J. Appl. Comput. Mech., 12(3) (2026) 739-758
DOI: 10.22055/jacm.2025.48406.5213

ISSN: 2383-4536
jacm.scu.ac.ir

Shahid Chamran University of Ahvaz

Journal of Applied and Computational Mechanics

Research Paper

Adaptive Sliding Mode Control for Robust Trajectory Tracking of Car-Like Mobile Robots in Autonomous Navigation Systems

Son-Tung Dang[1], Xuan-Minh Dinh[2], Tuan-Anh Than Ngoc[3], Van-Tuan Nguyen[2], Dinh-Thai Kim[1], Xuan-Hai Le[1]

[1] International School, Vietnam National University, Hanoi (VNU-IS), Xuan Thuy Campus: Buildings E5 & G7, 144 Xuan Thuy, Cau Giay, Hanoi 100000, Vietnam, Email: 22078009@vnu.edu.vn (S.-T.D.); thaikd@vnu.edu.vn (D.-T.K.); hailx@vnu.edu.vn (X.-H.L.)

[2] Faculty of Mechanical Engineering and Mechatronics, Phenikaa University, Duong Noi, Hanoi 12116, Vietnam, Email: minh.dinhxuan@phenikaa-uni.edu.vn (X.-M.D.); tuan.nguyenvan@phenikaa-uni.edu.vn (V.-T.N.)

[3] HTI Group, 15F – VP2, Sun Square Building, 21 Le Duc Tho Street, Tu Liem Ward, Hanoi 100000, Vietnam, Email: anh.tnt@htigroup.vn

Received April 28 2025; Revised September 22 2025; Accepted for publication September 24 2025.
✉ Corresponding author: X.-H. Le (hailx@vnu.edu.vn)

Abstract. This study proposes an Adaptive Nonsingular Fast Terminal Sliding Mode Control (Adaptive-NFTSMC) control method for four-wheeled Car-Like Mobile Robots (CLMRs) to enhance trajectory-tracking performance. The controller extends the Nonsingular Fast Terminal Sliding Mode Control (NFTSMC) technique with Lyapunov-based adaptive laws that update control parameters and estimate the upper bound of disturbances. As a result, the system is able to effectively cope with uncertainties and disturbances without pre-assumptions on amplitude limits, overcoming the limitations commonly found in traditional NFTSMC. The approach features a simple structure, rapid response, absence of chattering, high tracking accuracy, robust stability, singularity avoidance, and finite-time convergence. Moreover, Adaptive-NFTSMC is designed for straightforward integration into standard autonomous navigation stacks, enhancing its practical applicability. Closed-loop stability is rigorously proven through Lyapunov analysis and verified via MATLAB/Simulink simulations. Simulation results show that Adaptive-NFTSMC outperforms classical SMC and an existing Adaptive NFTSMC (ANFTSMC), particularly under conditions with uncertainties and disturbances.

Keywords: Car-Like Mobile Robots, Nonsingular Fast Terminal Sliding Mode Control, Adaptive Nonsingular Fast Terminal Sliding Mode Control, Adaptive Trajectory Tracking Control.

1. Introduction

1.1. Motivation

The rapid development of mobile robots offers substantial economic benefits and significant opportunities across industry, defense, and service sectors. In industrial environments, mobile robots support smart warehouses and production lines by optimizing material handling and reducing costs. In defense applications, they perform surveillance, reconnaissance, and rescue missions to ensure safety in hazardous conditions. They are also crucial in healthcare assistance, intelligent transportation, and public services . With the global growth of the automotive industry, Reduced-Scale Mobile Robots (RSMRs) [1] such as Car-Like Mobile Robots (CLMRs) have become a central research focus and serve as platforms for testing control algorithms. Autonomous navigation systems, which include global planning, local planning, and trajectory-tracking modules, are under active development to enable robots to operate automatically, accurately, and efficiently. Among these, trajectory tracking directly impacts performance and stability.

Our goal is to propose an adaptive trajectory-tracking control method for a Car-Like Mobile Robot (CLMR) that features a simple structure, fast response, chattering-free operation, high accuracy, strong stability, singularity avoidance, and finite-time convergence. This method is designed for easy integration into navigation stacks on the Robot Operating System (ROS), enhancing real-world trajectory-tracking capabilities. Future experiments will be conducted on the CLMR platform shown in Fig. 1, which comprises a front steering mechanism and rear differential drive. Also, Fig. 2 demonstrates in detail the system's schematic. The robot is equipped with a Lidar A2 sensor to gather environmental data and create a surrounding map, along with a Human-Machine Interface (HMI) to monitor operations. The data is processed on the Jetson Nano Developer Kit embedded computer, then transmits linear velocity and reference steering velocity commands to the Arduino Mega 2560 R3 microcontroller. The Arduino is responsible for controlling the motors through two H-driver power circuits, and at the same time receives feedback signals from the encoders. In addition, the ESP32 module connects to the Arduino and transmits data to the Firebase platform, thereby updating the system status to the monitoring website. Supporting devices and power supplies are also fully integrated to ensure stable system operation.

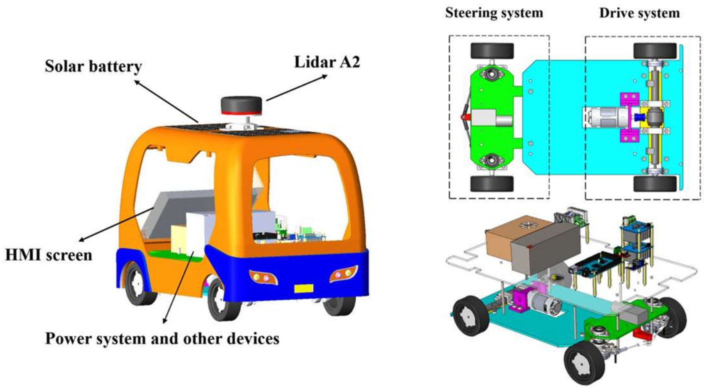

Fig. 1. Experimental Car-Like Mobile Robot system.

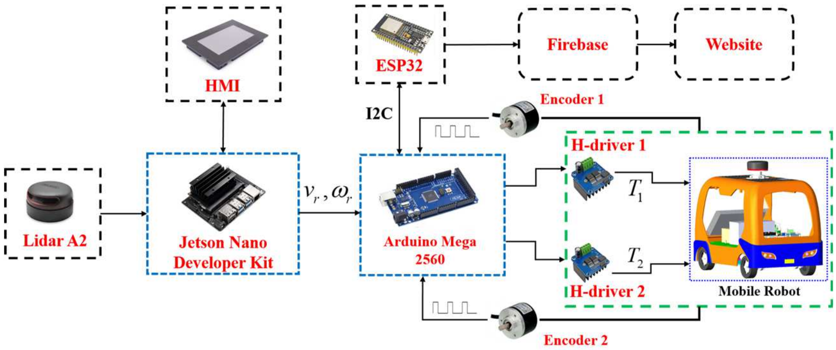

Fig. 2. Experimental Car-Like Mobile Robot system's schematic.

1.2. Related work

A considerable body of research has addressed the trajectory-tracking problem for mobile robots, demonstrating effectiveness across a range of control approaches. These include Proportional–Integral–Derivative (PID) control [2, 4], Sliding Mode Control (SMC) and Backstepping [5–12], Nonlinear Adaptive Control and Intelligent Control Methods [13–21], State Observer Design Techniques [22–24], and Model Predictive Control (MPC) [25–30].

Karthik et al. [2] proposed a PID controller to ensure that the robot maintains the desired lane position in a low-cost indoor navigation system. The PID method offers simplicity and ease of implementation. However, it struggles with highly nonlinear systems and requires careful tuning of the Proportional (P), Integral (I), and Derivative (D) gains to avoid oscillation and overshoot. Integrator windup can occur when the integral term continues to accumulate error even after actuator saturation, causing sluggish corrective action and reduced performance. To mitigate this, the authors incorporated an anti-windup loop and employed a Finite-Impulse-Response (FIR) filter to attenuate high-frequency noise, thereby improving stability and overall effectiveness. Cáceres Flórez et al. [3] also applied PID for a CLMR to track reference trajectories; despite achieving the control objectives, the system response was relatively slow and did not guarantee finite-time convergence. These limitations suggest that PID is not optimal for highly nonlinear systems or complex trajectories. To improve the performance, Numbi et al. [4] combined PD control with the Quantum-Behaved Particle Swarm Optimization (QPSO) algorithm to tune the controller parameters for CLMR trajectory tracking. The method showed better optimization capability than classical Particle Swarm Optimization (PSO), yet its theoretical accuracy advantage has not been fully realized in practice because of computational constraints in the absence of dedicated quantum hardware. For this reason, alternative approaches such as nonlinear, adaptive, or intelligent control based on fuzzy logic or neural networks may offer more effective solutions thanks to their superior adaptability to disturbances and uncertainties.

Within this context, Sliding Mode Control (SMC) constitutes an important branch of modern control theory and has been successfully applied in many fields [5]. The works [6–10] have implemented nonlinear control strategies for mobile robots based on SMC and Backstepping. The SMC approach in [6] is praised for its strong disturbance rejection and system robustness; however, it relies heavily on an accurate mathematical model and employs discontinuous switching functions that readily cause chattering, high-frequency oscillations in the control signal that degrade stability and performance, especially in highly nonlinear systems such as mobile robots. According to [11], this is the main shortcoming of SMC, and numerous techniques have been proposed to mitigate it. Recently, Yılmaz and Çorapsız [12] combined Higher-Order SMC (HOSMC) with Gaussian Process Regression (GPR), whose hyperparameters are optimized. Although this HOSMC-GPR scheme delivers impressive results, its computational cost limits real-time feasibility on CLMR and other resource-constrained platforms. Furthermore, SMC may converge slowly to the sliding surface. To address these issues, Nonsingular Fast Terminal SMC (NFTSMC) [7] and fast integral terminal SMC (FITSMC) [8] have been developed, offering superior performance and stability; nevertheless, the method in [8] still depends strongly on the system model and becomes unstable in the presence of disturbances. Other hybrid approaches that integrate the advantages of various nonlinear techniques have also been explored: Huang and Gao [9] introduced a new SMC algorithm combined with Backstepping for CLMR, while Sen et al. [10] proposed a Backstepping scheme integrated with Hierarchical SMC (HSMC) to improve trajectory tracking. Backstepping has proven effective owing to its recursive Lyapunov design and simple structure, yet it still depends on an accurate model and can suffer from the "explosion of complexity" when the computational load grows in strongly nonlinear systems. HSMC, for its part, reduces chattering while retaining robustness and disturbance rejection by decomposing the control into multiple sliding layers, thereby lowering the control-signal amplitude and enhancing performance in noisy environments; however, it likewise hinges on an accurate model and requires experience to tune the control parameters for each layer. These limitations have spurred the development of adaptive control algorithms that allow robots to cope more effectively with uncertainties and nonlinearities.

The studies [13–20] have proposed adaptive nonlinear control methods to enhance the trajectory-tracking ability of mobile robots. In [13], an Adaptive Backstepping technique for CLMR automatically tunes the gains based on the tracking error and the desired vehicle speed, achieving precise path following without violating velocity and steering constraints; however, like Huang and Gao [9], it considers only the kinematic model and neglects system dynamics. Although this reduces computational cost, accounting for dynamics is essential to improve accuracy and stability when CLMR operates at high speed or in complex scenarios. Zheng et al. [14] presented a barrier-function-based adaptive sliding-mode controller (BFASMC) designed to enhance tracking accuracy, transient response, and robustness. In contrast to conventional adaptive sliding-mode control, the proposed approach ensured convergence of the sliding variable to a neighborhood of the origin within a predefined time. The barrier function dynamically adjusted the control gain according to the disturbance magnitude, thereby reducing chattering, while an improved barrier formulation mitigated actuator saturation arising from physical constraints. While these contributions have clear theoretical significance, further research is warranted to broaden the range of applicable control strategies. Such work could expand the methodological repertoire available to practitioners, particularly in resource-constrained environments. It is also noted that the validation in Zheng et al. [14] is conducted on a two-rear-wheel differential-drive robot, which may not fully represent the complexities encountered in more sophisticated systems, such as CLMRs widely used in autonomous vehicle research.

The studies [15, 16] have developed intelligent control methods for mobile robots that exploit artificial neural networks. Dang et al. [15] proposed integrating a Radial-Basis-Function (RBF) Neural Network with an Adaptive Backstepping Hierarchical Sliding-Mode Controller (ABHSMC) to enhance trajectory tracking. The RBF network estimates unknown nonlinear model components, thereby improving accuracy and adaptability. Although simulation and experimental results confirm the effectiveness of ABHSMC, these works do not examine the system's finite-time convergence. More recently, Ly et al. [16] designed a Neural-Network Controller using a perceptron with three hidden layers trained by backpropagation on data from a time-varying-parameter PID controller. This approach achieves small tracking errors and good real-time adaptability, yet it is computationally intensive, especially on resource-constrained embedded hardware. Neural Network performance also hinges on training-data quality: when confronted with unseen trajectories or scenarios, the robot may respond inaccurately or with sub-optimal control. In short, neural-network-based controllers offer benefits but still face significant challenges related to complexity, data requirements, and computational demand.

In addition to Neural Network approaches, Fuzzy Logic [17-21] is widely applied to trajectory-tracking control, providing advantages over classical controllers. Ben Jabeur and Seddik [17] introduced two Mamdani Fuzzy Controllers for a mobile robot's drive system, achieving better performance and stability than a PID controller tuned via the Ziegler-Nichols method. Study [18] employed Type-2 Fuzzy Logic in a cooperative control strategy for CLMR so that linear and steering velocities converge to reference values; online parameter estimation allowed faster, more stable convergence than a grey-wolf-optimized scheme. Shirzadeh et al. [19] proposed a Fuzzy Nonlinear Sliding Mode Controller that reduces oscillations by tuning the gain of the sign function through a Fuzzy system, with parameters optimized by a genetic algorithm; it proved effective against disturbances and maintained desired trajectories. Nevertheless, these studies [17, 19] still rely on designer expertise: as noted in Shui et al. [20], crafting Fuzzy Rules requires careful tuning, or control performance may degrade in complex situations. Dinh et al. [21] presented a data-driven Generalized Predictive Controller (GPC) based on an Interval Type-2 T-S Fuzzy Neural Network (IT2TSFNN) for CLMR trajectory tracking. Dispensing with an explicit robot model, it uses historical operational data; simulations show that it mitigates uncertainty effects and yields high tracking accuracy, though its effectiveness remains limited by the quality of the collected data.

Observer-based techniques [22–24] have been proposed to accurately estimate disturbances. In Li et al. [22], a nonlinear disturbance observer combined with an Extended Kalman Filter (EKF) estimates random disturbances and mitigates stochastic noise, achieving high accuracy and fast convergence, although at the cost of increased computational complexity. Zhu et al. [23] integrated a Linear Extended State Observer (LESO) with a Backstepping controller to improve trajectory-tracking precision for CLMR; simulations and experiments validate its efficacy, but reliance on a purely kinematic model limits performance in high-speed or complex scenarios. Hussein et al. [24] used a High-Order Sliding-Mode Observer (HOSM) to estimate system states, paired with a Super-Twisting Controller (STC) to eliminate chattering and compensate for actuator faults. This approach has been validated in simulation and experiments, demonstrating robust trajectory tracking; however, designing and implementing such adaptive observer-based controllers requires advanced expertise, posing practical challenges for researchers and engineers new to the field.

In recent years, Model Predictive Control (MPC) techniques have been applied with growing frequency to mobile robots and autonomous vehicles [25-30]. Yeom [25] implemented a simple MPC framework for CLMR that guides the vehicle to its destination using GPS signals; road tests showed that the robot successfully reached the final goal. MPC is a flexible and efficient optimization approach, particularly effective at handling system constraints to achieve control objectives. Its drawbacks, however, include high computational demand, especially for nonlinear systems, and strong dependence on model accuracy, which can reduce performance when the model is uncertain. Cao et al. [26] developed a Linear Time-Varying MPC (LTVMPC) for trajectory tracking, focusing on cornering scenarios where maintaining precision and real-time performance is difficult; simulations demonstrated improved disturbance rejection, robustness, and tracking accuracy in complex environments, while requiring fewer computations than earlier MPC techniques proposed in Wang et al. [27]. Zhang et al. [28] proposed an MPC controller with input constraints satisfying a Lyapunov condition (LMPC) that eliminates the heavy computational load of terminal constraints during tracking, thereby lowering the required resources. In another work conducted by Bai et al. [29], a Neural-Network-Assisted Nonlinear MPC (NMPC) strategy was presented for CLMs, using a neural network to emulate an NMPC controller; simulations indicated better real-time performance than pure NMPC. Yeom [30] introduced a hybrid architecture that combines a Deep Neural Network (DNN) with MPC to optimize steering angle, velocity, and acceleration during trajectory tracking. The DNN models vehicle dynamics from data generated by a PID controller, and this model is then used to refine MPC during operation, achieving better tracking than standalone MPC. Nevertheless, implementing MPC-based techniques [25-30] still requires accurate models, specialized expertise, and powerful computing hardware, which hampers practical deployment.

Building on this analysis, the present study develops an adaptive control scheme for CLMR systems that combines a simple structure, rapid reference tracking, chattering suppression, high tracking accuracy, strong stability, singularity avoidance, and finite-time convergence, referred to as Adaptive-NFTSMC. The proposed method is constructed by integrating NFTSMC with adaptive adjustment laws designed based on Lyapunov stability analysis. Specifically, Adaptive-NFTSMC incorporates two adaptive laws, including one for updating the control parameters and another for estimating the upper bound of external disturbances. This adaptive mechanism enables the system to respond more effectively to uncertainties and unknown perturbations, while maintaining stability without requiring prior assumptions on the boundedness of disturbances and uncertainties, a common requirement in traditional NFTSMC controllers. Closed-loop stability is rigorously proven using Lyapunov theory and confirmed through numerical simulation in MATLAB/Simulink. Performance is compared with classical SMC [6] and an alternative adaptive NFTSMC (ANFTSMC) [7]; simulations show that the proposed algorithm not only yields superior accuracy but also maintains high control quality when facing external uncertainties and disturbances. Unlike existing adaptive methods, Adaptive-NFTSMC avoids the complexity and data requirements of Neural Network approaches [15, 16], does not rely on expert-defined rules as in Fuzzy-Logic

Controllers [17-21], is easier to implement than observer-based schemes [22-24], and does not demand the heavy computation associated with MPC [25-30]. In contrast to the work conducted by Zheng et al. [14], the present work replaces the barrier-function-based design with an NFTSMC framework that incorporates an online adaptation mechanism for both control parameters and the upper bound of disturbances. This enables the controller to operate without prior knowledge of the disturbance limit while still guaranteeing finite-time convergence. The proposed scheme adopts a simpler structure, facilitating practical implementation even by engineers with limited experience in control design. Furthermore, Adaptive-NFTSMC broadens the range of available control solutions for the engineering community, particularly in resource-constrained environments, due to its structural simplicity and ease of deployment.

1.3. Contributions, paper organization, and notation

The main contributions of this research are as follows:

- Developed an adaptive control scheme, Adaptive-NFTSMC, for CLMR systems by integrating NFTSMC with Lyapunov-based adaptive laws. Specifically, the controller incorporates two adaptive mechanisms, including one for automatic gain tuning and another for estimating an upper bound on disturbances. This design enables CLMR systems to cope more effectively with uncertainties and external perturbations without requiring prior assumptions on their boundedness, overcoming a common limitation of traditional NFTSMC.
- Adaptive-NFTSMC features a simple structure, fast reference tracking, no chattering, high accuracy, strong robustness, singularity avoidance, and finite-time convergence. Closed-loop stability is proven with Lyapunov theory and validated through MATLAB/Simulink simulations. Results show that Adaptive-NFTSMC outperforms both SMC and ANFTSMC, especially under uncertain and noisy conditions.

The remainder of this paper is organized as follows: Section 2 describes the CLMR system, including its mathematical model and autonomous navigation stack. Section 3 presents the proposed control method along with stability analysis. Section 4 provides simulation results and detailed discussion. Finally, Section 5 offers conclusions and outlines future research directions.

In this paper, $\mathbb{R}^{n\times m}$ denotes a matrix space with n rows and m columns. $\mathbb{R}^{n}$ is a column vector space with n real numbers elements, where $\mathbf{x}=[x_1,\ldots,x_n]^T\in\mathbb{R}^n$. $|\mathbf{x}|^\gamma = diag[|x_1|^\gamma,\ldots,|x_n|^\gamma]$ with $\gamma>0$, $|x_i|$ is the absolute value of the vector x_i with $i=1,\ldots,n$. $sig^\gamma(\mathbf{x})=[|x_1|^\gamma sign(x_1),\ldots,|x_n|^\gamma sign(x_n)]^T$, $d(sig^\gamma(\mathbf{x}))/dt=\gamma|\mathbf{x}|^{\gamma-1}\dot{\mathbf{x}}$. Moreover, $\|x\|_\infty$ is the infinity norm of the vector $\mathbf{x}$, and $sign(x_i)$ is the sign function of real numbers x_i.

2. Description of the CLMR System

This section introduces the mathematical model of the CLMR and outlines its autonomous navigation stack. The overview of the stack highlights the controller design objectives and provides the basis for the method proposed in this study.

2.1. Mathematical model of the CLMR

The mathematical model of CLMRs has already been developed and described in detail in earlier studies such as [4, 7, 13, 24]. It is derived from the Ackermann steering principle, which locates the instantaneous center of rotation O' along the rear axle. In Fig. 3, this center O' is obtained as the intersection of two straight lines, CO' and FO', that are drawn perpendicular to the headings of the front and rear wheels, respectively, with C being the midpoint of the rear axle. For simplicity, the kinematic model can be converted to a bicycle-type model, where the pair of rear wheels and the pair of front wheels are each represented by a single wheel. The front wheel steers while the rear wheel rolls in a fixed direction. In this study, the mathematical model of the CLMR system is formulated based on several basic assumptions. The robot operates on a two-dimensional plane defined by the OXY coordinate frame, moves at low speed, and is represented using the bicycle model, which is considered sufficient to capture its motion characteristics. The kinematics are derived from planar geometric relations under the assumption of zero lateral velocity, meaning no side-slip relative to the body's longitudinal axis. Under low-speed conditions, it is also reasonable to assume pure rolling without wheel slip, as reported in [13].

Figure 3 defines the model parameters: $L(m)$ is the wheelbase, i.e., the distance between the front and rear axles; $l(m)$ is half the track width, affecting the overall width of the robot; and b is the distance from the center of mass to the midpoint of the rear axle, influencing inertial properties. The linear velocity and steering velocity are denoted by $v(m/s)$ and $\omega(rad/s)$, respectively, are the control inputs. The state variables x,y give the position, $\theta(rad)$ is the heading angle measured counter-clockwise from the X-axis, and $\delta(rad)$ is the steering angle in the global OXY frame, bounded by the mechanical limits of the steering linkage. R denotes the distance from the instantaneous center O' to the midpoint C of the rear axle. Physical parameters such as the wheelbase L directly affect the turning radius and cornering stability, while the commanded v and ω are adjusted to drive the robot along the desired trajectory.

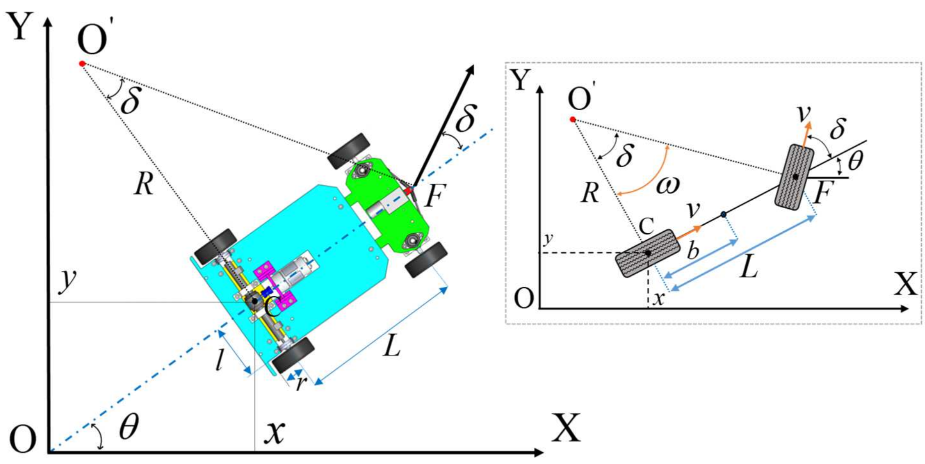

Fig. 3. Mathematical model of CLMR system.

The kinematic model of CLMR is established from geometric relations in the global frame OXY, as illustrated in Fig. 3. A concise analysis of this model appears in [13]; here, we give a more detailed derivation. First, because the robot moves with a linear speed υ along its body axis, its velocity in the global frame can be decomposed into X and Y components via trigonometric projections:

$$\begin{cases} \dot{x} = \upsilon\cos(\theta) \\ \dot{y} = \upsilon\sin(\theta) \end{cases} \tag{1}$$

where $\dot{x}$ and $\dot{y}$ are the translational velocities along the global X and Y axes, respectively.

According to the Ackermann steering principle, the robot turns about an instantaneous centre of rotation O'. The turning radius R, measured from the midpoint C of the rear axle to O', is given by the geometric relation:

$$R = \frac{L}{\tan(\delta)} \tag{2}$$

with δ is the front-wheel steering angle in the global frame and L is the wheel base of the robot.

From Eq. (2), we readily derive the following physical relationship, where $\dot{\theta}$ is the robot's angular velocity:

$$\dot{\theta} = \frac{\upsilon}{R} = \left(\frac{\upsilon}{L}\right)\tan(\delta) \tag{3}$$

where $\dot{\theta}$ is the robot's angular velocity.

Finally, the steering angle $\dot{\delta}$ or steering rate ω is one of the two principal control inputs in CLMR motion control. Their relationship is:

$$\dot{\delta} = \omega \tag{4}$$

Combining Eq. (1) through Eq. (4), the complete kinematic model of a CLMR is expressed as the system:

$$\begin{cases} \dot{x} = \upsilon\cos(\theta) \\ \dot{y} = \upsilon\sin(\theta) \\ \dot{\theta} = \upsilon / R = (\upsilon / L)\tan(\delta) \\ \dot{\delta} = \omega \end{cases} \tag{5}$$

At this stage, we define the robot's generalized state vector as:

$$\mathbf{q} = \begin{bmatrix} x & y & \theta & \delta \end{bmatrix}^T \tag{6}$$

To obtain the dynamic model, crucial for designing and controlling CLMR, we employ the Euler-Lagrange equation [4]:

$$\frac{d}{dt}\left(\frac{\delta L(\mathbf{q},\dot{\mathbf{q}},t)}{\delta\dot{\mathbf{q}}}\right) - \frac{\delta L(\mathbf{q},\dot{\mathbf{q}},t)}{\delta\mathbf{q}} = \mathbf{Q} \tag{7}$$

where $\dot{\mathbf{q}}$ is the first-time derivative of $\mathbf{q}$, $\delta L(\mathbf{q},\dot{\mathbf{q}},t)$ is the Larangian, and $\mathbf{Q}$ is the generalized force/torque vector acting on the system. The Lagrangian is defined as [4]:

$$L(\mathbf{q},\dot{\mathbf{q}},t) = KE - PE \tag{8}$$

with KE is the kinetic energy and PE is the potential energy.

Because the CLMR moves on level ground, the potential energy is zero, so Eq. (8) reduces to the kinetic energy only:

$$L(\mathbf{q},\dot{\mathbf{q}},t) = KE = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) + I_c(\dot{x}\sin(\theta) - \dot{y}\cos(\theta))\dot{\theta} + \frac{1}{2}I_b\dot{\theta}^2 + I_w\dot{\theta}\dot{\delta} + \frac{1}{2}I_w\dot{\delta}^2 \tag{9}$$

here, $I_b = m_c(L-b)^2 + 4l^2m_w + I_{bzz} + 4I_{wzz}, I_c = m_c(L-b) + 2Lm_w, I_w = 2I_{wzz}, m = m_c + 4m_w$ with I_w is the moment of inertia of each wheel about the wheel axle, I_{bzz}, I_{wzz} are the moments of inertia of the mobile robot and each wheel about the z-axis, respectively, m_c and m_w are the masses of the mobile base and each wheel about the vertical z-axis (perpendicular to the plane of motion, about which yaw rotation occurs).

The generalized force/torque vector is:

$$\mathbf{Q} = \mathbf{B}(\mathbf{q})\mathbf{T} - \mathbf{A}^T(\mathbf{q})\lambda \tag{10}$$

where $\mathbf{q} \in \mathbb{R}^{4\times1}$ is the generalized coordinate vector, $\mathbf{T} \in \mathbb{R}^{2\times1}$ is the input torque vector, $\mathbf{B}(\mathbf{q}) \in \mathbb{R}^{4\times2}$ is the transformation matrix, $\mathbf{A}(\mathbf{q}) \in \mathbb{R}^{2\times4}$ is the constraint matrix, and $\lambda \in \mathbb{R}^{2\times1}$ is the vector of Lagrange multipliers. By substituting the Lagrangian expressions from Eq. (9) and Eq. (10) into the Euler–Lagrange Eq. (7), we obtain the CLMR system's dynamic model in the following form [7, 24]:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{V}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} = \mathbf{B}(\mathbf{q})\mathbf{T} - \mathbf{A}^T(\mathbf{q})\lambda \tag{11}$$

in which, $\ddot{\mathbf{q}}$ is the second derivative with respect to time of the state vector $\mathbf{q}$, $\mathbf{q} \in \mathbb{R}^{4\times1}$ is the coordinates of the robot, $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{4\times4}$ is a positive definite symmetry matrix, $\mathbf{V}(\mathbf{q},\dot{\mathbf{q}}) \in \mathbb{R}^{4\times4}$ is the Coriolis and centrifugal force matrices, $\mathbf{T} \in \mathbb{R}^{2\times1}$ is the input control torque vector. Specifically, these matrices and vectors are calculated as follows [7, 24]:

$$\mathbf{V}(\mathbf{q},\dot{\mathbf{q}}) = \begin{bmatrix} 0 & 0 & I_c\dot{\theta}\cos(\theta) & 0 \\ 0 & 0 & I_c\dot{\theta}\sin(\theta) & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{B}(\mathbf{q}) = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ L\sin(\delta)\cos(\delta) & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{M}(\mathbf{q}) = \begin{bmatrix} m & 0 & I_c\sin(\theta) & 0 \\ 0 & m & -I_c\cos(\theta) & 0 \\ I_c\sin(\theta) & -I_c\cos(\theta) & I_b & I_w \\ 0 & 0 & I_w & I_w \end{bmatrix}, \mathbf{A}^T(\mathbf{q})\lambda = \mathbf{A}^T(\mathbf{q})\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} \tag{12}$$

From Eq. (5) and Eq. (6), it can be rewritten as follows:

$$\dot{\mathbf{q}} = \mathbf{S}\mathbf{w} \tag{13}$$

in which:

$$\mathbf{S} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ \dfrac{\tan(\delta)}{L} & 0 \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{4\times 2},\ \mathbf{w} = [\upsilon \quad \omega]^T \in \mathbb{R}^{2\times 1} \tag{14}$$

Substitute Eq. (13) into Eq. (11) with the note that $\mathbf{S}^T\mathbf{A}^T(\mathbf{q}) = 0$. Now we obtain the dynamic equation of CLMR as follows:

$$\bar{\mathbf{M}}(\mathbf{q})\dot{\mathbf{w}} + \bar{\mathbf{V}}(\mathbf{q},\dot{\mathbf{q}})\mathbf{w} = \bar{\mathbf{B}}(\mathbf{q})\mathbf{T} \tag{15}$$

The matrices and vectors in Eq. (15) are defined as follows:

$$\bar{\mathbf{M}}(\mathbf{q}) = \mathbf{S}^T\mathbf{M}(\mathbf{q})\mathbf{S} \in \mathbb{R}^{2\times 2},\ \bar{\mathbf{B}}(\mathbf{q}) = \mathbf{S}^T\mathbf{B}(\mathbf{q}) \in \mathbb{R}^{2\times 2},\ \bar{\mathbf{V}}(\mathbf{q},\mathbf{q}) = \mathbf{S}^T(\mathbf{M}(\mathbf{q})\dot{\mathbf{S}} + \mathbf{V}(\mathbf{q},\dot{\mathbf{q}})\mathbf{S}) \in \mathbb{R}^{2\times 2} \tag{16}$$

In practice, we need to consider the uncertainty in the model and the unknown external disturbance. At this point, Eq. (15) is rewritten as follows:

$$\bar{\mathbf{M}}(\mathbf{q})\dot{\mathbf{w}} + \bar{\mathbf{V}}(\mathbf{q},\dot{\mathbf{q}})\mathbf{w} + \bar{\mathbf{F}}(\mathbf{q},\dot{\mathbf{q}},t) = \bar{\mathbf{B}}(\mathbf{q})\mathbf{T} \tag{17}$$

$$\bar{\mathbf{F}}(\mathbf{q},\dot{\mathbf{q}},t) = \Delta\bar{\mathbf{M}}(\mathbf{q})\dot{\mathbf{w}} + \Delta\bar{\mathbf{V}}(\mathbf{q},\dot{\mathbf{q}})\mathbf{w} + \mathbf{f}(t) + \mathbf{d}(t) \tag{18}$$

in which, $\Delta\bar{\mathbf{M}}(\mathbf{q})$ and $\Delta\bar{\mathbf{V}}(\mathbf{q},\dot{\mathbf{q}})$ are uncertain or incompletely modeled components in the system dynamics model, $\mathbf{f}(t) \in \mathbb{R}^{2\times 1}$ is the friction vector, $\mathbf{d}(t) \in \mathbb{R}^{2\times 1}$ are the disturbing forces.

Now, Eq. (17) is re-represented for controller design purposes as follows:

$$\dot{\mathbf{w}} = \bar{\mathbf{M}}^{-1}(\mathbf{q})[-\bar{\mathbf{V}}(\mathbf{q},\dot{\mathbf{q}})\mathbf{w} - \bar{\mathbf{F}}(\mathbf{q},\dot{\mathbf{q}},t) + \bar{\mathbf{B}}(\mathbf{q})\mathbf{T}] = \bar{\mathbf{M}}^{-1}(\mathbf{q})\tau + \bar{\mathbf{D}}(\mathbf{q},\dot{\mathbf{q}}) + \Delta\bar{\mathbf{F}}(\mathbf{q},\dot{\mathbf{q}}) \tag{19}$$

in which, vector $\bar{\mathbf{D}}(\mathbf{q},\dot{\mathbf{q}}) = -\bar{\mathbf{M}}^{-1}(\mathbf{q})\bar{\mathbf{V}}(\mathbf{q},\dot{\mathbf{q}})\mathbf{w}$ represents the nominal dynamic components that have been predetermined, $\tau = \bar{\mathbf{B}}(\mathbf{q})\mathbf{T}$ is the control signal, and $\Delta\bar{\mathbf{F}}(\mathbf{q},\dot{\mathbf{q}}) = -\bar{\mathbf{M}}^{-1}(\mathbf{q})\bar{\mathbf{F}}(\mathbf{q},\dot{\mathbf{q}},t)$ is the total disturbance component of the system, caused by model error and unknown external disturbance.

Assumption 1: The sum of the uncertain components and the unknown external disturbance is bounded:

$$\left\|\Delta\bar{\mathbf{F}}(\mathbf{q},\dot{\mathbf{q}})\right\|_\infty = \left\|-\bar{\mathbf{M}}^{-1}(\mathbf{q})\bar{\mathbf{F}}(\mathbf{q},\dot{\mathbf{q}},t)\right\|_\infty \le \varepsilon \tag{20}$$

where ε is a positive constant that bounds the combined effect of disturbances and uncertainties, which is not required to be known in advance in this paper.

2.2. Autonomous Navigation Stack

As discussed in the dynamic-analysis section, the proposed Adaptive-NFTSMC algorithm will be embedded in a real-world A*-TEB navigation stack [15] to form the Adaptive-NFTSMC Navigation Stack illustrated in Fig. 4. The architecture combines a global planner based on the A-star (A*) algorithm with a local planner that uses the Time-Elastic-Band (TEB) method for obstacle avoidance. Detailed descriptions of the TEB planner for CLMR systems can be found in [31, 32]. In this work, Adaptive-NFTSMC is integrated into the A*-TEB framework to improve system stability and trajectory-tracking accuracy, thereby enhancing overall navigation performance. Both A* and TEB are open-source packages natively supported by the ROS middleware, which facilitates rapid and efficient deployment in practical applications.

According to [15, 31], within the A*-TEB framework, a CLMR system must follow the locally optimal path computed by TEB at every instant by achieving the reference linear speed υ_r and steering angle δ_r. In practice, however, directly controlling the steering-angle error is problematic: Steering angle is a highly nonlinear quantity that can change abruptly when the path must deviate sharply to avoid obstacles. Immediate compensation is therefore required, yet this is difficult to realize because of the physical limits and actuation lag of mechanical steering, which degrades system stability.

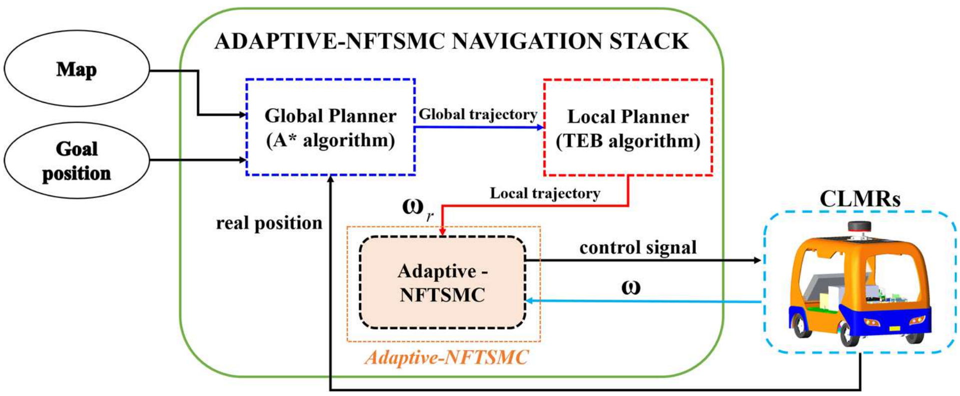

Fig. 4. Adaptive-NFTSMC Navigation Stack Structure for CLMR System.

To address this issue, our Adaptive-NFTSMC Navigation Stack modifies the control objective: instead of regulating the steering angle directly, the CLMR is commanded to track the reference velocity vector $\mathbf{w}_r = [\upsilon_r \quad \omega_r]^T$. When the robot accurately follows these reference velocity values, the calculated optimal trajectory is executed naturally, while maintaining obstacle avoidance and respecting physical constraints. Therefore, controlling the actual velocity to quickly converge to the reference velocity is a prerequisite for ensuring accurate trajectory tracking. In addition, from a practical perspective, velocity is a state variable that can be easily measured with high accuracy using common sensors, which makes the implementation and tuning of the controller simpler and more reliable.

Based on this principle, the study chooses the control objective as velocity tracking, with the control error defined directly from the difference between the actual velocity and the reference velocity. This design makes the Adaptive-NFTSMC algorithm easy to integrate into the navigation stack and suitable for practical applications in the later implementation phase.

3. Proposed Method and Stability Analysis

In automatic control, SMC [6] is renowned for its simple structure and its strong robustness against model uncertainties and environmental disturbances [33]. Based on Lyapunov stability theory, SMC has emerged as a systematic approach that balances performance indices through appropriate tuning of control parameters, enabling the system to reach an asymptotically stable state, and has been widely applied in practice [34]. However, because convergence is only asymptotic, SMC cannot reach equilibrium in a finite time, which is problematic for applications that require rapid response. To overcome this drawback, Fast Terminal Sliding Mode Control (FTSMC) and Non-Singular Fast Terminal Sliding Mode Control (NFTSMC) were developed to ensure finite-time convergence. Although FTSMC accelerates convergence, it suffers from singularities when state variables approach zero. NFTSMC addresses this weakness by employing non-singular sliding surfaces, ensuring fast and stable convergence over the entire operating domain —a desirable feature for mobile-robot control. Nevertheless, NFTSMC remains sensitive to modeling errors and unknown disturbances in practical environments. To address these limitations, Adaptive-NFTSMC is developed by integrating NFTSMC with online adaptive laws derived through Lyapunov stability analysis, using the Lyapunov function defined in Eq. (35). Specifically, the controller incorporates two real-time adaptation mechanisms: a control-parameter update law (Eq. (33)) and a disturbance-bound estimation law (Eq. (34)). The former adjusts the control gain in real time based on the tracking error and its derivative, while the latter estimates and compensates for unknown disturbances without prior knowledge of their characteristics. Both adaptive mechanisms are proven to be Lyapunov-stable, enabling the system to better cope with uncertainties and disturbances while maintaining stability without the prior assumption that such effects must be bounded, thereby overcoming a common limitation of conventional NFTSMC controllers.

The overall structure of the Adaptive-NFTSMC controller is depicted in Fig. 5, illustrating its architectural simplicity and online self-tuning capability. The following sections detail our NFTSMC and Adaptive-NFTSMC controller designs for the CLMR system:

3.1. Nonsingular Fast Terminal Sliding Mode Control for CLMR

With $\bar{\mathbf{M}}^{-1} = \bar{\mathbf{M}}^{-1}(\mathbf{q}), \bar{\mathbf{D}} = \bar{\mathbf{D}}(\mathbf{q},\dot{\mathbf{q}}), \Delta\bar{\mathbf{F}} = \Delta\bar{\mathbf{F}}(\mathbf{q},\dot{\mathbf{q}})$, we can rewrite Eq. (19) as follows:

$$\dot{\mathbf{w}} = \bar{\mathbf{M}}^{-1}\tau + \bar{\mathbf{D}} + \Delta\bar{\mathbf{F}} \tag{21}$$

We set the system error as follows:

$$\mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} \upsilon - \upsilon_r \\ \omega - \omega_r \end{bmatrix} = \mathbf{w} - \mathbf{w}_r \tag{22}$$

with $\mathbf{w}$ is the real velocity vector and $\mathbf{w}_r$ is the reference velocity vector.

Remark 1: The choice of velocity as the deviation is to ensure the stability and fast response of the system when combined with the TEB local planner. In addition, this choice of deviation is more suitable for real-world simulation because velocity is a state variable that is very easy to measure directly.

Derivative Eq. (22) with respect to time, combined with Eq. (21), gives:

$$\dot{\mathbf{e}} = \dot{\mathbf{w}} - \dot{\mathbf{w}}_r = \bar{\mathbf{M}}^{-1}\tau + \bar{\mathbf{D}} + \Delta\bar{\mathbf{F}} - \dot{\mathbf{w}}_r \tag{23}$$

Inspired by studies [33, 34], in this study, the sliding surface is designed as follows:

$$\mathbf{s} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \int_0^t \mathbf{e}dt + \alpha sig^{\varphi}\left(\int_0^t \mathbf{e}dt\right) + \beta sig^{\frac{a}{b}}(\mathbf{e}) \tag{24}$$

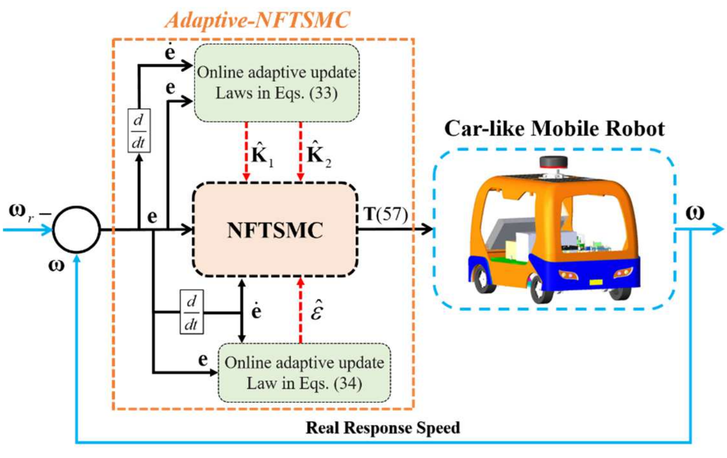

Fig. 5. Structure of the Adaptive-NFTSMC controller for the CLMR system.

where, according to the study [33], we chose a and b as odd positive constants satisfying the condition $1 < a/b < 2$ and $\varphi > a/b$ to easily calculate derivatives and avoid singularities. Although with other constraints, regardless of a/b, we are able to calculate the derivative, but the formula might be more complicated. Moreover, we also chose $\alpha > 0$, $\beta > 0$ as constants to adjust the weights of the components in the sliding surface. Finally, s_1, s_2 are the variables representing the sliding surface of e_1, e_2, respectively. The sliding surface design at Eq. (24) is a common structure, which ensures fast convergence, avoids singularities, and reduces vibration, as verified in [33, 35].

Deriving the sliding surface (24), we get:

$$\dot{\mathbf{s}} = \mathbf{e} + \alpha\varphi\left|\int_0^t \mathbf{e}dt\right|^{\varphi-1}\mathbf{e} + \beta\frac{a}{b}|\mathbf{e}|^{\frac{a}{b}-1}\dot{\mathbf{e}} \tag{25}$$

Substituting Eq. (23) into Eq. (25), we have:

$$\dot{\mathbf{s}} = \mathbf{e} + \alpha\varphi\left|\int_0^t \mathbf{e}dt\right|^{\varphi-1}\mathbf{e} + \beta\frac{a}{b}|\mathbf{e}|^{\frac{a}{b}-1}(\bar{\mathbf{M}}^{-1}\tau + \bar{\mathbf{D}} + \Delta\bar{\mathbf{F}} - \dot{\mathbf{w}}_r) \tag{26}$$

We denote $\chi = \mathbf{e} + \alpha\varphi\left|\int_0^t \mathbf{e}dt\right|^{\varphi-1}\mathbf{e} \in \mathbb{R}^{2\times1}$ and $\xi = \beta\frac{a}{b}|\mathbf{e}|^{\frac{a}{b}-1} \in \mathbb{R}^{2\times2}$. Thus, Eq. (26) can be rewritten as follows:

$$\dot{\mathbf{s}} = \chi + \xi(\bar{\mathbf{M}}^{-1}\tau + \bar{\mathbf{D}} + \Delta\bar{\mathbf{F}} - \dot{\mathbf{w}}_r) \tag{27}$$

Based on the study [8], our NFTSMC control signal is designed in Eq. (28). Different from the previous study, we choose the sliding surface defined in Eq. (24) and propose a different design for the CLMR system to simplify while still ensuring the efficiency of trajectory tracking control:

$$\tau = \bar{\mathbf{M}}\left[-\mathbf{K}_1\mathbf{s} - \mathbf{K}_2 sign(\mathbf{s}) + \dot{\mathbf{w}}_r - \bar{\mathbf{D}} - \xi^{-1}\chi\right] \tag{28}$$

with $\mathbf{K_1} = diag[k_1, k_1]$, $\mathbf{K_2} = diag[k_2, k_2]$ are the diagonal matrices with $k_1 > 0, k_2 > 0$ are constants.

Theorem 1: Based on the analysis of the dynamic model in Eq. (21), combined with the NFTSMC control signal in Eq. (28) and the sliding surface in Eq. (24), the stability of the CLMR system is only guaranteed when the system operates under ideal conditions, i.e., without model errors and external interference.

Proof. We choose the Lyapunov function as follows:

$$V = V_{s_1} + V_{s_2} = \frac{1}{2}\mathbf{s}^T\mathbf{s} = \frac{1}{2}s_1^2 + \frac{1}{2}s_2^2 \tag{29}$$

Then, we derive Eq. (29) and combine it with Eq. (27):

$$\dot{V} = \mathbf{s}^T\dot{\mathbf{s}} = \mathbf{s}^T(\chi + \xi(\bar{\mathbf{M}}^{-1}\tau + \bar{\mathbf{D}} + \Delta\bar{\mathbf{F}} - \dot{\mathbf{w}}_r)) \tag{30}$$

Substituting the control signal designed in Eq. (28) into Eq. (30) gives:

$$\dot{V} = \mathbf{s}^T\dot{\mathbf{s}} = \mathbf{s}^T\xi(-\mathbf{K}_1\mathbf{s} - \mathbf{K}_2 sign(\mathbf{s}) + \Delta\bar{\mathbf{F}}) \tag{31}$$

Remark 2: When the CLMR system operates under ideal operating conditions, $\dot{V} = \mathbf{s}^T\xi(-\mathbf{K}_1\mathbf{s} - \mathbf{K}_2 sign(\mathbf{s})) \le 0$, the stability of the system is guaranteed according to Lyapunov theory. On the contrary, when $\dot{V} = \mathbf{s}^T\xi(-\mathbf{K}_1\mathbf{s} - \mathbf{K}_2 sign(\mathbf{s}) + \Delta\bar{\mathbf{F}})$, based on Assumption 1 in Eq. (20), the system is only stable when $k_2 \ge \varepsilon$ is selected to ensure $\dot{V} \le= \mathbf{s}^T\xi(-\mathbf{K}_1\mathbf{s}) \le 0$. However, when k_2 is increased too much, the system is prone to chattering. In addition, in practice, due to the existence of model errors and many disturbing factors affecting the CLMR, the choice of $k_2 \ge \varepsilon$ is difficult to achieve. Therefore, the NFTSMC designed in Eq. (28) does not ensure the stability of the CLMR system in the presence of unknown disturbances. The proof is complete.

3.2. Adaptive Nonsingular Fast Terminal Sliding Mode Control for CLMR system

Based on Theorem 1 and Remark 2, it is evident that NFTSMC alone does not ensure the stability of the CLMR system in the presence of unknown disturbances. To address this, Adaptive-NFTSMC is formulated by integrating NFTSMC with an online adaptive adjustment law, comprising the control-parameter update law in Eq. (33) and the disturbance upper-bound estimation law in Eq. (34). This formulation allows the system to better accommodate uncertainties and external disturbances. In this study, the Adaptive-NFTSMC control signal is designed as follows:

$$\tau = \bar{\mathbf{M}}\left[-\hat{\mathbf{K}}_1\mathbf{s} - (\hat{\mathbf{K}}_2 + \hat{\varepsilon}) sign(\mathbf{s}) + \dot{\mathbf{w}}_r - \bar{\mathbf{D}} - \xi^{-1}\chi\right] \tag{32}$$

The parameters in the Adaptive-NFTSMC control signal in Eq. (32) are designed with the adaptive online update law as follows:

$$\begin{aligned} \hat{\mathbf{K}}_1 &= diag[k_1|x_1|^{\lambda_1-1}, k_1|x_2|^{\lambda_1-1}] \\ \hat{\mathbf{K}}_2 &= diag[k_2|y_1|^{\lambda_2-1}, k_2|y_2|^{\lambda_2-1}] \end{aligned} \tag{33}$$

with $x_i = e_i$ if $|e_i| > \eta_1$, or $x_i = \eta_1$ if $|e_i| \le \eta_1$; $y_i = \dot{e}_i$ if $|\dot{e}_i| > \eta_2$, or $y_i = \eta_2$ if $|\dot{e}_i| \le \eta_2$. With $i = 1,2$; $\eta_1 > 0, \eta_2 > 0$ are constants; $\lambda_1 > 0$ and $\lambda_2 > 0$ are constants. Through practical testing, we recommend choosing parameters λ_1, λ_2 that satisfy $\lambda_1 \in [0.5, 1]$, $\lambda_2 \in [1, 1.5]$.

The upper bound of the noise is designed with the adaptive online update law as follows:

$$\dot{\hat{\varepsilon}} = \eta_3\mathbf{s}^T|e|^{\frac{a}{b}-1} sign(\mathbf{s}) \tag{34}$$

with $\eta_3 > 0$ is a constant.

Theorem 2: Based on the analysis of the dynamic model in Eq. (21), the Adaptive-NFTSMC control signal is designed in Eq. (32), the sliding surface in Eq. (24), together with the online adaptive update rules in Eq. (33) and Eq. (34), the stability of the CLMR system is guaranteed even when the system is affected by model errors and noise. Moreover, the adaptive parameters also include and converge to real values over time.

Proof. At this point, we choose the new Lyapunov function as follows:

$$V = V_{s_1} + V_{s_2} + V_{\varepsilon} = \frac{1}{2}\mathbf{s}^T\mathbf{s} + \frac{a}{b}\frac{\beta}{2\eta_3} - (\hat{\varepsilon} - \varepsilon)^2 \tag{35}$$

Derive Eq. (35), we have:

$$\dot{V} = \mathbf{s}^T\dot{\mathbf{s}} + \frac{a}{b}\frac{\beta}{\eta_3}\dot{\hat{\varepsilon}}(\hat{\varepsilon} - \varepsilon)^2 \tag{36}$$

Substituting Eq. (34) into Eq. (36), we have:

$$\dot{V} = \mathbf{s}^T\dot{\mathbf{s}} + \frac{a}{b}\beta\mathbf{s}^T|e|^{\frac{a}{b}-1} sign(\mathbf{s})(\hat{\varepsilon} - \varepsilon)^2 \tag{37}$$

Substituting Eq. (27) with the Adaptive-NFTSMC control signal in Eq. (32) into Eq. (37), we have:

$$\dot{V} = \mathbf{s}^T\xi(-\hat{\mathbf{K}}_1\mathbf{s} - \hat{\mathbf{K}}_2 sign(\mathbf{s})) + \mathbf{s}^T\xi(\Delta\bar{\mathbf{F}} - \varepsilon sign(\mathbf{s})) \tag{38}$$

From assumption 1 in Eq. (20) and Eq. (38), we have:

$$\dot{V} \le \mathbf{s}^T\xi(-\hat{\mathbf{K}}_1\mathbf{s} - \hat{\mathbf{K}}_2 sign(\mathbf{s})) = \sum_{i=1}^{2}\xi_i(-k_1|x_i|^{\lambda_1-1}s_i^2 - k_2|y_i|^{\lambda_2-1}|s_i|) \le 0 \tag{39}$$

with $\xi_1 > 0, \xi_2 > 0$ are the elements on the main diagonal of the diagonal matrix ξ_1.

Remark 3: From Eq. (39), the derivative of the Lyapunov function satisfies $\dot{V} \le 0$, indicating that the Lyapunov function V in Eq. (35) is bounded and non-increasing over time. This ensures that the system is asymptotically stable according to Lyapunov theory. Then, the sliding surface $s \to 0$ after a finite time (see convergence analysis in Section 3.3 below). Based on the adaptive law in Eq. (34), we have:

$$\dot{\hat{\varepsilon}} = \eta_3\mathbf{s}^T|e|^{\frac{a}{b}-1} sign(\mathbf{s})$$

When $s \to 0$, we deduce $\dot{\hat{\varepsilon}} \to 0$. This implies that $\hat{\varepsilon}$ will converge to a constant value. Similarly, the adaptive control coefficients in Eq. (33), including $\hat{\mathbf{K}}_1, \hat{\mathbf{K}}_2$, are functions that depend nonlinearly on the magnitude of the error e_i and the derivative of the error $\dot{e}_i$, together with the tuning parameters λ_1 and λ_2. When $\mathbf{e} \to 0$ and $\dot{\mathbf{e}} \to 0$, we have $x_i \to \eta_1$ and $y_i \to \eta_2$. With the parameter choices $\lambda_1 \in [0.5,1]$, $\lambda_2 \in [1,1.5]$ as suggested in Eq. (33), the exponential terms $|x_i|^{\lambda_1-1}, |y_i|^{\lambda_2-1}$ will tend to stabilize small values. Consequently, the components of $\hat{\mathbf{K}}_1, \hat{\mathbf{K}}_2$ converge to corresponding small stable values, completing the proof.

3.3. Finite-time convergence analysis of the CLMR's system

Based on Eq. (35) and Eq. (39), we can deduce:

$$\begin{cases} \dot{V}_{s_1} \le -\xi_1 k_1|x_1|^{\lambda_1-1}s_1^2 - \xi_1 k_2|y_1|^{\lambda_2-1}|s_1| \\ \dot{V}_{s_2} \le -\xi_2 k_1|x_2|^{\lambda_1-1}s_2^2 - \xi_2 k_2|y_2|^{\lambda_2-1}|s_2| \end{cases} \tag{40}$$

Based on Eq. (29), Eq. (40) is equivalent to:

$$\begin{cases} \dot{V}_{s_1} \le -2\xi_1 k_1|x_1|^{\lambda_1-1}V_{s_1} - \sqrt{2}\xi_1 k_2|y_1|^{\lambda_2-1}V_{s_1}^{1/2} \\ \dot{V}_{s_2} \le -2\xi_2 k_1|x_2|^{\lambda_1-1}V_{s_2} - \sqrt{2}\xi_2 k_2|y_2|^{\lambda_2-1}V_{s_2}^{1/2} \end{cases} \tag{41}$$

Here, to simplify the analysis process, we set the variables as follows:

$$\begin{aligned} &\rho_1 = 2\xi_1 k_1|x_1|^{\lambda_1-1}; \rho_2 = \sqrt{2}\xi_1 k_2|y_1|^{\lambda_2-1} \\ &\rho_3 = 2\xi_2 k_1|x_2|^{\lambda_1-1}; \rho_4 = \sqrt{2}\xi_2 k_2|y_2|^{\lambda_2-1} \end{aligned} \tag{42}$$

Equation. (41) becomes:

$$\begin{cases} \dot{V}_{s_1} = \dfrac{dV_{s_1}}{dt} \le -\rho_1 V_{s_1} - \rho_2 V_{s_1}^{1/2} \le 0 \\ \dot{V}_{s_2} = \dfrac{dV_{s_2}}{dt} \le -\rho_3 V_{s_2} - \rho_4 V_{s_2}^{1/2} \le 0 \end{cases} \tag{43}$$

From Eq. (43), we deduce:

$$\begin{cases} dt \le \dfrac{-dV_{s_1}}{\rho_1 V_{s_1} + \rho_2 V_{s_1}^{1/2}} \le \dfrac{-2dV_{s_1}^{1/2}}{\rho_1 V_{s_1}^{1/2} + \rho_2} \\ dt \le \dfrac{-dV_{s_2}}{\rho_3 V_{s_2} + \rho_4 V_{s_2}^{1/2}} \le \dfrac{-2dV_{s_2}^{1/2}}{\rho_3 V_{s_2}^{1/2} + \rho_4} \end{cases} \tag{44}$$

We suppose after some time, the CLMR system reaches the desired reference values. Then the Lyapunov auxiliary functions satisfy $V_{s_1(t_r)} = 0$ and $V_{s_2(t_r)} = 0$. Then Eq. (44) is equivalent to:

$$\begin{cases} \int_0^{t_r} dt \le \int_0^{t_r} \dfrac{-2dV_{s_1}^{1/2}}{\rho_1 V_{s_1}^{1/2} + \rho_2} = \dfrac{2}{\rho_1}\ln\left(\dfrac{\rho_1 V_{s_1(t=0)}^{1/2} + \rho_2}{\rho_2}\right) \\ \int_0^{t_r} dt \le \int_0^{t_r} \dfrac{-2dV_{s_2}^{1/2}}{\rho_3 V_{s_2}^{1/2} + \rho_4} = \dfrac{2}{\rho_3}\ln\left(\dfrac{\rho_3 V_{s_2(t=0)}^{1/2} + \rho_4}{\rho_4}\right) \end{cases} \tag{45}$$

Remark 4: From Eq. (45), according to Lyapunov stability theory, the CLMR system can converge to 0 quickly after a finite period of time. From there, it ensures that the velocity errors of the system also converge to 0, helping the system achieve the desired stable state. In which the time satisfies:

$$t_r \le \min\left\{\frac{2}{\rho_1}\ln\left(\frac{\rho_1 V_{s_1(t=0)}^{1/2} + \rho_2}{\rho_2}\right), \frac{2}{\rho_3}\ln\left(\frac{\rho_3 V_{s_2(t=0)}^{1/2} + \rho_4}{\rho_4}\right)\right\} \tag{46}$$

3.4. Chattering-free for CLMR system

Theorem 3: To minimize the chattering phenomenon for the CLMR system, the *sign*(**s**) function in the Adaptive-NFTSMC control signal in Eq. (32) is replaced by a soft sigmoid function, which helps smooth the control signal while still ensuring the stability and convergence in finite time of the system. Each component s_i in the vector **s** is replaced by the following expression:

$$\frac{s_i}{|s_i| + \sigma} \tag{47}$$

where $|s_i|$ is the absolute value of s_i in vector **s**, and $\sigma > 0$ is the parameter tuned to reduce chattering.

Proof. The Adaptive-NFTSMC control signal is rewritten as follows:

$$\tau = \bar{\mathbf{M}}\left[-\hat{\mathbf{K}}_1\mathbf{s} - (\hat{\mathbf{K}}_2 + \hat{\varepsilon})\frac{\mathbf{s}}{|\mathbf{s}| + \sigma} + \dot{\mathbf{w}}_r - \bar{\mathbf{D}} - \xi^{-1}\chi\right] \tag{48}$$

where:

$$\frac{\mathbf{s}}{|\mathbf{s}| + \sigma} = \left[\frac{s_1}{|s_1| + \sigma}, \frac{s_2}{|s_2| + \sigma}\right]^T \tag{49}$$

Due to the nature of the function $s_i / (|s_i| + \sigma)$ being bounded in the range of values (-1, 1). Then, Eq. (39) becomes:

$$\dot{V} \le \sum_{i=1}^{2}\xi_i\left(-k_1|x_i|^{\lambda_1 - 1}s_i^2 - k_2|y_i|^{\lambda_2 - 1}\frac{s_i^2}{|s_i| + \sigma}\right) \le 0 \tag{50}$$

with the values $i = 1, 2,$ we have:

$$\dot{V}_{s_i} \le \xi_i\left(-k_1|x_i|^{\lambda_1 - 1}s_i^2 - k_2|y_i|^{\lambda_2 - 1}\frac{s_i^2}{|s_i| + \sigma}\right) \tag{51}$$

The variables in Eq. (42) are reset as follows:

$$\rho_i = 2\xi_i\left(k_1|x_i|^{\lambda_1 - 1} + \frac{k_2|y_i|^{\lambda_2 - 1}}{|s_i| + \sigma}\right) \tag{52}$$

At this point, Eq. (43) becomes:

$$\dot{V}_{s_i} = \frac{dV_{s_i}}{dt} \le -\rho_i V_{s_i} \le 0 \tag{53}$$

From Eq. (53) we deduce:

$$dt \le \frac{-dV_{s_i}}{\rho_i V_{s_i}} \tag{54}$$

Suppose, after some time, the CLMR system reaches the desired reference values. Then the Lyapunov auxiliary functions satisfy $V_{s_i(t_r)} = 0.$ At this time, the Eq. (54) is equivalent to:

$$\int_0^{t_r} dt \le \int_0^{t_r} \frac{-dV_{s_i}}{\rho_i V_{s_i}} = \frac{1}{\rho_i}\ln(\rho_i V_{s_i(t=0)}) \tag{55}$$

From Eq. (55) we get:

$$t_r \le \min\left\{\frac{1}{\rho_1}\ln\left(\rho_1 V_{s_1(t=0)}\right), \frac{1}{\rho_2}\ln\left(\rho_2 V_{s_2(t=0)}\right)\right\} \tag{56}$$

Remark 5: Based on Eq. (50) and Eq. (56), the system is stable according to Lyapunov theory and converges in finite time. The proof is complete.

Table 1. Parameter sensitivity analysis ($\uparrow$: increase, $\downarrow$: decrease).

Parameters	Convergence time	Overshoot	Steady-state tracking error	Control signal
$\alpha^{\uparrow}$	$\downarrow$	$\downarrow$	$\downarrow$	$\uparrow$
$\beta^{\uparrow}$	$\uparrow$	$\downarrow$	$\uparrow$	$\uparrow$
$k_1^{\uparrow}$	$\downarrow$	$\downarrow$	$\downarrow$	$\uparrow$
$k_2^{\uparrow}$	$\downarrow$	$\downarrow$	$\downarrow$	$\uparrow$
$\sigma^{\uparrow}$	$\uparrow$	$\uparrow$	$\uparrow$	$\downarrow$

Table 2. Parameters in the mathematical model of the CLMR system.

Parameters	Value
m_c	2.3 (kg)
m_w	0.23 (kg)
L	0.2 (m)
l	0.05 (m)
b	0.1 (m)
I_{bzz}	0.015 ($kg.m^2$)
I_{wzz}	0.002 ($kg.m^2$)

From the detailed analysis mentioned above, the input torque vector of the robot with Adaptive-NFTSMC is designed by:

$$\mathbf{T} = \bar{\mathbf{B}}^{-1}(\mathbf{q})\tau = \bar{\mathbf{B}}^{-1}(\mathbf{q})\left(\bar{\mathbf{M}}\left[-\hat{\mathbf{K}}_1\mathbf{s} - (\hat{\mathbf{K}}_2 + \hat{\varepsilon})\frac{\mathbf{s}}{|\mathbf{s}| + \sigma} + \dot{\mathbf{w}}_r - \bar{\mathbf{D}} - \xi^{-1}\chi\right]\right) \tag{57}$$

The performance of Adaptive-NFTSMC in Eq. (57) depends mainly on a set of key parameters including α, β, k_1, k_2,σ. To assess the influence of each parameter, a sensitivity analysis is performed in which the value of each parameter is varied individually while the others are kept fixed, and the corresponding system responses are recorded. The results, summarized in Table 1, illustrate the role and impact of each parameter on the control performance. These findings provide an important basis for selecting the control parameters.

The parameter sensitivity analysis indicates that increasing k_1 and k_2 shortens the convergence time and reduces both overshoot and steady-state error; however, it also increases the control signal amplitude. In contrast, increasing σ lengthens the convergence time, increases overshoot and steady-state error, but reduces the control signal amplitude. For the two parameters α and β, the results show that they have opposite effects. Specifically, increasing the values of α reduces the convergence time and steady-state error, but increases the control signal amplitude. Meanwhile, increasing the values of β lengthens the convergence time and increases the steady-state error, although it also increases the control signal amplitude.

4. Simulation Results and Analysis

In this study, MATLAB/Simulink is used to evaluate the effectiveness of the Adaptive-NFTSMC control algorithm. The simulations employ the ODE4 (Runge–Kutta) solver in fixed-step mode with an automatic step size and a total simulation time of 25 s. Regarding the hardware configuration, consistent with the description in [36], all simulations are executed on a computer equipped with an Intel® Core™ i7-11850H processor (8 cores, 2.50 GHz) and 32 GB of RAM, providing sufficient computational capacity and memory to run the control algorithm and ensure stable operation in MATLAB R2023b. The Adaptive-NFTSMC control method is compared with other methods such as SMC [6] and ANFTSMC [7]. The simulation scenarios are carefully set up in two operating conditions of the CLMR system, including ideal conditions and conditions affected by disturbances. To ensure objectivity when simulating, evaluating, and comparing the Adaptive-NFTSMC, SMC, and NFTSMC control methods, the parameters in the mathematical model of the CLMR system are taken from the study [24] with the parameters defined in Table 2.

In this simulation case, the CLMR is required to follow a circular trajectory set by the following equation:

$$\begin{cases} x_r = 1.5\sin(0.4t) \\ y_r = -1.5\cos(0.4t) + 1.5 \end{cases} \tag{58}$$

From Eq. (53), the equations of reference linear velocity υ_r (m/s), and reference steering velocity ω_r (rad/s) are calculated as follows:

$$\begin{cases} \upsilon_r = \sqrt{\dot{x}_r^2 + \dot{y}_r^2} \\ \omega_r = \dot{\delta}_r \end{cases} \tag{59}$$

Moreover, reference yaw angle θ_r (rad), reference yaw angular velocity $\dot{\theta}_r$ (rad/s), and reference steering angle δ_r (rad), are calculated as follows:

$$\begin{cases} \theta_r = a\tan 2\{\dot{y}_r, \dot{x}_r\} \\ \dot{\theta}_r = \dfrac{\ddot{y}_r\dot{x}_r - \dot{x}_r\dot{y}_r}{\upsilon_r^2} \\ \delta_r = a\tan 2\{\dot{\theta}_r, L, \upsilon_r\} \end{cases} \tag{60}$$

Table 3. Parameters of the Adaptive-NFTSMC controller.

Parameters	Value	Parameters	Value
α	0.8	k_1	50
φ	2.6	k_2	30
β	1	λ_1	0.7
a / b	9 / 7	λ_2	1.1
η_1	3×10^{-5}	η_3	0.001
η_2	3×10^{-5}	σ	0.8

In addition, the boundary conditions for the steering angle in the simulation are given as follows: $|\delta|\le 45^{\circ}$. Suppose, in this case, the CLMR system is moving with the initial linear velocity and steering velocity of $v(0)=1$ (m/s) and $\omega(0)=1$ (rad/s), respectively, at the initial position $(x,y)=(0,0)$, suddenly, the reference trajectory must be changed to the circular trajectory established in Eq. (58) with the new linear velocity, new steering velocity, and other reference values established in Eq. (59).

The use of circular trajectories in simulation is particularly practical, as it not only evaluates the trajectory-tracking capability of the Adaptive-NFTSMC algorithm for a common trajectory type encountered in tasks such as navigating around obstacles or moving within a specified area, but also directly supports testing of the Adaptive-NFTSMC navigation stack. In this simulation scenario, the TEB provides the robot with linear velocity commands and the reference steering velocity in Eq. (59) to track the desired circular trajectory defined in Eq. (58). This trajectory facilitates straightforward determination of reference values and enables accurate simulation of the expected motion behavior, thereby providing a reliable basis for assessing the algorithm's performance. In contrast, other trajectory types often require more complex calculations. In real environments, trajectories may occur unpredictably; according to [15], the robot's objective is to rapidly and accurately achieve the reference velocity values generated by the TEB, thereby maintaining adherence to the local reference trajectory and the intended motion behavior.

Based on the careful analysis of the sensitivity of the control parameters in Table 1, combined with the recommendations in [33] for selecting $1<a/b<2$ and $\varphi>a/b$, and by using the trial-and-error method to select $\lambda_1\in[0.5,1]$, $\lambda_2\in[1,1.5]$, and $\eta_1,\eta_2,\eta_3>0$. The parameters of the Adaptive-NFTSMC controller are selected and shown in Table 3.

To comprehensively assess the performance of the Adaptive-NFTSMC algorithm on the CLMR system, simulation experiments were conducted and analyzed under two distinct scenarios. The first scenario represents an ideal operating condition in which the system is entirely free from disturbances, meaning that the CLMR operates under perfect conditions. In contrast, the second scenario serves as a rigorous test, where the system operates under synthetic disturbances, including both random and fixed-amplitude signals, directly applied to the input torque to replicate real-world challenges. In both scenarios, the Adaptive-NFTSMC algorithm is directly compared with two other control methods: SMC and ANFTSMC. In the presented plots, the reference signals are shown as black dashed lines, Adaptive-NFTSMC as red dashed lines, ANFTSMC as green dashed lines, and SMC as blue dashed lines. The details of each scenario are provided in Sections 4.1 and 4.2, respectively.

4.1. CLMR system operates under ideal conditions

In this simulation scenario, the CLMR system operates under ideal conditions, entirely free from interference. Figure 6 presents the tracking performance of the CLMR using the Adaptive-NFTSMC, ANFTSMC, and SMC control algorithms. All three methods successfully track the predefined circular trajectory, although the deviation from the reference signal differs noticeably. The Adaptive-NFTSMC algorithm produces tracking results closely aligned with the reference trajectory, demonstrating high-precision control capability. The ANFTSMC method also achieves good tracking performance but exhibits a small deviation, whereas the SMC method shows a larger deviation, particularly in the magnified area on the right side of Fig. 6. These results indicate that Adaptive-NFTSMC not only maintains superior accuracy but also addresses the limitations of traditional control approaches, such as SMC, under ideal operating conditions. Theoretically, Adaptive-NFTSMC combines NFTSMC with the sliding surface defined in Eq. (24) and the adaptive laws in Eqs. (33) and (34), offering several advantages, including strong trajectory-tracking ability, fast response, singularity avoidance, and rapid convergence to the equilibrium state. Furthermore, the online parameter adjustment mechanism and disturbance upper-bound estimation enable the system to maintain stability without requiring prior assumptions that uncertainties and disturbances must be suppressed, which is a common constraint in conventional NFTSMC controllers.

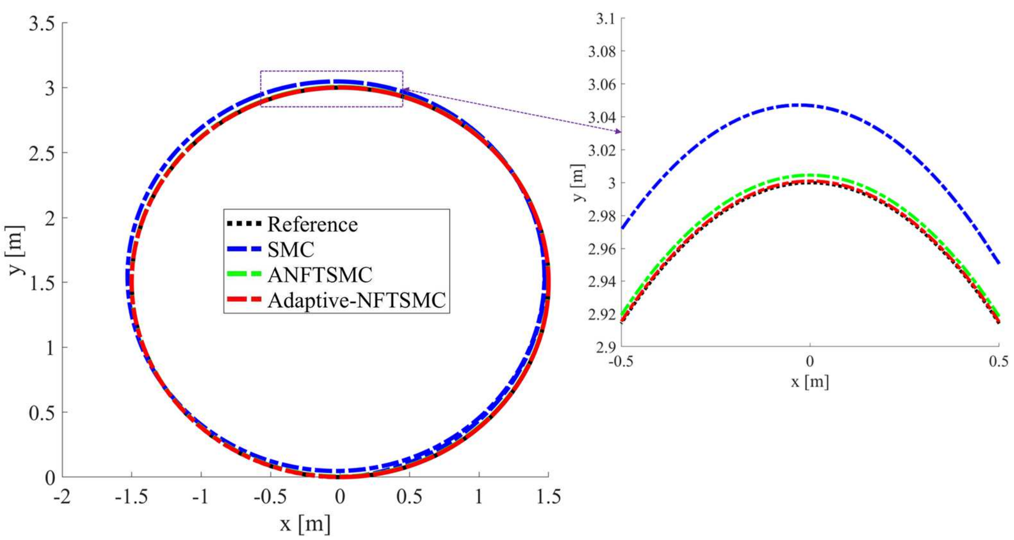

Fig. 6. Tracking trajectory of the CLMR under ideal operating conditions.

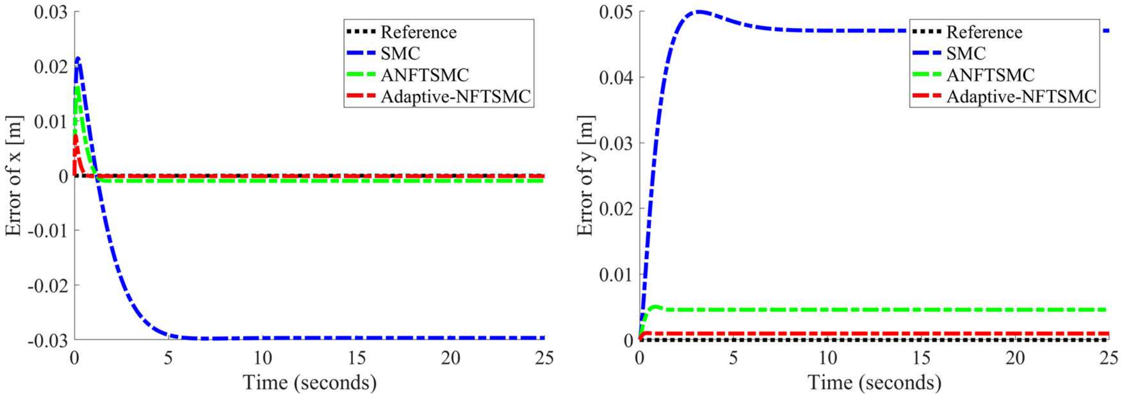

Fig. 7. Tracking Error of the CLMR under ideal operating conditions.

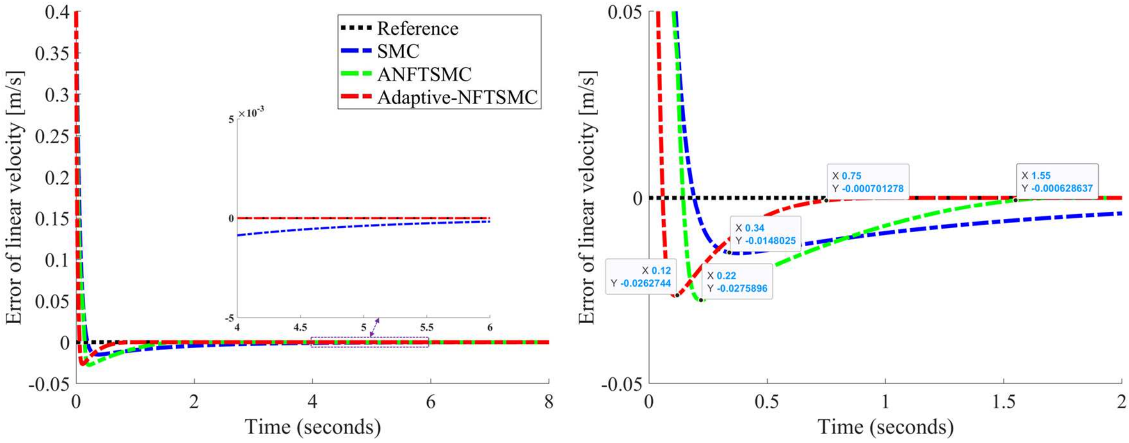

Fig. 8. Linear velocity error for the CLMR system under ideal operating conditions.

To further evaluate the trajectory-tracking capability, Fig. 7 illustrates the tracking errors along the x-axis and y-axis of the CLMR under ideal operating conditions. The results indicate that all three algorithms — Adaptive-NFTSMC, ANFTSMC, and SMC — exhibit overshoot in the initial stage, with clearly different amplitudes. Adaptive-NFTSMC achieves the smallest overshoot, approximately 0.007 m on the x-axis and 0.001 m on the y-axis, compared to ANFTSMC at approximately 0.016 m (x-axis) and 0.005 m (y-axis), and SMC with the largest overshoot at approximately 0.021 m (x-axis) and 0.050 m (y-axis). Following this stage, Adaptive-NFTSMC eliminates errors rapidly, maintaining very small values in steady state with a convergence time of about 0.75 s on both axes, and steady-state errors approaching zero — more specifically, 0.0009 m on the y-axis. In contrast, ANFTSMC converges more slowly, with a convergence time of about 1.55 s, steady-state error of approximately 0.0008 m on the x-axis, and 0.0046 m on the y-axis. The SMC method performs worst among the three, with the slowest convergence time of up to about 6 s, and steady-state errors of approximately 0.03 m (x-axis) and 0.047 m (y-axis). Overall, Adaptive-NFTSMC not only achieves the smallest transitional error but also ensures the fastest convergence rate and near-complete elimination of steady-state error, confirming its superior performance compared to ANFTSMC and SMC.

To evaluate the velocity-tracking performance of the control algorithms, Figs. 8 and 9 present the linear and steering velocity errors of the CLMR system under ideal operating conditions. The results indicate that all three controllers—SMC, ANFTSMC, and Adaptive-NFTSMC—drive the velocity error close to zero within a short period. Regarding overshoot, both ANFTSMC and Adaptive-NFTSMC exhibit small values of approximately 0.03 m/s for both linear and steering velocities. In contrast, SMC achieves a smaller overshoot in linear velocity (approximately 0.015 m/s) but a larger overshoot in steering velocity (approximately 0.04 m/s). As shown clearly in Figs. 8 and 9, Adaptive-NFTSMC produces the smallest velocity deviations among the three methods, with asymptotic tracking errors approaching zero and a fast convergence time of about 0.75 s, consistent with the trajectory convergence time in Fig. 7. This advantage demonstrates Adaptive-NFTSMC's capability to reduce oscillations and improve control quality for both linear and steering velocities. For SMC and ANFTSMC, the convergence times for both velocity components are approximately 6 s and 1.55 s, respectively. Thus, Adaptive-NFTSMC achieves both the smallest error and the fastest convergence, followed by ANFTSMC, with SMC being the slowest. These results also explain why, in Fig. 7, Adaptive-NFTSMC tracks the trajectory more accurately than the other two methods. Maintaining very small deviations in both linear and steering velocities helps the robot's dynamic state closely follow the reference values, limit the accumulation of errors over time, and prevent trajectory drift. In other words, precise velocity control is a key factor in ensuring accurate trajectory tracking.

The detailed analysis above shows that under ideal operating conditions, the Adaptive-NFTSMC algorithm outperforms ANFTSMC and SMC. To quantify this performance more clearly, Fig. 10 presents a comparison of Root Mean Square Error (RMSE) for four quantities, including linear velocity, steering velocity, x-axis trajectory, and y-axis trajectory. RMSE is calculated by the formula:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(e_i)^2} \tag{61}$$

with e_i is the error between the actual value and the reference value at the sample i and N is the total number of samples.

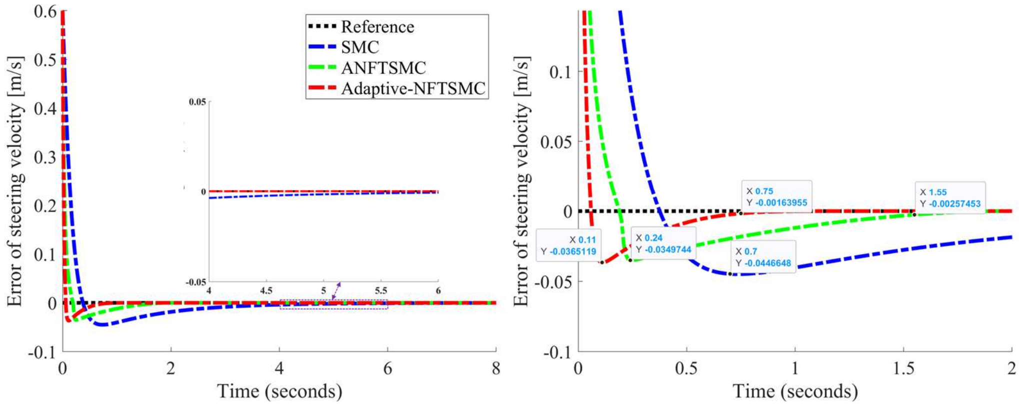

Fig. 9. Steering velocity error for the CLMR system under ideal operating conditions.

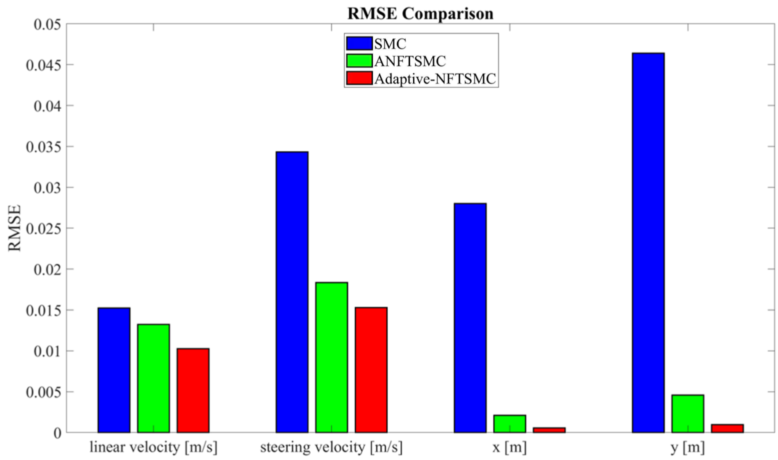

Fig. 10. RMSE for the CLMR system under ideal operating conditions.

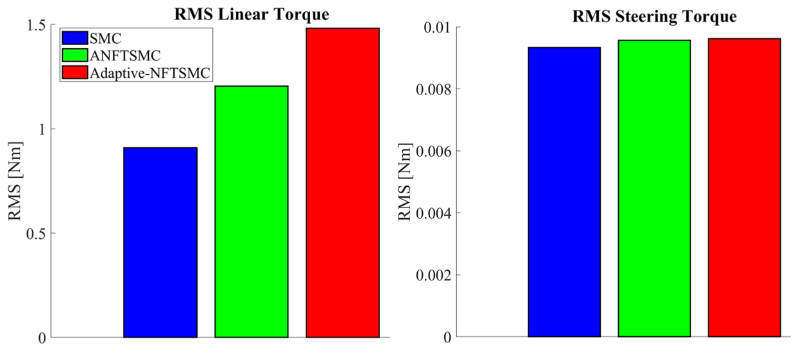

Fig. 11. RMS for the CLMR system's input torque under ideal operating conditions.

The results in Fig. 10 show that Adaptive-NFTSMC consistently achieves the smallest RMSE across all four criteria, followed by ANFTSMC and, lastly, SMC. Specifically, the linear velocity RMSE of Adaptive-NFTSMC is approximately 0.01 m/s, compared to 0.013 m/s for ANFTSMC and 0.015 m/s for SMC. This corresponds to a reduction of about 23% relative to ANFTSMC and about 33% relative to SMC. For steering velocity, the RMSE of Adaptive-NFTSMC is 0.015 m/s, while ANFTSMC and SMC record 0.018 m/s and 0.034 m/s, respectively—equivalent to reductions of approximately 16.7% and 55.9%. Notably, the RMSE values for the x-axis and y-axis trajectories of Adaptive-NFTSMC are nearly asymptotically zero and significantly smaller than those of ANFTSMC and SMC, with reductions exceeding 90% compared to SMC. For the x-axis, the RMSE of Adaptive-NFTSMC is only 0.0005 m, which is 75% lower than ANFTSMC (0.002 m) and 98.2% lower than SMC (0.028 m). For the y-axis, the RMSE is 0.0009 m, representing reductions of 80% compared to ANFTSMC (0.0045 m) and 98% compared to SMC (0.046 m). These results reinforce the conclusion that maintaining extremely small velocity deviations not only improves instantaneous kinematic performance but also substantially reduces cumulative trajectory error, ensuring that the robot follows the reference trajectory with high stability and accuracy.

To objectively evaluate energy consumption, Fig. 11 presents the root mean square (RMS) value of the control torque, calculated using the following formula:

$$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(T_i)^2} \tag{62}$$

with T_i is the input torque at the sample i and N is the total number of samples.

Table 4. Control performance under ideal operating conditions.

Criteria	Adaptive-NFTSMC	ANFTSMC	SMC
Convergence Time (s)	**0.75**	1.55	6
X-axis RMSE (m)	**0.0005**	0.002	0.028
Y-axis RMSE (m)	**0.0009**	0.0045	0.046
Linear velocity RMSE (m/s)	**0.01**	0.013	0.015
Steering velocity RMSE (m/s)	**0.015**	0.015	0.034
Linear torque RMS (Nm)	1.48	1.2	**0.9**
Steering torque RMS (Nm)	0.0096	0.0095	**0.0093**

The results in Fig. 11 indicate that the linear torque RMS of Adaptive-NFTSMC is approximately 1.48 Nm, which is about 21% higher than ANFTSMC (1.20 Nm) and about 64% higher than SMC (0.90 Nm). This implies that the torque requirement is greater, leading to higher energy consumption. However, this trade-off is considered reasonable, as it delivers superior velocity- and trajectory-tracking performance. In contrast, the steering torque RMS values for all three methods are nearly equivalent. Moreover, to balance energy consumption and control quality, the control parameters can be adjusted as shown in Table 3. For clarity, Table 4 provides a summary of the performance of Adaptive-NFTSMC, ANFTSMC, and SMC in this simulation scenario.

4.2. CLMR system operates under the influence of noise

In practical applications, CLMR systems are frequently subjected to various unpredictable external disturbances arising from factors such as terrain irregularities, wheel friction, variable payloads, and environmental conditions. In this test scenario, an extreme case is constructed to evaluate the disturbance tolerance of the Adaptive-NFTSMC algorithm. Specifically, the system is subjected to synthetic disturbances consisting of random-amplitude noise ranging from –0.4 Nm to 0.4 Nm and fixed-amplitude noise equal to 40% of the input control torque. Both types of disturbances are directly applied to the control torque signal to simulate severe operating conditions. Figure 12 depicts the unknown external disturbances introduced into the system, where the linear disturbance torque is represented by the orange dashed line and the steering disturbance torque by the purple dashed line. It can be observed that the amplitude of the linear disturbance fluctuates considerably more than that of the steering disturbance, reflecting the greater sensitivity of the linear velocity channel to external mechanical and environmental influences. In contrast, the steering torque is generally smaller, as it is primarily affected by the rotational inertia and frictional resistance of the steering mechanism, which are substantially lower than the resistive forces acting on the linear velocity control drive. As in the first scenario, the Adaptive-NFTSMC algorithm is comprehensively compared with ANFTSMC and SMC.

The tracking results under disturbance conditions are presented in Fig. 13. It can be observed that, although all three controllers maintain a trajectory shape close to the reference path, their tracking deviations differ noticeably. The SMC controller exhibits the largest tracking error, particularly in curved sections, due to the pronounced influence of external disturbances. The ANFTSMC method significantly improves accuracy but still shows a measurable deviation from the reference trajectory. In contrast, the Adaptive-NFTSMC algorithm demonstrates superior disturbance-compensation capability, enabling the trajectory to almost coincide with the reference path, especially in regions requiring continuous directional adjustments. This observation is consistent with the theoretical framework: traditional SMC not only struggles to meet the requirement for fast response but also depends on predefining the disturbance bound to ensure asymptotic stability based on Lyapunov theory—an assumption that is difficult to satisfy in practice. Adaptive-NFTSMC addresses this by integrating NFTSMC with adaptive tuning laws, including the control-parameter update law in Eq. (33), which adjusts the parameter magnitude in real time based on the error and its derivative, and the disturbance upper-bound estimation law in Eq. (34), which compensates for the effects of unknown disturbances without prior knowledge of their characteristics. These two mechanisms have been shown to ensure stability and finite-time convergence when the disturbance amplitude is bounded. As a result, Adaptive-NFTSMC maintains accurate trajectory tracking even under the combined effects of fixed-amplitude and random disturbances—a level of robustness not achieved by ANFTSMC or SMC.

To further evaluate the tracking performance, Fig. 14 presents the tracking errors along the x- and y-axes of the CLMR under operating conditions affected by external noise. In terms of tracking error amplitude, the SMC controller produces the largest deviations, with maximum values of approximately 0.3 m on the x-axis and 0.22 m on the y-axis, accompanied by pronounced fluctuations throughout the process. The ANFTSMC method significantly reduces the error amplitude but still exhibits maximum values of about 0.064 m on the x-axis and 0.08 m on the y-axis. In contrast, the Adaptive-NFTSMC algorithm maintains very small errors, almost completely eliminating the steady-state error on both axes and converging rapidly to the reference values. Specifically, Adaptive-NFTSMC achieves maximum steady-state error amplitudes of 0.006 m on the x-axis and 0.004 m on the y-axis.

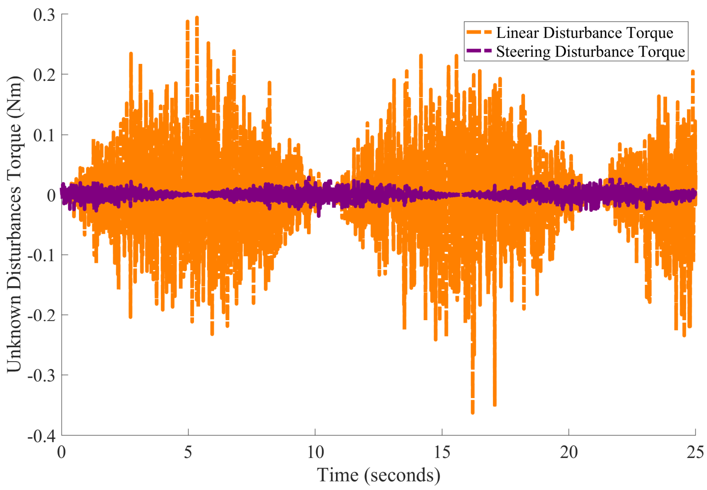

Fig. 12. Unknown external disturbances affecting CLMR.

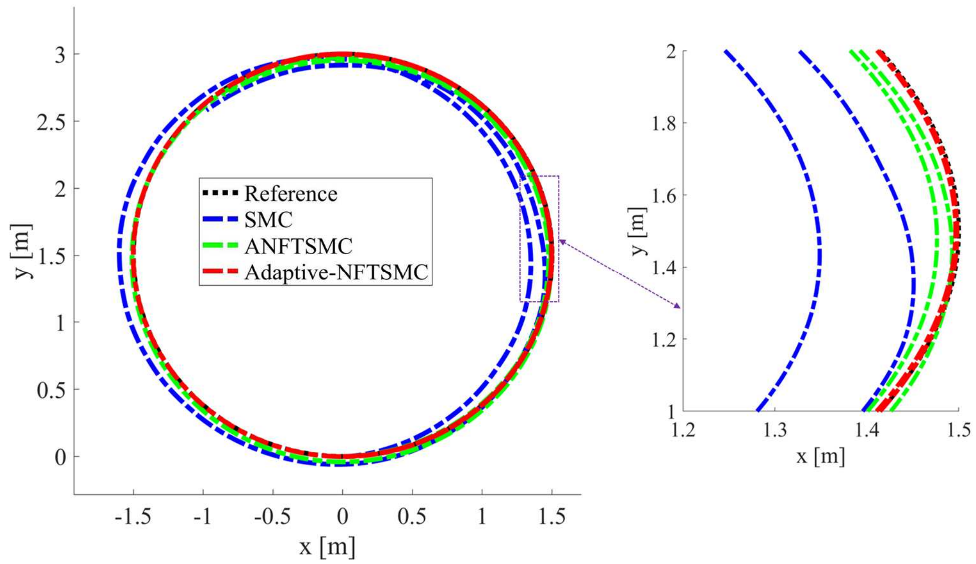

Fig. 13. CLMR's trajectories under disturbance

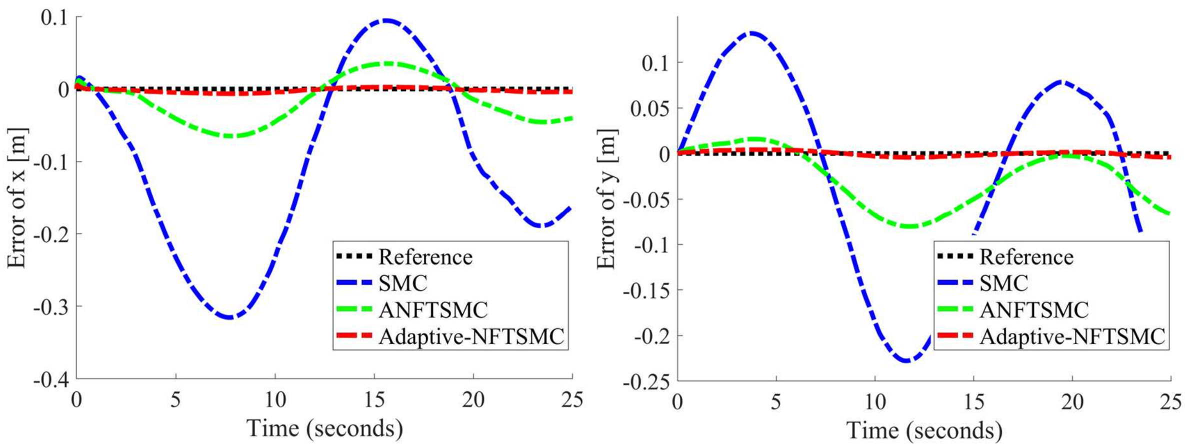

Fig. 14. CLMR's tracking error under disturbance.

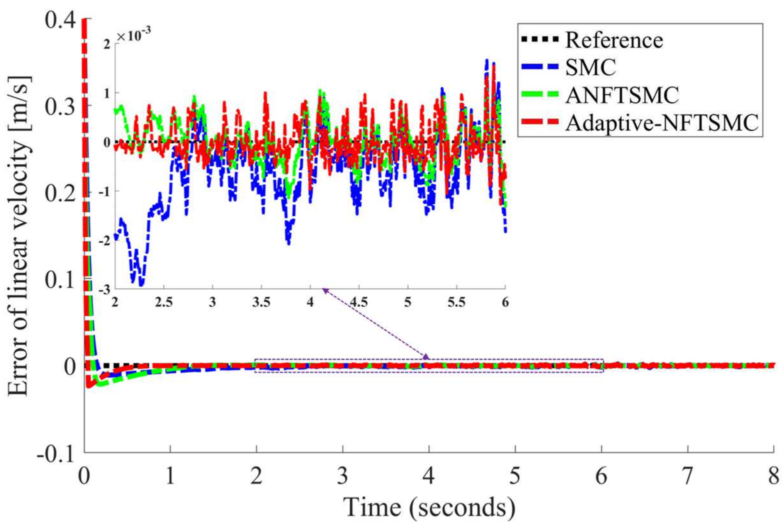

Fig. 15. CLMR's linear velocity error under disturbance.

Regarding convergence time, SMC reaches stability after approximately 7.5 s, while Adaptive-NFTSMC stabilizes after about 0.75 s and ANFTSMC after about 2.6 s for both axes. These results highlight the superior noise-compensation capability of Adaptive-NFTSMC, which is consistent with the theoretical foundation, as the algorithm incorporates an adaptive mechanism that enables real-time parameter estimation and adjustment, ensuring stability and high accuracy even when the noise characteristics are unknown in advance.

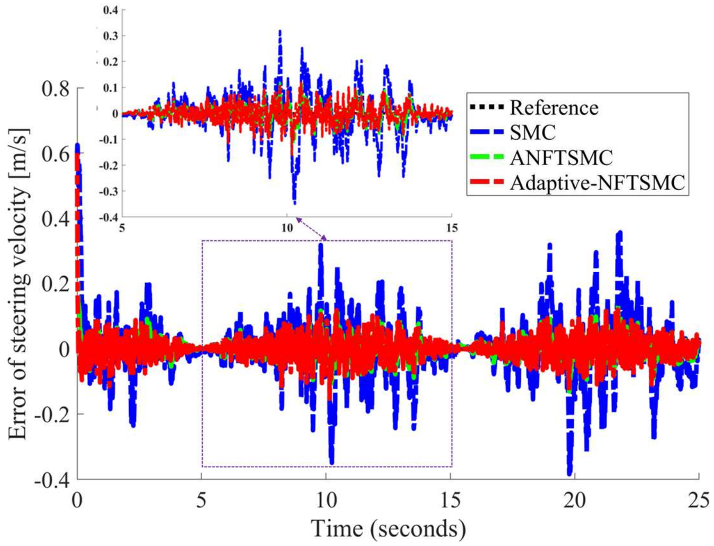

Fig. 16. CLMR's steering velocity error under disturbance.

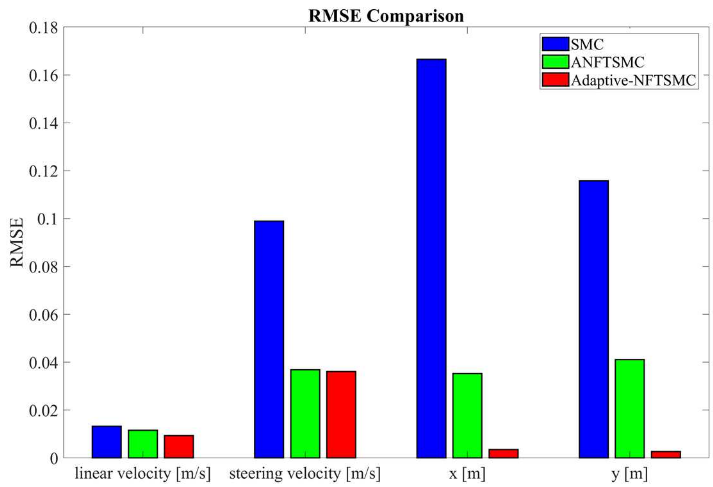

Fig. 17. CLMR's RMSE under disturbance.

Figures 15 and 16 provide detailed illustrations of the linear velocity and steering velocity errors of the CLMR system under the influence of external disturbances. The results show that the Adaptive-NFTSMC controller outperforms both SMC and ANFTSMC in controlling linear and steering velocities when disturbances are present. In the transient stage, Adaptive-NFTSMC and ANFTSMC rapidly reduce the error from the initial value to near zero with small overshoot, whereas SMC exhibits stronger oscillations. Once the system reaches steady state, both Adaptive-NFTSMC and ANFTSMC maintain the error at a very small level with narrow-range oscillations, with Adaptive-NFTSMC keeping the values closer to zero due to its adaptive parameter adjustment mechanism. During periods of strong disturbance, SMC displays large-amplitude fluctuations, while Adaptive-NFTSMC demonstrates superior noise-compensation capability, maintaining stable deviations. Specifically, for linear velocity, Adaptive-NFTSMC achieves deviations close to zero, as shown in Fig. 15. For steering velocity, Adaptive-NFTSMC achieves very small fluctuations, with a maximum amplitude of only 0.1 m/s, as shown in Fig. 16. Overall, Adaptive-NFTSMC not only improves tracking accuracy but also enhances disturbance rejection and system stability, enabling fast response and maintaining high performance in dynamic operating environments.

To quantitatively assess the performance of the controllers, Fig. 17 compares the RMSE values for four parameters—linear velocity, steering velocity, x-axis trajectory, and y-axis trajectory—when the CLMR operates under external disturbances. The results show that SMC has the highest RMSE across all parameters, indicating limited capability in disturbance compensation and trajectory maintenance. ANFTSMC achieves notable improvements over SMC, reducing RMSE in all metrics. However, Adaptive-NFTSMC continues to demonstrate clear superiority, achieving the lowest RMSE in every category. Specifically, the linear velocity RMSE of Adaptive-NFTSMC is approximately 0.009 m/s, compared to 0.011 m/s for ANFTSMC and 0.013 m/s for SMC, representing reductions of about 18% relative to ANFTSMC and 30% relative to SMC. For steering velocity, the RMSE of Adaptive-NFTSMC is 0.0361 m/s, compared to 0.0368 m/s for ANFTSMC and 0.098 m/s for SMC, corresponding to reductions of about 2% and 63%, respectively. For the x-axis trajectory, the RMSE of Adaptive-NFTSMC is only 0.0035 m, which is 90% lower than ANFTSMC (0.035 m) and 98% lower than SMC (0.166 m). For the y-axis trajectory, the RMSE of Adaptive-NFTSMC is 0.0027 m, representing reductions of 93.4% compared to ANFTSMC (0.041 m) and 97.6% compared to SMC (0.115 m). These quantitative results confirm that the adaptive adjustment mechanism of Adaptive-NFTSMC not only enhances velocity error compensation but also significantly improves trajectory accuracy in highly dynamic operating environments.

Figure 18 presents the RMS values of the linear control torque and steering control torque, which serve as indicators of the CLMR system's energy consumption under external disturbances. For linear torque, SMC records the lowest RMS value at approximately 0.76 Nm, followed by ANFTSMC at about 1.05 Nm, while Adaptive-NFTSMC has the highest value at approximately 1.27 Nm.

Table 5. Control performance under disturbance conditions.

Criteria	Adaptive-NFTSMC	ANFTSMC	SMC
Convergence Time (s)	**0.75**	2.6	7.5
X-axis RMSE (m)	**0.0035**	0.035	0.166
Y-axis RMSE (m)	**0.0027**	0.041	0.115
Linear velocity RMSE (m/s)	**0.09**	0.011	0.013
Steering velocity RMSE (m/s)	**0.0361**	0.0368	0.098
Linear torque RMS (Nm)	1.27	1.05	**0.76**
Steering torque RMS (Nm)	0.0087	0.0069	**0.0064**

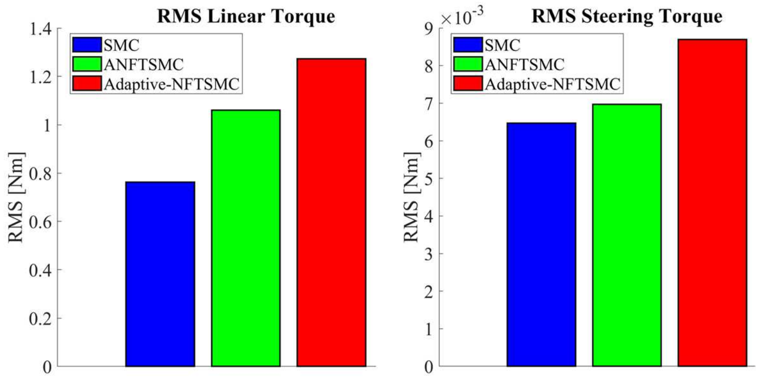

Fig. 18. CLMR's RMS for input torque under disturbance.

For steering torque, the RMS differences are smaller, with SMC at approximately 0.0064 Nm, ANFTSMC at 0.0069 Nm, and Adaptive-NFTSMC at 0.0087 Nm. These results indicate that, although Adaptive-NFTSMC requires a higher control torque, this increase is justified by its superior velocity- and trajectory-tracking accuracy, as demonstrated in Figs. 13–17. Moreover, with the rapid advancement of modern battery technology, energy storage and delivery capabilities have improved significantly, making this increase in energy consumption entirely acceptable in practice. Therefore, the higher energy demand of Adaptive-NFTSMC can be considered a reasonable trade-off for its enhanced tracking performance and robustness against disturbances. A summary of the performance of Adaptive-NFTSMC, ANFTSMC, and SMC in this challenging and realistic simulation scenario is provided in Table 5.

5. Conclusion and Future Works

This study developed a novel adaptive control method, Adaptive-NFTSMC, for a Car-Like Mobile Robot to improve the trajectory tracking ability of the system. The method is built on the Nonsingular Fast Terminal Sliding Mode Control (NFTSMC) technique, combined with flexible online adaptation mechanisms. Adaptive-NFTSMC helps the system cope effectively with uncertainties and disturbances, achieve high accuracy and stability, avoid chattering, and converge in finite time. Another advantage of Adaptive-NFTSMC is its simple structure, which makes it easy to integrate into autonomous navigation stacks, expanding the potential for practical applications. The stability of the closed-loop system has been proven by Lyapunov theory and verified through simulations on MATLAB/Simulink. Simulation results show that Adaptive-NFTSMC outperforms SMC and other adaptive NFTSMC techniques (such as ANFTSMC), especially under the condition of system noise and uncertainty. In the future, we will conduct real-world experiments to integrate Adaptive-NFTSMC into autonomous navigation systems.

Author Contributions

S.-T. Dang and X.-M. Dinh planned the study, developed the control algorithms, and proposed the simulation scenarios. S.-T. Dang and X.-M. performed the detailed quantitative simulations and wrote the manuscript. T.-A. Than Ngoc and V.-T. Nguyen jointly built the mathematical model and verified the correctness of the theory. D.-T. Kim and X.-H. Le jointly provided comments to improve the manuscript. The manuscript was completed with the contributions of all authors. All authors discussed the results, reviewed, and approved the final manuscript.

Acknowledgments

To complete this research, we would like to thank the Faculty of Mechanical Engineering and Mechatronics Phenikaa University, Vietnam; International School, Vietnam National University; and HTI Group.

Conflict of Interest

The authors declared no potential conflicts of interest concerning the research, authorship, and publication of this article.

Funding

This research is funded by International School, Vietnam National University, Hanoi (VNU-IS) under project number CS.2024-04.

Data Availability Statements

The datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request.

Nomenclature

x, y	Position of the rear-axle midpoint [m]	θ	Heading angle [rad]
δ	Front-wheel steering angle [rad]	v	Linear velocity [m/s]
ω	Steering velocity [rad/s]	R	Instantaneous turning radius [m]
L	Wheelbase (front-rear axle distance) [m]	l	Half rear-wheel track width [m]
I_w	Wheel's Moment of inertia [kg.m^2]	I_{bzz}	Robot's moments of inertia [kg.m^2]
I_{wzz}	Z axis's moments of inertia [kg.m^2]	m_c	Robot chassis mass [kg]
m_w	Mass of each wheel [kg]	b	Distance from CG to rear axle [m]

References

[1] Xie, Z., Ramezani, M., Levinson, D., Reduced-Scale Mobile Robots for Autonomous Driving Research, *IEEE Transactions on Intelligent Transportation Systems*, 25(11), 2024, 15367-15387.
[2] Karthik, R., Menaka, R., Kishore, P., Aswin, R., Vikram, C., Dual mode PID controller for path planning of encoder-less mobile robots in warehouse environment, *IEEE Access*, 12, 2024, 21634-21646.
[3] Cáceres Flórez, C. A., Rosário, J. M., Amaya, D., Control structure for a car-like robot using artificial neural networks and genetic algorithms, *Neural Computing and Applications*, 32(20), 2020, 15771-15784.
[4] Numbi, J., Zioui, N., Tadjine, M., Quantum Particle Swarm Optimisation Proportional–Derivative Control for Trajectory Tracking of a Car-like Mobile Robot, *Electronics*, 14(5), 2025, 832.
[5] Liu, S., Lin, Z., Huang, W., Yan, B., Adaptive sliding mode attitude control of two-wheeled robots for planetary auxiliary: From theory to applications, *Aerospace Science and Technology*, 151, 2024, 109332.
[6] Medina, L., Guerra, G., Herrera, M., Guevara, L., Camacho, O., Trajectory tracking for non-holonomic mobile robots: A comparison of sliding mode control approaches, *Results in Engineering*, 22, 2024, 102105.
[7] Hussein, M., Zhang, Y., Liu, Z., Adaptive non-singular fast terminal sliding mode control for car-like vehicles with faded neighborhood information and actuator faults, *Journal of Intelligent & Robotic Systems*, 110(2), 2024, 1-14.
[8] Dang, S. T., Dinh, X. M., Nguyen Van, T., Pham, T. N., Nguyen, N. L., Le, H. X., Enhanced Trajectory Tracking for Car-Like Mobile Robots, *Proc. of the International Conference on Applied Mathematics and Computer Science*, Springer Nature, Cham, Switzerland, pp. 421-432, 2024.
[9] Huang, H., Gao, J., Backstepping and novel sliding mode trajectory tracking controller for wheeled mobile robots, *Mathematics*, 12(10), 2024, 1458.
[10] Sen, P. T. H., Minh, N. Q., Minh, P. X., A new tracking control algorithm for a wheeled mobile robot based on backstepping and hierarchical sliding mode techniques, *Proc. of 2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)*, Bangkok, Thailand, pp. 25-28, 2019.
[11] Yesudhas, A. A., Palanimuthu, K., Lee, S. R., Jeong, J. H., Joo, Y. H., Performance enhancement of PMSG-based WECS using robust adaptive fuzzy sliding mode control, *Control Engineering Practice*, 156, 2025, 106211.
[12] Yılmaz, M., Çorapsız, M. F., A robust MPPT method based on optimizable Gaussian process regression and high order sliding mode control for solar systems under partial shading conditions, *Renewable Energy*, 239, 2025, 122339.
[13] Hu, J., Zhang, Y., Rakheja, S., Adaptive trajectory tracking for car-like vehicles with input constraints, *IEEE Transactions on Industrial Electronics*, 69(3), 2021, 2801-2810.
[14] Zheng, Y., Zheng, J., Shao, K., Zhao, H., Xie, H., Wang, H., Adaptive trajectory tracking control for nonholonomic wheeled mobile robots: A barrier function sliding mode approach, *IEEE/CAA Journal of Automatica Sinica*, 11(4), 2024, 1007-1021.
[15] Dang, S. T., Dinh, X. M., Kim, T. D., Xuan, H. L., Ha, M. H., Adaptive backstepping hierarchical sliding mode control for 3-wheeled mobile robots based on RBF neural networks, *Electronics*, 12(11), 2023, 2345.
[16] Ly, T. T. K., Thai, N. H., Phong, L. T., Design of neural network-PID controller for trajectory tracking of differential drive mobile robot, *Vietnam Journal of Science and Technology*, 62(2), 2024, 374-386.
[17] Ben Jabeur, C., Seddik, H., Design of a PID optimized neural networks and PD fuzzy logic controllers for a two-wheeled mobile robot, *Asian Journal of Control*, 23(1), 2021, 23-41.
[18] Bouhamatou, Z., Abedssemed, F., Fuzzy synergetic control for dynamic car-like mobile robot, *Acta Mechanica et Automatica*, 16(1), 2022, 48-57.
[19] Shirzadeh, M., Amirkhani, A., Shojaeefard, M. H., Behroozi, H., Adaptive fuzzy sliding mode and indirect radial-basis-function neural network controller for trajectory tracking control of a car-like robot, *International Journal of Nonlinear Analysis and Applications*, 10(1), 2019, 153-166.
[20] Shui, Y., Zhao, T., Dian, S., Hu, Y., Guo, R., Li, S., Data-driven generalized predictive control for car-like mobile robots using interval type-2 T-S fuzzy neural network, *Asian Journal of Control*, 24(3), 2022, 1391-1405.
[21] Dinh, X. M., Luong, T. G., Le, X. H., Nguyen, V. T., Kim, D. T., Adaptive Fuzzy Dynamic Surface Control with Sliding Mode Control for Enhanced Trajectory Tracking of Delta Robots, *Journal of Control, Automation and Electrical Systems*, 2025, 1-25.
[22] Li, L., Wang, T., Xia, Y., Zhou, N., Trajectory tracking control for wheeled mobile robots based on nonlinear disturbance observer with extended Kalman filter, *Journal of the Franklin Institute*, 357(13), 2020, 8491-8507.
[23] Zhu, C., Li, B., Zhao, C., Wang, Y., Trajectory Tracking Control of Car-like Mobile Robots Based on Extended State Observer and Backstepping Control, *Electronics*, 13(8), 2024, 1563.
[24] Hussein, M., Zhang, Y., Liu, Z., Distributed fault-tolerant formation control design via high-order sliding mode for a team of car-like vehicles, *IEEE Transactions on Intelligent Vehicles*, 9(2), 2023, 4221-4231.
[25] Yeom, K., Kinematic and dynamic controller design for autonomous driving of car-like mobile robot, *International Journal of Mechanical Engineering and Robotics Research*, 9(7), 2020, 1058-1064.
[26] Cao, J., Song, C., Peng, S., Song, S., Zhang, X., Xiao, F., Trajectory tracking control algorithm for autonomous vehicle considering cornering characteristics, *IEEE Access*, 8, 2020, 59470-59484.
[27] Wang, Z., Li, G., Jiang, H., Chen, Q., Zhang, H., Collision-free navigation of autonomous vehicles using convex quadratic programming-based model predictive control, *IEEE/ASME Transactions on Mechatronics*, 23(3), 2018, 1103-1113.
[28] Zhang, J. J., Fang, Z. L., Zhang, Z. Q., Gao, R. Z., Zhang, S. B., Trajectory tracking control of nonholonomic wheeled mobile robots using model predictive control subjected to lyapunov-based input constraints, *International Journal of Control, Automation and Systems*, 20(5), 2022, 1640-1651.
[29] Bai, G., et al., Path tracking for car-like robots based on neural networks with NMPC as learning samples, *Electronics*, 11(24), 2022, 4232.
[30] Yeom, K., Design of deep neural network based model predictive controller for a car-like mobile robot, *International Journal of Mechanical Engineering and Robotics Research*, 11(8), 2022, 606-613.
[31] Rösmann, C., Hoffmann, F., Bertram, T., Kinodynamic trajectory optimization and control for car-like robots, *Proc. of 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, Canada, pp. 5681-5686, 2017.
[32] Wang, Z., Li, P., Li, Q., Wang, Z., Li, Z., Motion Planning Method for Car-Like Autonomous Mobile Robots in Dynamic Obstacle Environments, *IEEE Access*, 11, 2023, 137387-137400.
[33] Yu, X., Feng, Y., Man, Z., Terminal sliding mode control–an overview, *IEEE Open Journal of the Industrial Electronics Society*, 2, 2020, 36-52.
[34] Shao, K., Zheng, J., Fu, M., Review on the developments of sliding function and adaptive gain in sliding mode control, *Journal of Automation and Intelligence*, 2025, https://doi.org/10.1016/j.jai.2025.06.001.
[35] Nguyen, V. T., Lin, C. Y., Su, S. F., Sun, W., Er, M. J., Global finite time active disturbance rejection control for parallel manipulators with unknown bounded uncertainties, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(12), 2020, 7838-7849.

[36] Dinh, X. M., Dang, X. D., Le, H. X., Nguyen, N. L., Precise trajectory tracking control of 3-DOF delta robots using dynamic surface control combined with sliding mode control, *Journal of Mechanical Science and Technology*, 2025, https://doi.org/10.1007/s12206-025-0951-5.

ORCID iD

Son-Tung Dang https://orcid.org/0009-0007-6035-7916
Xuan-Minh Dinh https://orcid.org/0009-0002-6044-1492
Tuan-Anh Than Ngoc https://orcid.org/0009-0002-7099-2597
Van-Tuan Nguyen https://orcid.org/0009-0006-2329-8403
Dinh-Thai Kim https://orcid.org/0000-0002-9060-4769
Xuan-Hai Le https://orcid.org/0009-0000-5320-6975

How to cite this article: Dang, S.T., Dinh, X.M., Than Ngoc, T.A., Nguyen, V.T., Kim, D.T., Le, X.H., Adaptive Sliding Mode Control for Robust Trajectory Tracking of Car-Like Mobile Robots in Autonomous Navigation Systems, *J. Appl. Comput. Mech.*, 12(3), 2026, 739-758. https://doi.org/10.22055/jacm.2025.48406.5213