reinforcement learning method is adopted to learn the tracking experience during the whole tracking process, just like "acquired knowledge". The main contribution of our proposed method are as follows.

- A hybrid control strategy combining mode-based method and deep reinforcement learning method for tracking control is proposed, which shows better performance both in accuracy and efficiency.
- The state is defined including current tracking errors, given control inputs and one-step errors, which is one of the keys to efficient convergence of tracking control.

The reminder of this paper is organized as follows. In Section II, the kinematics mode of NWMR with constraints and the given control law based Lyapunvo theory are described. In Section III, we elaborate our hybrid control strategy combined mode-based control and actor-critic based DRL in detail. In Section IV, we present the simulation results of our method under periodic and random disturbances. Finally, we conclude the full text in Section V.

II. GIVEN CONTROL LAW BASED ON KINEMATICS MODEL

A. KINEMATICS MODEL OF NWMR WITH VELOCITY AND ACCELERATION CONSTRAINTS

As shown in Fig. 1, a NWMR with two drive wheels whose axis is connected through the geometric center of the robot body. The left and right drive wheels are respectively driven by two hub motors to realize the forward, backward and turning of the robot. Point C is midpoint between two hub motors' axial connections, and its coordinate in the global coordinate system is (x,y), θ is orientation of mobile robot. v is linear velocity of robot, and ω is angular velocity robot. What's more, r represents radius of outer ring of drive wheel, L denotes vertical distance between the center of the drive wheel and the midpoint C of the shaft center line. v_l, v_r are the line speeds of the right and left drive wheels, respectively.

The kinematics of nonholonomic wheeled mobile robot can be denoted as:

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{1}$$

Although our method is based on a kinematics model, dynamic constraints must still be considered, because the control inputs for NWMR need a certain response time, and

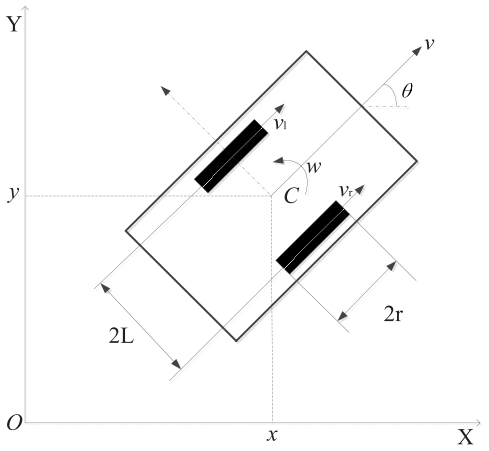

FIGURE 1. Nonholonomic wheeled mobile robot.

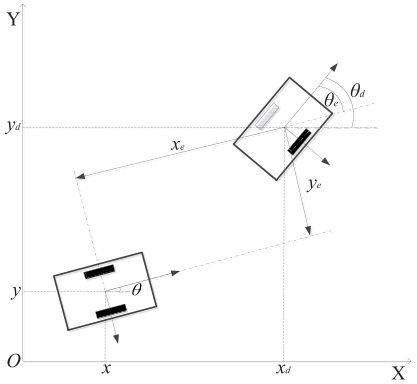

FIGURE 2. Tracking control of NWMR.

they cannot achieve a sudden change. The linear velocity v, angular velocity ω, linear acceleration a, and angular acceleration α for NWMR are all bounded,

$$\begin{aligned} |v| &\le v_{max} \\ |\omega| &\le \omega_{max} \\ |a| &\le a_{max} \\ |\alpha| &\le \alpha_{max} \end{aligned} \tag{2}$$

B. GIVEN CONTROL LAW

As shown in Fig. 2, for any given mobile robot's desired pose $\mathbf{q}_d^{\mathrm{T}} = [x_d, y_d, \theta_d]^{\mathrm{T}}$, the current pose error of the robot in the global coordinate system is:

$$\begin{aligned} \tilde{\mathbf{q}}_e &= R(\tilde{\theta})(\mathbf{q}_d - \tilde{\mathbf{q}}) \\ &= \begin{bmatrix} \tilde{x}_e \\ \tilde{y}_e \\ \tilde{\theta}_e \end{bmatrix} = \begin{bmatrix} \cos\tilde{\theta} & \sin\tilde{\theta} & 0 \\ -\sin\tilde{\theta} & \cos\tilde{\theta} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - \tilde{x} \\ y_d - \tilde{y} \\ \theta_d - \tilde{\theta} \end{bmatrix} \end{aligned} \tag{3}$$

where, the current pose of robot consists of two parts, one is the ideal true value of pose, another is bounded uncertainty caused by external disturbance or noise,

$$\tilde{\mathbf{q}} = \mathbf{q} + \mathbf{n} \tag{4}$$

The error state dynamics can be written as follow:

$$\begin{aligned}\dot{\tilde{x}}_e &= \omega\tilde{y}_e - v + v_d\cos\tilde{\theta}_e\\ \dot{\tilde{y}}_e &= -\omega\tilde{x}_e + v_d\sin\tilde{\theta}_e\\ \dot{\tilde{\theta}}_e &= \omega_d - \omega\end{aligned} \tag{5}$$

Tracking control is to seek $\mathbf{u} = (v, \omega)^{\mathrm{T}}$, which makes current pose error converge to zero with $t \to \infty$, similar to [6], our given kinematics control law is chose as:

$$\mathbf{u}_g = \begin{bmatrix} v_g \\ \omega_g \end{bmatrix} = \begin{bmatrix} k_1\tilde{x}_e + v_d\cos\tilde{\theta}_e \\ 2v_d\tilde{y}_e\cos\dfrac{\tilde{\theta}_e}{2} + \omega_d + k_2\sin\dfrac{\tilde{\theta}_e}{2} \end{bmatrix} \tag{6}$$

where k_1 and k_2 are positive constant.

The stability of closed-loop system could be proved according Lyapunov theory, see Appendix.

With the control input in (6), NWMR will move to a new pose $\tilde{\mathbf{q}}'$ (contain bounded uncertainty, same as (4)) in global coordinate, so the one-step error will be described as,

$$\begin{aligned}\tilde{\mathbf{q}}'_e &= \mathbf{q}_d - \tilde{\mathbf{q}}'\\ &= \begin{bmatrix} \tilde{x}'_e \\ \tilde{y}'_e \\ \tilde{\theta}'_e \end{bmatrix} = \begin{bmatrix} \cos\tilde{\theta}' & \sin\tilde{\theta}' & 0 \\ -\sin\tilde{\theta}' & \cos\tilde{\theta}' & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - \tilde{x}' \\ y_d - \tilde{y}' \\ \theta_d - \tilde{\theta}' \end{bmatrix}\end{aligned} \tag{7}$$

In other words, the pose error of NWMR has made a transition from the previous error $\tilde{\mathbf{q}}_e$ in (3) to the latest error $\tilde{\mathbf{q}}'_e$ in (7).

Theoretically, the tracking error could gradually converge to zero, when the time tends to infinity. It can be denoted as following:

$$\lim_{t\to\infty} \left|\tilde{\mathbf{q}}'_e\right| \approx 0 \tag{8}$$

However, according to error dynamics equation in Appendix, the nolinear kinematics controller could be suboptimal in finite time, and external disturbance or noise during the tracking control could also lead to a result that the tracking error converges to a certain positive value c within a finite time t_0 instead.

$$\lim_{t\to t_0} \left|\tilde{\mathbf{q}}'_e\right| >= c \tag{9}$$

Besides, once the given control law (6) is determined, the converge performance of closed-loop system is determined. It can not be able to adjust itself to obtain more precise control performance, so additional strategy is needed to improve it.

III. HYBRID CONTROL STRATEGY INCORPORATING DEEP REINFORCEMENT LEARNING APPROACH

In this section, we consider a method of deep reinforcement learning, to help NWMR learn the acquired control law from tracking error caused by given control law in secion II-B.

A. FINITE MDP

To convert our acquired control problem of wheeled mobile robot to a general RL problem, we model it as a finite Markov decision process [31], which is the fundamental property for RL theory.

Firstly, we define the state $\mathbf{s}_k$ at kth time step as,

$$\begin{aligned}\mathbf{s}_k &= \left[\tilde{\mathbf{q}}_e^{\mathrm{T}}(k), \mathbf{u}_g^{\mathrm{T}}(k), \tilde{\mathbf{q}}_e^{'\mathrm{T}}(k)\right]\\ &= \left[\tilde{x}_e(k), \tilde{y}_e(k), \tilde{\theta}_e(k), v_g(k), \omega_g(k), \tilde{x}'_e(k), \tilde{y}'_e(k), \tilde{\theta}'_e(k)\right]\end{aligned} \tag{10}$$

And, the ation at kth time step, called acquired control law, is obtained with a deterministic policy μ,

$$\mathbf{a}_k = \mathbf{u}_a(k) = [v_a(k), \omega_a(k)]^{\mathrm{T}} = \mu(\mathbf{s}_k) \tag{11}$$

Then, the hybrid tracking control input at current time is,

$$\mathbf{u}(k) = \mathbf{u}_g(k) + \mathbf{u}_a(k) \tag{12}$$

The immediate reward r_k at kth time step is,

$$r_k = -(\left|\tilde{x}'_e(k)\right| + \left|\tilde{y}'_e(k)\right| + \left|\tilde{\theta}_e(k)\right|) \tag{13}$$

The cumulative reward of whole learning proscess is calculated with a discount constant $0 < \gamma <= 1$,

$$G_k = \sum_{i=1}^{N} \gamma^{i-1} r_{k+i} \tag{14}$$

It should be mentioned that the state of our RL problem includes not only the tracking error $\tilde{\mathbf{q}}_e(k), \tilde{\mathbf{q}}'_e(k)$, but also the given control vector $\mathbf{u}_g$, which is one of key part of our method, and without it could lead our strategy to fail.

Therefore, the error vectors $s(k)$ constitute a finite state space $\mathcal{S}$, the adjustment control vectors $\mathbf{u}_a(k)$ constitute a finite action space $\mathcal{A}$, plus a reward function r_k, a markovian system for tracking control is completed. and $[\mathbf{s}_k, \mathbf{a}_k, r_k, \mathbf{s}_{k+1}]$ is called a transition. In this problem, we are seeking a optimal policy μ^* to maximize cumulative reward with a DRL method.

B. ACQUIRED CONTROL LEARNING WITH ACTORS-CRITICS ARCHITECTURE

Due to the experience model is difficult to accurately represent with mathematical expressions, the method in this paper is constructed with deep deterministic policy gradient algorithm [32]. The deterministic policy μ is represented by actor network $\pi(s; \lambda)$ with parameters λ, neural networks with parameters β is used to represent critic network $Q(s, a; \beta)$. And another two deep neural networks with parameters $\hat{\lambda}$ and $\hat{\beta}$ are used to represent target actor network $\hat{\pi}(s; \hat{\lambda})$ and target critic network $\hat{Q}(s, a; \hat{\beta})$. The first two networks form a real-time network, the weights are updated in real time. As well, the latter two form a target network which is updated with a soft strategy. During the training process, the parameters of all networks will be updated with the continuously transitions

in replay buffer. Our optimal acquired control law will be obtained when the training is done.

In order to update actor network, we first calculate the gradient of critic network, TD error $L(\beta)$ of $Q(s, a; \beta)$ is defined as mean square of target Q value and current Q value:

$$L(\beta) = \frac{1}{N}\sum_{k=1}^{N}(T_k - Q(s_k, a_k|\beta))^2 \tag{15}$$

where, N denotes the number of transitions in a small batch; T_i, as the label, is output of target critic network $\hat{Q}(s, a; \hat{\beta})$:

$$T_k = r(s_k, a_k) + \gamma \hat{Q}(s_{k+1}, \hat{\pi}(s_{k+1}|\hat{\lambda})|\hat{\beta}) \tag{16}$$

Thus, the gradient of TD error is:

$$\nabla_\beta L(\beta) = -\frac{2}{N}\sum_{k=1}^{N}(T_k - Q(s_k, a_k|\beta))\frac{\partial Q(s, a; \beta)}{\partial \beta} \tag{17}$$

The critic network is updated with,

$$\beta \leftarrow \beta + \mathrm{L_c} \cdot \nabla_\beta L(\beta) \tag{18}$$

where $\mathrm{L_c}$ is the learning rate.

For actor network $\pi(s; \lambda)$, it is the map from input observation $\mathbf{s}_k \in \mathcal{S}$ to output action $\mathbf{a}_k \in \mathcal{A}$. Assuming the transitions in replay buffer is distributed according to a strategy $\phi(a|s)$ and probability density function is ρ^ϕ, the objective function of actor network can be defined as:

$$\begin{aligned} F_\phi(\pi_\lambda) &= \int_S \rho^\phi(s) Q^\pi(s, \pi_\lambda(s)) ds \\ &= \mathbb{E}_{S\sim\rho^\phi}[Q(s, \pi_\lambda(s))] \end{aligned} \tag{19}$$

According to [33], off-policy deterministic policy gradient is:

$$\begin{aligned} \nabla F_\phi(\pi_\lambda) &\approx \int_S \rho^\phi(s) \nabla_\lambda \pi_\lambda(a|s) Q^\pi(s, a) ds \\ &= \mathbb{E}_{S\sim\rho^\phi}[\nabla_\lambda \pi_\lambda(s) \nabla_a Q^\pi(s, a)|_{a=\pi_\lambda(s)}] \end{aligned} \tag{20}$$

So when we got the mini-batch data randomly from the replay memory buffer, the policy gradient is:

$$\begin{aligned} \nabla F_\phi(\pi_\lambda) = \frac{1}{N}\sum_{k=1}^{N}(&\nabla_\lambda \pi_\lambda(s|\lambda)|_{s=s_k} \\ &\cdot \nabla_a Q^\pi(s, a|\beta)|_{s=s_k, a=\pi_\lambda(s)}) \end{aligned} \tag{21}$$

Thus, the actor network is updated with,

$$\lambda \leftarrow \lambda + \mathrm{L_a} \cdot \nabla F_\phi(\pi_\lambda) \tag{22}$$

where $\mathrm{L_a}$ is learning rate.

For the stability of training, parameter vectors of target critic network and target actor network are updated in this way:

$$\hat{\lambda} = \varepsilon\lambda + (1 - \varepsilon)\hat{\lambda} \tag{23}$$

$$\hat{\beta} = \varepsilon\beta + (1 - \varepsilon)\hat{\beta} \tag{24}$$

where $\varepsilon \ll 1$.

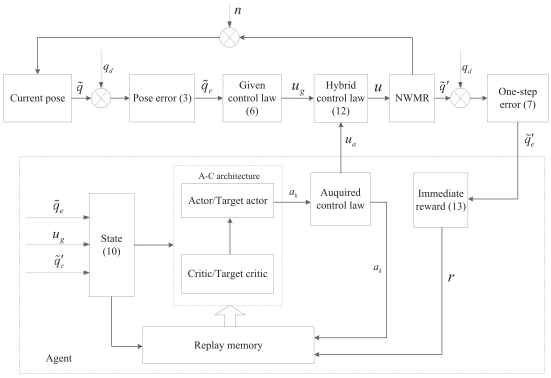

FIGURE 3. Control block diagram of our hybrid control strategy for NWMR.

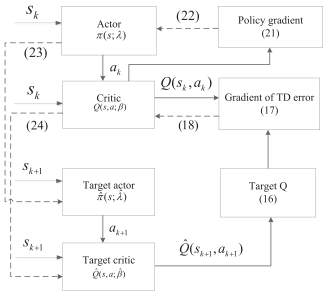

FIGURE 4. Actor-Critic architechture in our hybrid control strategy.

Finally, when the training is over, the optimal acquired control law u_a^* will be obtained with the optimal actor network π^* with optimal parameter vector λ^*,

$$\mathbf{u}_a^* = \pi^*(\mathbf{s}; \lambda^*) \tag{25}$$

We use fully connected model to build target actor network, so it can be expressed with the forward model as follows:

$$\begin{aligned} \mathbf{l}_1 &= f_1(\mathbf{W}_1^*\mathbf{s} + \mathbf{b}_1^*) \\ \mathbf{l}_2 &= f_2(\mathbf{W}_2^*\mathbf{l}_1 + \mathbf{b}_2^*) \\ &\vdots \\ \mathbf{l}_{n-1} &= f_{n-1}(\mathbf{W}_{n-1}^*\mathbf{l}_{n-2} + \mathbf{b}_{n-1}^*) \\ \mathbf{u}_a^* &= f_n(\mathbf{W}_n^*\mathbf{l}_{n-1} + \mathbf{b}_n^*) \end{aligned} \tag{26}$$

where $\mathbf{l_i}$ is output of ith layer, $\mathbf{W}_i^*, \mathbf{b}_i^* \in \lambda^*$, are network parameters, f_i is activation function of ith layer, n is number of layers.

So far, the hybrid tracking control law for NWMR is obtained after training, which combined given control from kinematics control method and acquired control from DRL method. The control block diagram is showed in Fig. 3 and Fig. 4 The pseudocode of our tracking control method is shown in Algorithm 1:

IV. SIMULATION RESULTS

In this section, simulations are developed to demonstrate that above proposed method could achieve tracking control for NWMR effectively. We first try to track a circle, and it can

Algorithm 1 Hybird Strategy of Tracking Control for NWMR

Require: $\mathbf{q}_r, \mathbf{q}_0, k_1, k_2, v_d, \omega_d, N, \gamma, \varepsilon$

Initialize/load actor network and critic network, λ, β
Initialize target network, $\hat{\lambda} = \lambda, \hat{\beta} = \beta$
Initialize replay buffer
for episode=1 to Max-ep **do**
  get initial pose observation of NWMR, compute intital state $\mathbf{q}_e^0, u_g^0, \mathbf{q}_e^{0'}$
  Initialize cumulative error to zero
  for step=1 to Max-step **do**
    compute given control $\mathbf{u}_g^k$ according to $\mathbf{q}_e^k$, (6)
    compute acquired control $\mathbf{u}_a^k$ according to $\left[\mathbf{q}_e^k, \mathbf{u}_g^k, \mathbf{q}_e^{'k}\right]$, (Sec.III-B)
    execute $\mathbf{u}^k = \mathbf{u}_g^k + \mathbf{u}_a^k$
    store *kth* transition to replay buffer
    if number of transitions > Memory **then**
      extract randomly a batch of transitions from R
      update actor network and critic network, (18), (22)
      update target network, (23), (24)
    end if
  end for
end for

TABLE 1. Parameters of tracking circle.

ξ_1	ξ_2	ξ_3	k_1	k_1
1.0	1.0	1.0	0.3	0.1

be defined as:

$$x_d = 2\cos\theta$$
$$y_d = 2\sin\theta$$

The cumulative error including pose error and control error is introduced to be a criteria of tracking performance, and a bigger value (because it is negative) means the better tracking accuracy:

$$E = -\sum_{k=1}^{N} \xi_1(|x_e(k)| + |y_e(k)|) + \xi_2 |\theta_e(k)| + \xi_3(|v_d - v(k)| + |\omega_d - \omega(k)|)$$

where ξ_1, ξ_2, ξ_3 are weight coefficients corresponding to pose error, orientation error and control vector error.

The bounded values in (2), $v_{max} = 2m/s$, $\omega_{max} = 1rad/s$, $a_{max} = 1m/s^2$, $\alpha_{max} = 1.5rad/s^2$. The parameters of tracking circle are given in Tab. 1. k_1 and k_2 are fine tuned with above criteria E, $v_d = 1m/s$, $\omega_d = 0.5rad/s$.

The uncertainty in (4) is chosen as periodic disturbance,

$$\mathbf{n} = \begin{bmatrix} n_x \\ n_y \\ n_\theta \end{bmatrix} = \begin{bmatrix} 0.002\sin(\pi t) \\ 0.002\cos(\pi t) \\ 0.005\sin(\pi t) \end{bmatrix}$$

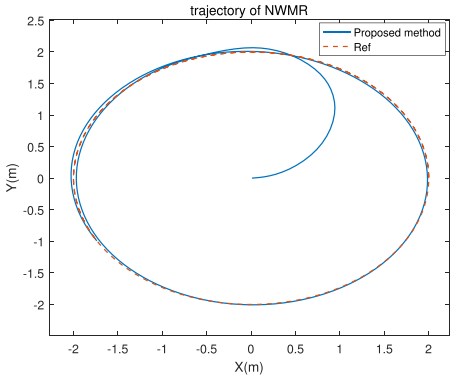

FIGURE 5. Trajectory of tracking circle with our method.

TABLE 2. Hyperparameters of network.

layer	actor/target actor	critic/target critic	activation function
input layer	8	10	ReLU
1st hidden layer	60	60	ReLU
2nd hidden layer	60	60	ReLU
3rd hidden layer	60	10	ReLU
output layer	2	1	Tanh

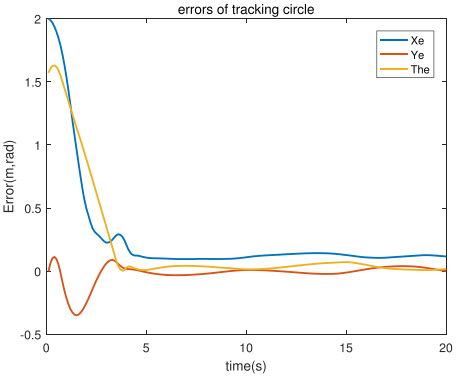

FIGURE 6. Errors of tracking circle with our method.

The network architecture in section III-B is built with the aid of Tensorflow,[1] actor/target actor network havs a similar deep fully connected neural network with critic/target critic network, the hyperparameters are shown in Tab.2. Besides, the maximum size of replay buffer is 5000, the size of batch is 32, learning rates of actor network and critic network are 0.001, 0.002. In our training, it is a total of 400 episodes and 200 steps in each episode, and sampling time is $0.1s$, initial state of NWMR is $(0, 0, 0)^{\mathrm{T}}$.

The results of our proposed method are showed in Figs. 5–9. The trajectory of NWMR can be seen in Fig. 5, desired trajectory is shown with a dotted line, and another tracjectory is solid line. From Fig. 6, it can be observed that the tracking errors all converge to near zero. The given control signals, acquired control signals and final hybrid control inputs of (12) are shown in Figs. 7–9, respectively. The

[1] An open source machine learning library

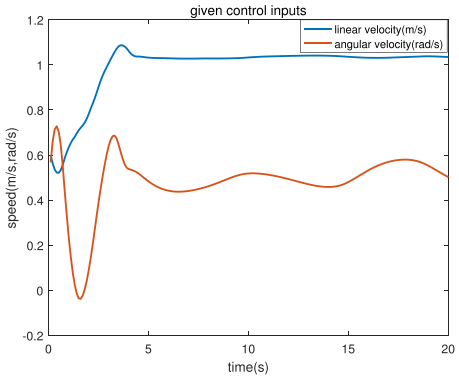

FIGURE 7. Given control signals of tracking circle with our method.

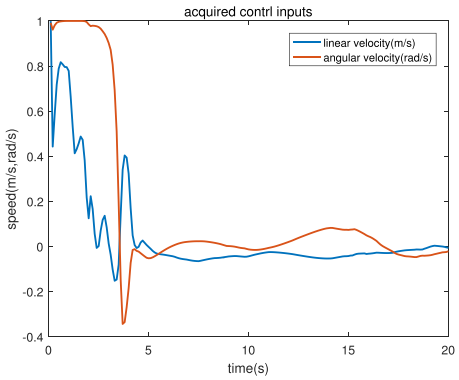

FIGURE 8. Acquired control signals of tracking circle with our method.

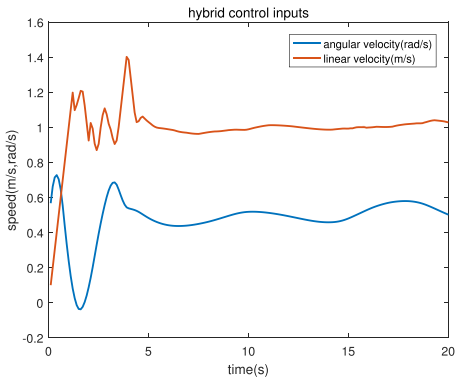

FIGURE 9. Hybrid control inputs of tracking circle with our method.

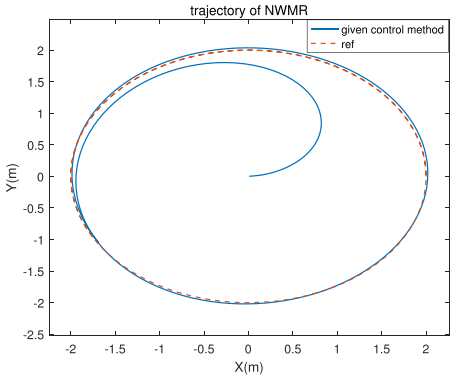

FIGURE 10. Trajectory of NWMR with classical control method.

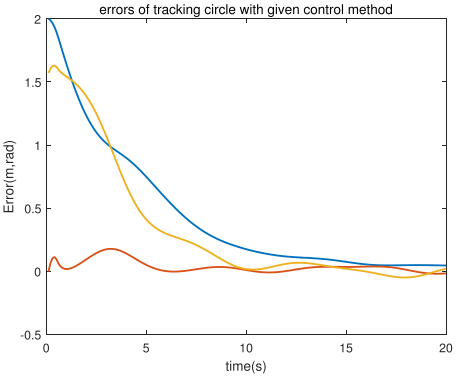

FIGURE 11. Errors of tracking circle with classical control method.

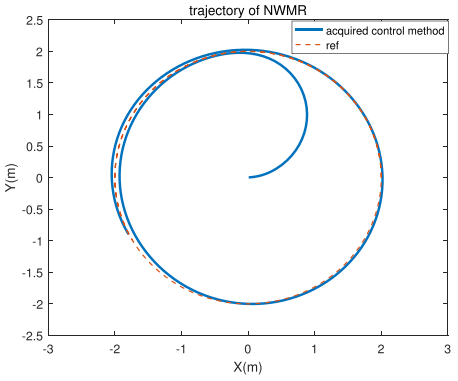

FIGURE 12. Trajectory of NWMR with leaning control method.

acquired control inputs works in the whole process, which prove the effectiveness of our method.

To make a comparison, we also test the performance with classical method, that is, only the given control approach (6) works. The results are depicted in Figs. 10–11. Comparing Fig. 10 with Fig. 5, our method obviously performs better, the comparison of Fig. 11 with Fig. 6 also proves this point. Actually, the cumulative error of classical method in Fig. 11 is -202.0255, while the one of our method in Fig. 5 is -110.4874. So, the addition of u_a in our method could improve tracking performance indeed.

Besides, we also try to use learning method, that is, only acquired control method (11) works, the results are depicted in Fig. 12–15. Comparing Fig. 12 with Fig. 5, the performance of tracking circle is similar to each other, the cumulative error of learning method in Fig. 12 is -110.1912. But the comparison of training process in Fig. 14 and Fig. 15 shows that our proposed method converge to stable within 300 episodes, and the fluctuation of the reward (Y axis) in former is more stable than the latter, it proves the superiority in training process of our method.

To further demonstrate the effectiveness of our method, we conduct another simulation to track the spiral trajectory,

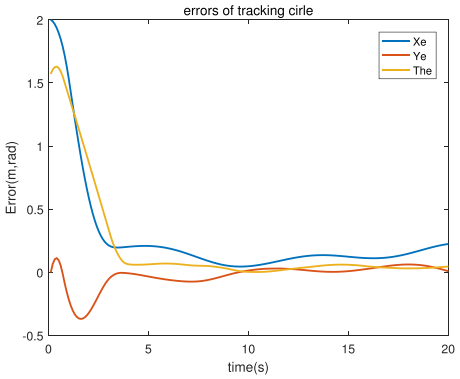

FIGURE 13. Errors of tracking circle with learning control method.

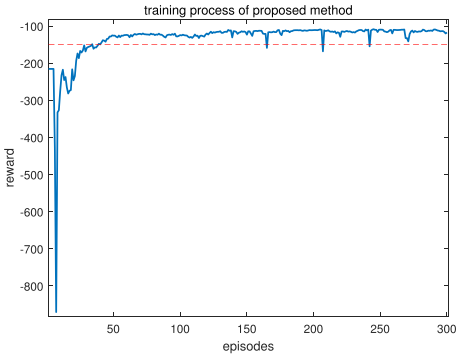

FIGURE 14. Training process of tracking cirle with our method, take Y=-150 (red dotted line) as reference.

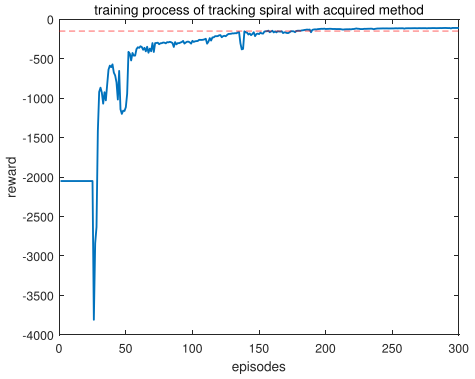

FIGURE 15. Training process of tracking cirle only with learning method, take Y=-150 (red dotted line) as reference.

defined as following,

$$x_d = 0.04t\cos(0.5t)$$

$$y_d = 0.04t\sin(0.5t)$$

The uncertainty in (4) is chosen as random disturbance,

$$\mathbf{n} = 0.002\sigma$$

where, $\sigma \sim \mathcal{N}(0, 1)$.

TABLE 3. Parameters of tracking spiral.

ξ_1	ξ_2	ξ_3	k_1	k_1
10.0	1.0	0.0	5.0	3.0

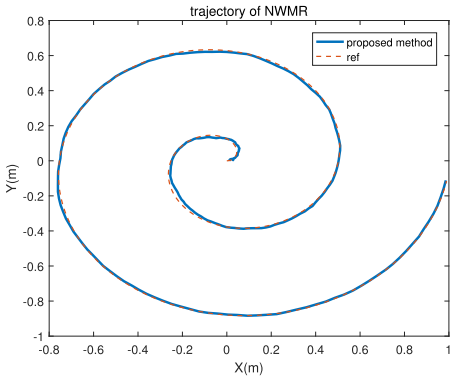

FIGURE 16. Trajectory of tracking spiral with our method.

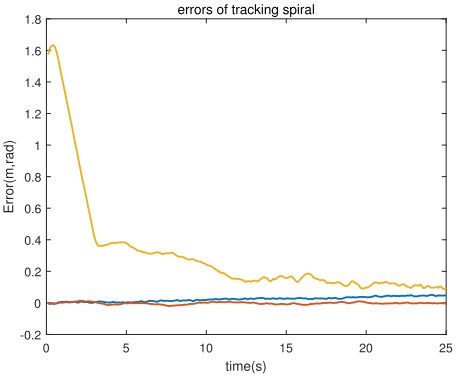

FIGURE 17. Errors of tracking spiral with our method.

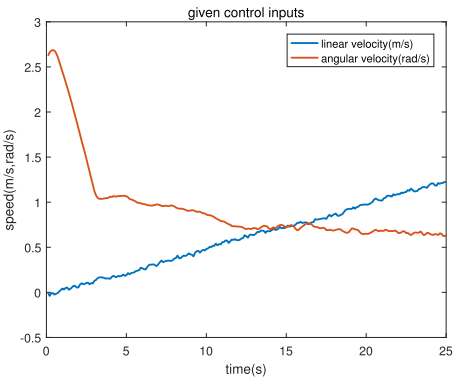

FIGURE 18. Given control signals of tracking spiral with our method.

And the parameters are given Tab. 3. The size of replay buffer is 5000, the size of batch is 32, learning rates of actor network and critic network are 0.0002, 0.001. The Maximum episode is 800, the maximum step in each spisode is 250, while other parameters remain unchanged.

The results are depicted in Figs. 16–20. The trajectory of NWMR is shown in Fig. 16, tracking errors are depicted

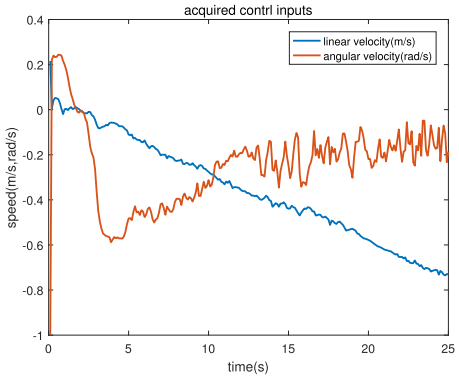

FIGURE 19. Acquired control signals of tracking spiral with our method.

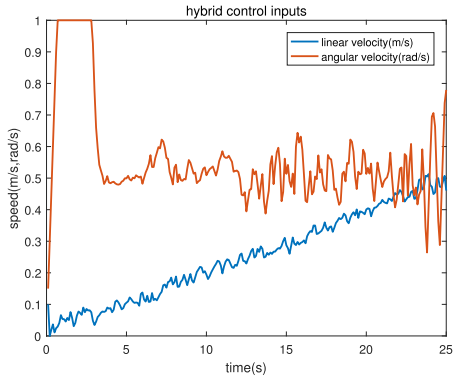

FIGURE 20. Hybrid control inputs of tracking spiral with our method.

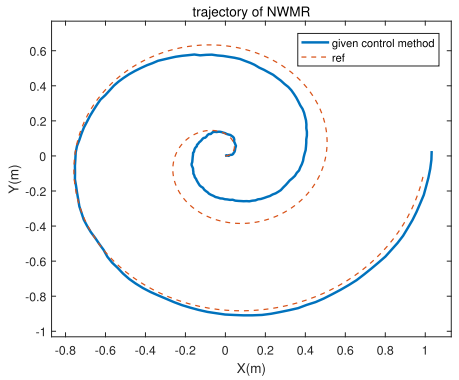

FIGURE 21. Trajectory of tracking spiral with classical method.

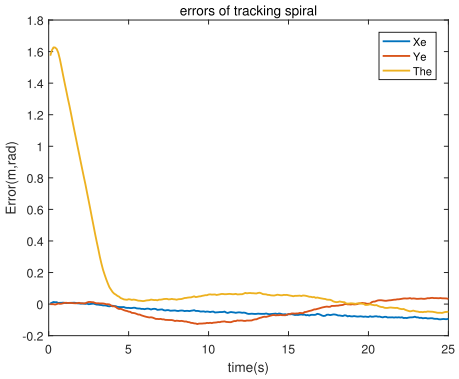

FIGURE 22. Errors of tracking spiral with classical method.

in Fig. 17, given control signals, acquired control signals and hybird control inputs can be seen in Fig. 18–20. From Fig. 18,

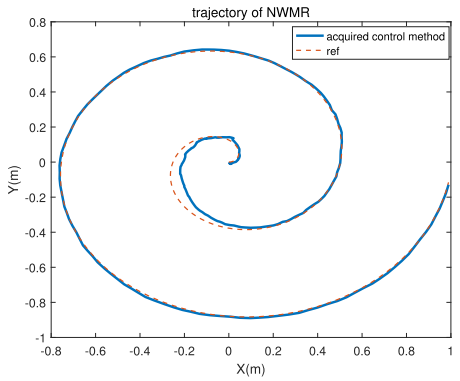

FIGURE 23. Trajectory of tracking spiral with learning method.

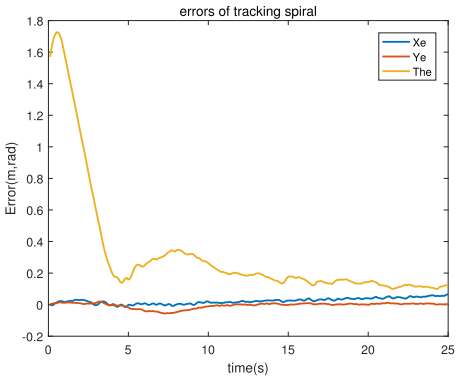

FIGURE 24. Errors of tracking spiral with learning method.

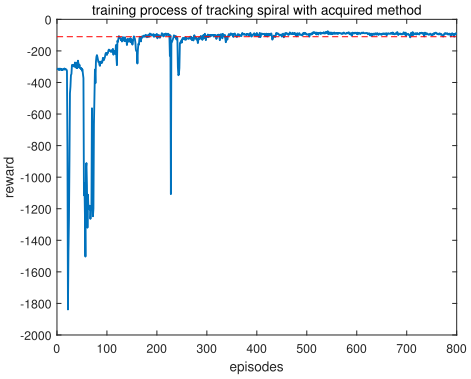

FIGURE 25. Training process of tracking spiral with our method, take Y=−110 (red dotted line) as reference.

the angular velocity exceeds the upper bound already, but the hybird control inputs in Fig. 20 is bounded.

The results with only classical control approach works are also illustrated as a comparison in Fig. 21 and Fig. 22. The cumulative error of classical method in Fig. 21 is -358.0541, while the one of our method in Fig. 16 is -93.7636. So, it also proved the effectiveness of our proposed method. The results with only learning method is Figs. 23–26. In Fig. 23, the cumulative error is -114.4648, comparing to Fig. 16, the tracking performance of our method is still better. According to training process in Fig. 25 and Fig. 26, our proposed method is obviously more stable, too.

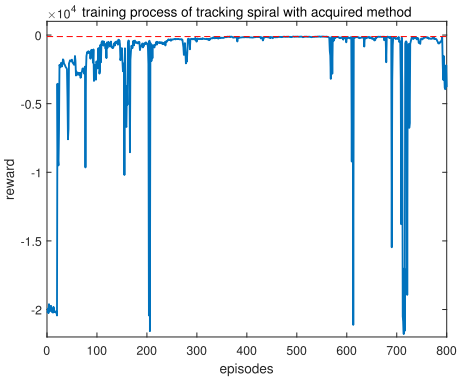

FIGURE 26. Training process of tracking spiral only with leaning method, take Y=−110 (red dotted line) as reference.

So, for tracking circler, our proposed method has similar tracking performance to learning method, but, the convergence performance of ours is better; for tracking spiral, our proposed method has advantage both in tracking performance and converge performance evidently.

V. CONCLUSION

In this research, the tracking control of NWMR with constraints and uncertainty has been addressed by our proposed hybrid control strategy, which is a combination of mode-based control method and learning based method. The kinematics control is severed as a given control (like "the talent"), the actor-critic based DRL method is used to learn a acquired control law to compensate the existing errors (like"the experience"). The results have demonstrated the effectiveness of our proposed method, and the comparisons show that our method has the advantage of less cumulative error, meanwhile, our method is more stable and efficient than learning based method.

The strategy provided in our effort could improve tracking and convergence performance, which is the vital function for a autonomous mobile robot. Although our method have been tested with tracking control of NWMR, it could also be applied to other complicated control problems.

APPENDIX

Substituting (6) to (5), the error dynamics can be rewritten as:

$$\dot{x}_e = -k_1 x_e + 2v_d y_e^2 \cos\frac{\theta_e}{2} + k_2 y_e \sin\frac{\theta_e}{2} + \omega_d y_e$$
$$\dot{y}_e = -2v_d x_e y_e \cos\frac{\theta_e}{2} - \omega_d x_e - k_2 x_e \sin\frac{\theta_e}{2} + v_d \sin\theta_e$$
$$\dot{\theta}_e = -2v_d y_e \cos\frac{\theta_e}{2} - k_2 \sin\frac{\theta_e}{2}$$

Defining the Lyapunov function,

$$L = \frac{1}{2}x_e^2 + \frac{1}{2}y_e^2 - 2\cos\frac{\theta_e}{2}$$

Deriving Lyapunov function along time:

$$\dot{L} = x_e\dot{x}_e + y_e\dot{y}_e + \dot{\theta}_e \sin\frac{\theta_e}{2}$$
$$= x_e(-k_1 x_e + 2v_d y_e^2 \cos\frac{\theta_e}{2} + k_2 y_e \sin\frac{\theta_e}{2} + \omega_d y_e)$$
$$+ y_e(-2v_d x_e y_e \cos\frac{\theta_e}{2} - \omega_d x_e - k_2 x_e \sin\frac{\theta_e}{2} + v_d \sin\theta_e)$$
$$+ (-v_d y_e \sin\theta_e - k_2 \sin^2\frac{\theta_e}{2})$$
$$= -k_1 x_e^2 - k_2 \sin^2\frac{\theta_e}{2} <= 0$$

According Lyapunov theory, the error dynamics will asymptotically converge to zero.

REFERENCES

[1] Y. Wang, Z. Miao, H. Zhong, and Q. Pan, "Simultaneous stabilization and tracking of nonholonomic mobile robots: A Lyapunov-based approach," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 4, pp. 1440–1450, Jul. 2015, doi: 10.1109/TCST.2014.2375812.

[2] S. Mobayen, "Finite-time tracking control of chained-form nonholonomic systems with external disturbances based on recursive terminal sliding mode method," *Nonlinear Dyn.*, vol. 80, nos. 1–2, pp. 669–683, Apr. 2015, doi: 10.1007/s11071-015-1897-4.

[3] H. Chen, B. Zhang, T. Zhao, T. Wang, and K. Li, "Finite-time tracking control for extended nonholonomic chained-form systems with parametric uncertainty and external disturbance," *J. Vibrat. Control*, vol. 24, no. 1, pp. 100–109, Jan. 2018, doi: 10.1177/1077546316633568.

[4] B. Seok Park, S. Jin Yoo, J. Bae Park, and Y. Ho Choi, "Adaptive neural sliding mode control of nonholonomic wheeled mobile robots with model uncertainty," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 1, pp. 207–214, Jan. 2009, doi: 10.1109/Tcst.2008.922584.

[5] Z. Li, J. Deng, R. Lu, Y. Xu, J. Bai, and C.-Y. Su, "Trajectory-tracking control of mobile robot systems incorporating neural-dynamic optimized model predictive approach," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 46, no. 6, pp. 740–749, Jun. 2016, doi: 10.1109/Tsmc.2015.2465352.

[6] M. Meza-Sánchez, E. Clemente, M. C. Rodríguez-Liñán, and G. Olague, "Synthetic-analytic behavior-based control framework: Constraining velocity in tracking for nonholonomic wheeled mobile robots," *Inf. Sci.*, vol. 501, pp. 436–459, Oct. 2019, doi: 10.1016/j.ins.2019.06.025.

[7] A. K. Khalaji, "PID-based target tracking control of a tractor-trailer mobile robot," *Proc. Inst. Mech. Eng., C, J. Mech. Eng. Sci.*, vol. 233, no. 13, pp. 4776–4787, Jul. 2019.

[8] M. M. Fateh and A. Arab, "Robust control of a wheeled mobile robot by voltage control strategy," *Nonlinear Dyn.*, vol. 79, no. 1, pp. 335–348, Jan. 2015, doi: 10.1007/s11071-014-1667-8.

[9] Z.-G. Hou, A.-M. Zou, L. Cheng, and M. Tan, "Adaptive control of an electrically driven nonholonomic mobile robot via backstepping and fuzzy approach," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 4, pp. 803–815, Jul. 2009, doi: 10.1109/Tcst.2009.2012516.

[10] L. Xin, Q. Wang, J. She, and Y. Li, "Robust adaptive tracking control of wheeled mobile robot," *Robot. Auto. Syst.*, vol. 78, pp. 36–48, Apr. 2016, doi: 10.1016/j.robot.2016.01.002.

[11] S. Peng and W. Shi, "Adaptive fuzzy output feedback control of a nonholonomic wheeled mobile robot," *IEEE Access*, vol. 6, pp. 43414–43424, 2018, doi: 10.1109/Access.2018.2862163.

[12] M. Boukattaya, N. Mezghani, and T. Damak, "Adaptive nonsingular fast terminal sliding-mode control for the tracking problem of uncertain dynamical systems," *ISA Trans.*, vol. 77, pp. 1–19, Jun. 2018, doi: 10.1016/j.isatra.2018.04.007.

[13] G. Bai, L. Liu, Y. Meng, W. Luo, Q. Gu, and J. Wang, "Path tracking of wheeled mobile robots based on dynamic prediction model," *IEEE Access*, vol. 7, pp. 39690–39701, 2019, doi: 10.1109/ACCESS.2019.2903934.

[14] T. P. Nascimento, C. E. T. Dórea, and L. M. G. Gonçalves, "Nonlinear model predictive control for trajectory tracking of nonholonomic mobile robots: A modified approach," *Int. J. Adv. Robotic Syst.*, vol. 15, no. 1, pp. 1–14, 2018, doi: 10.1177/1729881418760461.

[15] S. Roy, S. Nandy, R. Ray, and S. N. Shome, "Robust path tracking control of nonholonomic wheeled mobile robot: Experimental validation," *Int. J. Control, Autom. Syst.*, vol. 13, no. 4, pp. 897–905, Aug. 2015.

[16] H. Mirzaeinejad, "Optimization-based nonlinear control laws with increased robustness for trajectory tracking of non-holonomic wheeled mobile robots," *Transp. Res. C, Emerg. Technol.*, vol. 101, pp. 1–17, Apr. 2019, doi: 10.1016/j.trc.2019.02.003.

[17] S. Li, L. Ding, H. Gao, C. Chen, Z. Liu, and Z. Deng, "Adaptive neural network tracking control-based reinforcement learning for wheeled mobile robots with skidding and slipping," *Neurocomputing*, vol. 283, pp. 20–30, Mar. 2018, doi: 10.1016/j.neucom.2017.12.051.

[18] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, Aug. 2013, doi: 10.1177/0278364913495721.
[19] L. Zuo, X. Xu, C. Liu, and Z. Huang, "A hierarchical reinforcement learning approach for optimal path tracking of wheeled mobile robots," *Neural Comput. Appl.*, vol. 23, nos. 7–8, pp. 1873–1883, Dec. 2013, doi: 10.1007/s00521-012-1243-4.
[20] I. Carlucho, M. De Paula, S. Wang, Y. Petillot, and G. G. Acosta, "Adaptive low-level control of autonomous underwater vehicles using deep reinforcement learning," *Robot. Auto. Syst.*, vol. 107, pp. 71–86, Sep. 2018, doi: 10.1016/j.robot.2018.05.016.
[21] Y.-J. Liu, L. Tang, S. Tong, C. L. P. Chen, and D.-J. Li, "Reinforcement learning design-based adaptive tracking control with less learning parameters for nonlinear discrete-time MIMO systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 165–176, Jan. 2015, doi: 10.1109/TNNLS.2014.2360724.
[22] S. Li, L. Ding, H. Gao, Y.-J. Liu, N. Li, and Z. Deng, "Reinforcement learning neural network-based adaptive control for state and input time-delayed wheeled mobile robots," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 11, pp. 4171–4182, Nov. 2020, doi: 10.1109/TSMC.2018.2870724.
[23] Z. Yang, K. Merrick, L. Jin, and H. A. Abbass, "Hierarchical deep reinforcement learning for continuous action control," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5174–5184, Nov. 2018, doi: 10.1109/TNNLS.2018.2805379.
[24] B. Rubí, B. Morcego, and R. Pérez, "Deep reinforcement learning for quadrotor path following with adaptive velocity," *Auton. Robots*, pp. 1–16, Oct. 2020, doi: 10.1007/s10514-020-09951-8.
[25] K. Chatzilygeroudis, V. Vassiliades, F. Stulp, S. Calinon, and J.-B. Mouret, "A survey on policy search algorithms for learning robot controllers in a handful of trials," *IEEE Trans. Robot.*, vol. 36, no. 2, pp. 328–347, Apr. 2020, doi: 10.1109/TRO.2019.2958211.
[26] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 7559–7566.
[27] Y. Hu, W. Wang, H. Liu, and L. Liu, "Reinforcement learning tracking control for robotic manipulator with kernel-based dynamic model," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3570–3578, Sep. 2020, doi: 10.1109/TNNLS.2019.2945019.
[28] W. Shi, S. Song, C. Wu, and C. L. P. Chen, "Multi pseudo Q-learning-based deterministic policy gradient for tracking control of autonomous underwater vehicles," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 12, pp. 3534–3546, Dec. 2019, doi: 10.1109/TNNLS.2018.2884797.
[29] M. Han, L. Zhang, J. Wang, and W. Pan, "Actor-critic reinforcement learning for control with stability guarantee," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 6217–6224, Oct. 2020, doi: 10.1109/LRA.2020.3011351.
[30] P. Cai, X. Mei, L. Tai, Y. Sun, and M. Liu, "High-speed autonomous drifting with deep reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1247–1254, Apr. 2020, doi: 10.1109/LRA.2020.2967299.
[31] R. S. Sutton and A. G. Barto, "The reinforement learning problem," in *Reinforcement learning: An Introduction*, 2th ed. Cambridge, MA, USA: MIT Press, 2012, ch. 3, sec. 6, pp. 58–59.
[32] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016, pp. 1329–1338.
[33] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. 31st Int. Conf. Mach. Learn.*, vol. 32, Jun. 2014, pp. 387–395.

XUESHAN GAO received the M.S. degree in mechanical engineering from the Harbin Institute of Technology and Miyazaki University, in 1996, and the Ph.D. degree in mechanical engineering from the Harbin Institute of Technology, in 2002.

He is currently a Professor and a Ph.D. Supervisor with the Beijing Institute of Technology, China. His research interests include mobile robot and medical robot. He is a Senior Member of the Chinese Mechanical Engineering Society (CMES).

RUI GAO received the B.S. degree in mechanical engineering from Northeast Petroleum University, China, in 2014. He is currently pursuing the Ph.D. degree with the School of Mechatronical Engineering, Beijing Institute of Technology, Beijing, China.

His currently interests include autonomous navigation of mobile robot, including motion control, VSLAM, motion planning, and autonomous control of wheeled mobile robot.

PENG LIANG received the M.S. degree in mechanical engineering from the Guangxi University of Science and Technology, China, in 2017. He is currently pursuing the Ph.D. degree with the School of Mechatronical Engineering, Beijing Institute of Technology, Beijing, China. His currently interests include autonomous mobile robot, multi-freedom robot modeling, and robust control.

QINGFANG ZHANG received the M.S. degree in mechanical engineering from the North China University of Technology, China, in 2019. She is currently pursuing the Ph.D. degree with the School of Mechatronical Engineering, Beijing Institute of Technology, Beijing, China. Her currently interests include autonomous mobile robot and autonomous control of special mobile robot.

RUI DENG received the B.S. degree in mechanical engineering from the Beijing Institute of Technology, Beijing, China, in 2020, where he is currently pursuing the M.S. degree with the School of Mechatronical Engineering. His currently interests include motion control of mobile robot and deep reinforcement learning.

WEI ZHU received the M.S. and Ph.D. degrees in mechanical engineering from the Harbin Institute of Technology, in 2002 and 2006, respectively.

He is currently a Professor with the Beijing Institute of Technology, China. His research interests include machine vision, visual inspection, and deep reinforcement learning.