Trajectory Prediction-Based Adaptive Takeover Method for Mobile Robot Teleoperation Systems With Communication Delays

Ruoshan Wang, Pengfei Li, Yun-Sheng Zhao, Yun-Bo Zhao, and Yu Kang, *Senior Member, IEEE*

***Abstract*—As a typical application of the Internet of Things (IoT), mobile robot teleoperation systems play a crucial role in dangerous tasks or special scenarios. The passive takeover mechanism provides effective safety assurance for such systems, relying on a takeover request (TOR) generator to issue warnings and an authority transfer controller to manage control handover. However, variable communication delays can degrade the safety and smoothness of the passive takeover process. To address this issue, we propose a trajectory prediction-based adaptive takeover method that redesigns both the TOR generator and the authority transfer controller. The TOR generator issues early warnings based on the predicted robot trajectory in autonomous control mode. The authority transfer controller uses the control input that tracks the predicted human-intended trajectory as a reconstruction of the delayed human input, and adaptively adjusts the weights between the autonomous input and the reconstructed human input based on the risk indicator and the delay impact level. The effectiveness of the proposed method is validated through the human-in-the-loop simulation. The results show that our method ensures better safety and smoothness in the takeover process compared to other methods.**

***Index Terms*—Authority transfer, communication delay, mobile robot, takeover, teleoperation.**

I. Introduction

With the rapid advancement of Internet of Things (IoT) technology, mobile robot teleoperation systems have been extensively deployed across various domains, including remote vehicle operations [1], exploration [2], and rescue missions [3], [4]. In traditional leader–follower teleoperation systems, the robot passively executes human commands, requiring continuous operator involvement and imposing a substantial burden [5]. Advances in artificial intelligence and automation have increased the autonomy of mobile robots, relieving human operators from continuous control. However, achieving full autonomy remains challenging due to environmental complexity and limited algorithm generalization [6]. Consequently, ensuring safety in the presence of mobile robot autonomy has become a central issue in mobile robot teleoperation systems.

To address safety concerns, a passive takeover mechanism initiated by the robot proves effective [7], enabling human operators to assume control from the robot in scenarios exceeding its autonomous capabilities. This mechanism necessitates robotic platforms equipped with autonomous decision-making capabilities. When the autonomous control system detects situations beyond its capabilities, it generates a takeover request (TOR) to alert human operators. Subsequent successful human response triggers an authority transfer protocol, transferring control from autonomous operation to human teleoperation. With the growing prevalence of robots with limited autonomous intelligence, the design of such takeover mechanisms becomes increasingly important.

Typical passive takeover architectures comprise two core components: the TOR generator and the authority transfer controller. The key challenge for the TOR generator lies in determining the appropriate timing for issuing warnings, with current implementations typically adopting predictive or reactive strategies. Predictive approaches issue warnings based on predefined boundaries of autonomous capabilities [8], [9], while reactive strategies consider the time required for humans to prepare for takeover [10], [11]. The authority transfer controller determines how control is handed over to the human operator. Immediate binary switching upon human input detection, though straightforward, often compromises the smoothness of robot operation [12], [13]. Emerging solutions employ progressive authority transfer architectures that dynamically allocate control through weighted integration of human and robot inputs, effectively balancing safety requirements with operational smoothness. Current research advancements in this domain demonstrate improved system performance through multidimensional considerations, including conflict resolution [13], [14], risk assessments [15], [16], and situation awareness metrics [15], [17].

However, communication network delays, particularly variable ones, present significant challenges to the passive takeover in mobile robot teleoperation systems. Prior research has only given this factor limited attention. These challenges can be categorized into two main aspects. First, traditional TOR generator designs may fail to account for communication delays, preventing timely human takeover and thus compromising system safety. Second, variable communication delays can undermine the performance of traditional authority transfer controllers by causing human control inputs to reflect past

Received 10 February 2026; revised 28 March 2026; accepted 15 April 2026. Date of publication 21 April 2026; date of current version 29 June 2026. This work was supported by the National Natural Science Foundation of China under Grant 62473352, Grant 62173317, Grant 92467302, and Grant U25A20454; and in part by the Key Science and Technology Project of Anhui Province under Grant 202523j08050018. *(Corresponding author: Pengfei Li.)*

Ruoshan Wang, Pengfei Li, and Yun-Sheng Zhao are with the Department of Automation, University of Science and Technology of China, Hefei 230027, China (e-mail: ruoshanwang@mail.ustc.edu.cn; puffylee@ustc.edu.cn; zys1030@mail.ustc.edu.cn).

Yun-Bo Zhao and Yu Kang are with the Department of Automation, University of Science and Technology of China, Hefei 230027, China, and also with the Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei 230026, China (e-mail: ybzhao@ustc.edu.cn; kangduyu@ustc.edu.cn).

Data is available online at https://github.com/whhpnn/AdaptiveTakeover.git

Digital Object Identifier 10.1109/JIOT.2026.3686125

rather than current conditions of the robot [18], [19]. This temporal mismatch increases cognitive workload, hindering the operator's ability to take over control swiftly and effectively, thus resulting in unstable operation and posing safety risks [20], [21].

To address the above challenges, we propose a trajectory prediction-based adaptive takeover method, which redesigns both the TOR generator and the authority transfer controller. Specifically, the TOR generator predicts the mobile robot's future trajectory in autonomous control mode to issue early warnings. The authority transfer controller predicts the trajectory of the mobile robot that reflects human intent, reconstructs the delayed human input into the control input that tracks the human-intended trajectory, and adaptively adjusts the weights between the autonomous input and the reconstructed human input based on the risk indicator and the delay impact level.

Our main contributions are summarized as follows.

1) A new remote passive takeover framework has been proposed, which is capable of compensating for the impact of variable communication delays to ensure the safety and smoothness of the takeover process.
2) A trajectory-prediction-based TOR generation mechanism has been designed that anticipates takeover demands under autonomous control mode and issues early warnings to the human operator despite communication delays, a critical factor overlooked in [10] and [11].
3) An adaptive authority transfer mechanism has been developed, which, compared with existing strategies [14], [15], can address the temporal mismatch between human and robot by replacing human inputs impacted by delay with reconstructed ones, while accounting for the delay-induced increase in cognitive workload when determining how to transfer authority.

The remainder of this article is organized as follows. Section II provides the problem statement. Section III describes the details of our method. The experimental processes and results are shown in Section IV. Finally, Section V concludes this article.

II. Problem Formulation

The passive takeover framework for mobile robot teleoperation systems is illustrated in Fig. 1. It is composed of four essential components: a mobile robot, an onboard autonomous controller, an onboard passive takeover module, and a remote human operator. The information exchange between the human operator and the mobile robot occurs through communication networks, forming the human control loop. Onboard data transmission between the autonomous controller and the mobile robot forms the autonomous control loop. Detailed descriptions of each component are provided below.

A. Robot Model

The evolution of the mobile robot is described by the following discrete-time model:

$$x(k+1) = f(x(k), u(k)) \tag{1}$$

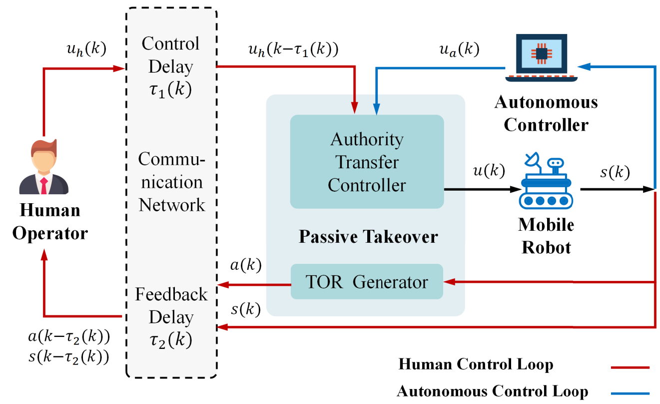

Fig. 1. Passive takeover framework for mobile robot teleoperation systems. The TOR generator issues a signal $a(k)$ based on the sensing data $s(k)$ to warn the human operator. The authority transfer controller generates the hybrid input $u(k)$ based on the autonomous input $u_a(k)$ and the delayed human input $u_h(k - \tau_1(k))$, thereby transferring control authority from robot to human.

where $x(k) \in \mathbb{X} \subseteq \mathbb{R}^n$ represents the robot state, $u(k) \in \mathbb{U} \subseteq \mathbb{R}^m$ is the control input, $\mathbb{X}$ and $\mathbb{U}$ are the constraint sets for the robot state and the control input, respectively, and k denotes the index of the discrete-time step.

Remark 1: The system in (1) represents a commonly used general form for modeling the evolution of mobile robots [22], [23]. The specific form of the model depends on the type of mobile robot. For instance, wheeled robots are often described using the bicycle model [24] or differential drive kinematics [25], whereas six degrees-of-freedom dynamic systems are suitable for modeling aerial robots [26].

B. Autonomous Controller

The autonomous controller is capable of independently operating the mobile robot in most task scenarios. Specifically, it extracts the robot state $x(k)$ and environmental state $v(k)$ from sensing data $s(k)$. The robot state $x(k)$ represents the mobile robot's intrinsic characteristics, such as velocity and position. The environmental state $v(k)$ describes the robot's surroundings, such as obstacle locations and road distributions. Subsequently, the controller generates autonomous inputs $u_a(k)$ based on $x(k)$ and $v(k)$. In conclusion, the autonomous controller can be expressed by the following equation:

$$[x(k), v(k)] = g(s(k)) \tag{2}$$

$$u_a(k) = \pi(x(k), v(k)) \tag{3}$$

where g and π can be interpreted as the state estimation law and the control law, respectively.

Note that the design of g depends on the type of information contained in $s(k)$. For example, when $s(k)$ represents data such as position, velocity, or acceleration, g is generally formulated as a simple linear mapping. In contrast, when $s(k)$ encompasses multimodal information, such as images or audio, g takes the form of a complex nonlinear mapping, often constructed using deep neural networks [27]. The control law π can be designed through traditional model-driven control techniques, including optimal control [28] and model predictive control (MPC) [29], or through intelligent data-driven approaches, such as reinforcement learning [30].

C. Communication Network

The communication network serves as the medium for exchanging both sensing data and control inputs between the human and the mobile robot. The communication network consists of two channels. One channel is responsible for transmitting human inputs to the mobile robot. The time delay of this channel is defined as the control delay, denoted as $\tau_1(k)$. The other channel is in charge of delivering the sensing data to the human operator. The time delay of this channel is defined as the feedback delay, denoted as $\tau_2(k)$. $\tau_1(k)$ and $\tau_2(k)$ satisfy the following commonly adopted assumption [31], [32].

Assumption 1: The control delays $\tau_1(k)$ and feedback delays $\tau_2(k)$ are time-varying and satisfy $0 \le \tau_1(k) \le T_1$, $0 \le \tau_2(k) \le T_2$ $\forall k \in \mathbb{Z}$.

D. Passive Takeover Module

When a mobile robot encounters intractable situations in autonomous control mode, it is necessary to issue warnings and transfer control authority to the human operator, referred to as passive takeover [7]. We define an admissible set $\boldsymbol{S}$ for the robot–environment system state tuple $(x(k), v(k))$, which characterizes the conditions under which the robot can safely and autonomously perform tasks

$$\boldsymbol{S} = \left\{(x, v) \;\middle|\; p\left(x(k), v(k)\right) \le \varepsilon\right\} \tag{4}$$

where $p(x(k), v(k))$ serves as a risk indicator that quantifies the potential threat to the autonomous controller's ability to accomplish the task. The takeover demand threshold ε defines the boundary within which the autonomous controller can maintain safe and independent control. A human takeover becomes necessary once $(x(k), v(k)) \notin \boldsymbol{S}$.

Remark 2: The definition of $p(x(k), v(k))$ has been extensively discussed in previous research. In [33], $p(x(k), v(k))$ was regarded as a driving risk potential determined by the consequences of the event and its probability of occurring. Chiu et al. [34] associated $p(x(k), v(k))$ with environmental threats and the internal state of the robot. In [35], $p(x(k), v(k))$ was defined in relation to the failure occurrence frequency, duration, the severity of harm, and influence on the availability. The passive takeover module depends on two components: the TOR generator and the authority transfer controller. A detailed description of these two modules follows.

Typically, TOR is generated only when a takeover is required. For mathematical convenience, the TOR generator is assumed to produce the TOR signal $a(k) \in \{0, 1\}$ at every time step. 1 and 0 denote the issuance and absence of a warning, respectively. Given that communication delays introduce additional time for the transmission of $a(k)$ and human inputs, it is imperative to properly design the $a(k)$ generation mechanism.

The authority transfer controller dynamically determines the allocation of control authority between the robot and the human operator, and generates a hybrid control input accordingly. A typical authority transfer controller can be described in the form of a linear weighting function [36]

$$u(k) = \lambda(k)\, u_a(k) + (1 - \lambda(k))\, u_h(k) \tag{5}$$

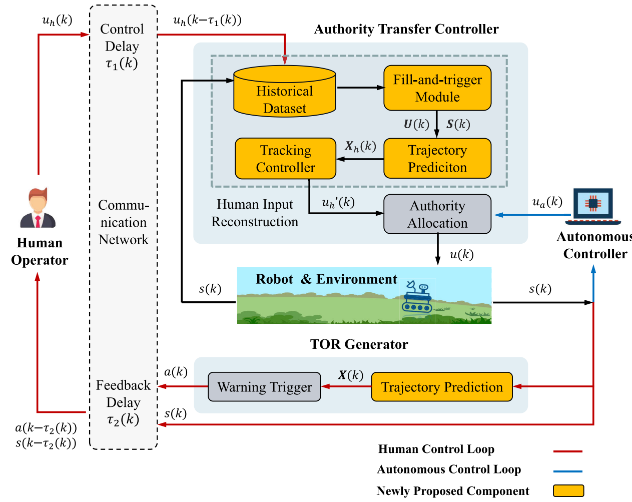

Fig. 2. Trajectory prediction-based adaptive takeover framework. The TOR generator predicts the future trajectory of the mobile robot in autonomous mode and then triggers a proactive warning. The authority transfer controller reconstructs the delayed human input $u_h(k - \tau_1(k))$ as $u_h'(k)$ via intended trajectory tracking, and then allocates control authority between $u_h'(k)$ and $u_a(k)$.

where $u(k)$ represents the hybrid input. $\lambda(k)$ reflects the adjustment of control weight, which is a function that gradually decays from 1 to 0. In particular, when $\lambda(k)$ is a step function jumping from 1 to 0, it indicates that the authority transfer is not gradual but occurs through a direct switch.

However, due to communication delays, the human input $u_h(k - \tau_1(k))$ received by the mobile robot lags behind the real-time sensing data $s(k)$. Accordingly, in the passive takeover of mobile robot teleoperation systems, the authority transfer controller adopts a more targeted formulation

$$u(k) = h\left(\lambda(k), u_a(k), u_h(k - \tau_1(k))\right) \tag{6}$$

where h represents a mechanism to generate the hybrid input $u(k)$. As $\lambda(k)$ gradually decreases from 1 to 0, $u(k)$ will change from being fully determined by $u_a(k)$ to being entirely independent of it. Considering that communication delays pose unique challenges to the safety and smoothness of the takeover process, both λ and h need to be carefully designed.

In summary, our objective is to design the TOR generator and the authority transfer controller (the weight adjustment mechanism λ and the hybrid input generation mechanism h), so that the passive takeover can still maintain safety and smoothness in the presence of two-channel communication delays $\tau_1(k)$ and $\tau_2(k)$.

III. Trajectory Prediction-Based Adaptive Takeover Method

A trajectory prediction-based adaptive takeover framework is proposed to compensate for communication delays, as shown in Fig. 2. The framework is architected around the TOR generator and the authority transfer controller. Sections III-A and III-B provide a detailed description of them.

A. Design of the TOR Generator

To compensate for the additional delays in transmitting $a(k)$ and $u_h(k)$, it is essential to anticipate potential takeover demands and issue warnings in advance. Based on the robot model defined in (1) and the autonomous control law in (3), we perform a forward-looking prediction of the mobile robot's future trajectory in autonomous mode. The predicted trajectory is denoted as $\boldsymbol{X}(k) = [x_1(k), \ldots, x_i(k), \ldots, x_T(k)]$, and is recursively computed as follows:

$$\begin{aligned} x_i(k) &= f(x_{i-1}(k), u_{i-1}(k)), \quad i = 1, \ldots, T \\ u_i(k) &= \pi(x_i(k), v(k)), \quad i = 1, \ldots, T \end{aligned} \tag{7}$$

where variables $x_0(k)$, $u_0(k)$, and $v(k)$ denote the actual robot state, autonomous input, and environmental state at step k, respectively. We approximate the environmental state at each step within the prediction horizon by maintaining $v(k)$ constant. The prediction horizon T is defined as the upper bound of the total delay, that is, $T = T_1 + T_2$.

Subsequently, we calculate the risk indicator $p(x_i(k), v(k))$ for each predicted robot state $x_i(k)$ and environment state $v(k)$. If there is a predicted step at which the corresponding state tuple $(x(k), v(k))$ violates the constraints of set $\boldsymbol{S}$ in (4), a warning will be issued. The specific mechanism for generating $a(k)$ is as follows:

$$a(k) = \begin{cases} 1, & \exists i \in \{1, \ldots, T\}, (x_i(k),\ v(k)) \notin \boldsymbol{S} \\ 0. & \forall i \in \{1, \ldots, T\}, (x_i(k),\ v(k)) \in \boldsymbol{S}. \end{cases} \tag{8}$$

The proposed TOR generator assumes a static surrounding environment within the prediction horizon. This assumption is reasonable for low-delay scenarios with limited environmental dynamics. However, under highly dynamic conditions, this approximation may lead to overly conservative or missed TOR triggers. Future work will consider incorporating dynamic environment uncertainty modeling into the design framework of the TOR generator.

B. Design of the Authority Transfer Controller

To compensate for communication delays, the design of the authority transfer controller is divided into two processes: human input reconstruction and authority allocation. The following sections provide detailed descriptions of them.

1) Human Input Reconstruction: Due to communication delays, the human inputs received by the mobile robot often lag behind the actual sensing data. If the authority transfer controller directly uses these delayed human inputs, the performance of the takeover process may be significantly degraded. Inspired by the work in [37], we predict the human-intended trajectory for the mobile robot and reconstruct the human inputs by tracking this predicted trajectory to compensate for delays. The reconstruction process is described in detail below.

First, we construct a historical dataset on the robot side to store human inputs and sensing data. For convenience, we denote the human input and sensing data stored at index k of the historical dataset as $u_h^H(k)$ and $s^H(k)$, respectively.

As shown in Fig. 3, when the robot perceives the sensing data $s(k)$ at time step k, it is directly stored in the historical dataset entry with the corresponding timestamp. Due to communication delays, the robot receives the human input $u_h(k-\tau_1(k))$ at time step k. We denote the sensing data actually observed by the human operator when issuing $u_h(k-\tau_1(k))$

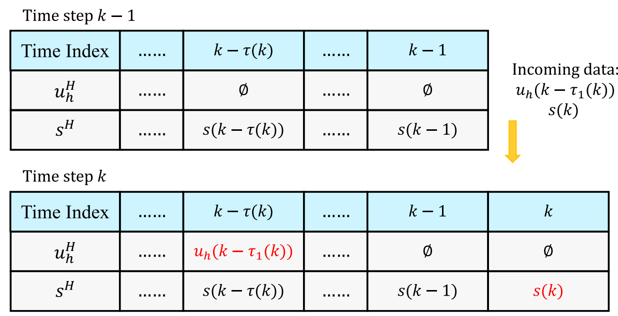

Fig. 3. Data storage process of the historical dataset (due to communication delays, there exist time index intervals in which s^H has value, whereas the corresponding u_h^H is empty).

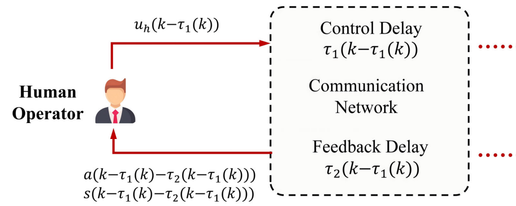

Fig. 4. Data flow at time step $k - \tau_1(k)$.

as $s(k-\tau(k))$, and store $u_h(k-\tau_1(k))$ in the entry aligned with $s(k-\tau(k))$.

We replace k in Fig. 1 with $k-\tau_1(k)$, resulting in Fig. 4, which demonstrates that the sensing data perceived by the human when issuing $u_h(k-\tau_1(k))$ is $s(k-\tau_1(k)-\tau_2(k-\tau_1(k)))$, where $\tau_2(k-\tau_1(k))$ is the feedback delay at time step $k-\tau_1(k)$. Therefore, the following relationships hold:

$$\begin{aligned} s^H(k) &= s(k) \\ u_h^H(k-\tau(k)) &= u_h(k-\tau_1(k)) \\ \tau(k) &= \tau_1(k) + \tau_2(k-\tau_1(k)). \end{aligned} \tag{9}$$

The historical dataset continuously receives new sensing data and human inputs. Considering the storage constraints, we set a predefined storage limit for the dataset. Once the dataset reaches this limit, the system adopts a first-in, first-out strategy to remove older data and add the latest data, as the subsequent trajectory prediction module relies only on relatively recent data within a fixed-length sliding window.

Second, we extract the sequence of human inputs $\boldsymbol{U}(k)$ and the sequence of sensing data $\boldsymbol{S}(k)$ from the historical dataset using a sliding window of fixed length W. These sequences are explicitly defined as follows:

$$\boldsymbol{U}(k) = \left[u_h^H(t_k), u_h^H(t_k+1), \ldots, u_h^H(t_k+W-1)\right] \tag{10}$$

$$\boldsymbol{S}(k) = \left[s^H(t_k), s^H(t_k+1), \ldots, s^H(t_k+W-1)\right] \tag{11}$$

where t_k and $t_k + W - 1$ denote the starting and ending timestamps of the sliding window at step k, respectively.

It is worth noting that the sliding window lacks sufficient historical human inputs during the takeover initiation phase. In addition, variations in communication delay may lead to packet reordering, which can also result in data missing.

To facilitate the handling of such missing data, we introduce a buffering and reordering module on the side of the human operator. This module presents the sensing data to the human

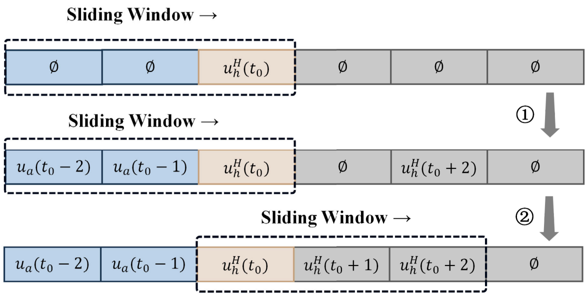

Fig. 5. Fill-and-trigger mechanism for the sliding window: 1) filling the missing human inputs based on autonomous inputs. The sliding window movement is not triggered because the next human input is empty and 2) the sliding window movement is triggered based on (13) and (14).

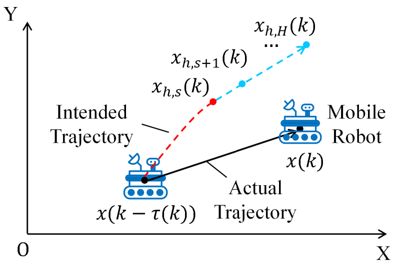

Fig. 6. Illustration of human-intended trajectory.

operator in a temporally aligned form, $s(t - T_2)$. In other words, the feedback delay $\tau_2(k)$ is uniformly extended to a fixed value T_2. As a result, only the problem of missing human inputs needs to be handled.

To address the aforementioned issue, we propose a fill-and-trigger mechanism for the sliding window, as illustrated in Fig. 5. We define $u_h(t_k + i) = \emptyset$ to indicate the absence of human input at timestamp $t_k + i$. The first human input recorded in the historical dataset is denoted by $u_h^H(t_0)$, which divides the dataset into two segments. The earlier segment corresponds to the period before the initiation of the takeover. Since the autonomous controller typically maintains stable performance before takeover, the autonomous inputs during this period can be regarded as approaching human intent. Consequently, we fill the missing values in this segment with the corresponding autonomous inputs

$$u_h^H(t_0 - i) = u_a(t_0 - i). \tag{12}$$

In contrast, the missing human inputs in the latter segment are primarily caused by packet reordering. Rather than applying input filling, we introduce a triggering mechanism to advance the sliding window. The event-triggering function $\phi(k)$ and the window update rule t_{k+1} are defined as follows:

$$\phi(k) = \begin{cases} 1, & \text{if } u_h^H(t_k + i) \neq \emptyset \quad \forall i \in \{0, 1, \ldots, W\} \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

$$\begin{aligned} t_{k+1} &= \begin{cases} t_k + D, & \phi(k) = 1 \\ t_k, & \phi(k) = 0 \end{cases} \\ D &= \max\left\{ d \in \mathbb{N} \;\middle|\; u_h^H(t_k + W - 1 + i) \neq \emptyset \right. \\ &\qquad \left. \forall i \in \{0, 1, \ldots, d\} \right\}. \end{aligned} \tag{14}$$

According to (13) and (14), assuming that the missing human inputs preceding $u_h^H(t_0)$ have been filled, the sliding window is updated as follows: if the next human input in the window's forward direction, $u_h^H(t_k + W)$, is missing, the sliding window remains stationary. Conversely, once $u_h^H(t_k + W)$ becomes nonempty, the window immediately shifts forward to the position preceding the next missing human input.

This fill-and-trigger mechanism ensures that the extracted human input sequence $\boldsymbol{U}(k)$ contains no missing entries. Moreover, based on timestamp alignment, the corresponding sensing data sequence $\boldsymbol{S}(k)$ can be readily obtained.

Third, we construct a prediction model to forecast the future trajectory of the mobile robot that reflects human intent. The prediction model is built upon a long short-term memory (LSTM) network, which is widely used for intended trajectory prediction due to its ability to capture long-term temporal dependencies. The trajectory prediction task is formulated as a multistep sequence-to-sequence problem, with the model adopting an encoder–decoder architecture. The encoder takes as input the sensing data sequence $\boldsymbol{S}(k)$ and the human input sequence $\boldsymbol{U}(k)$. The decoder outputs the intended state sequence $\boldsymbol{X}_h(k) = [x_{h,1}(k), \ldots, x_{h,i}(k), \ldots, x_{h,H}(k)]$, where H denotes the prediction horizon. Considering the necessity to track the human-intended trajectory for delay compensation, the prediction horizon is set to be no less than the upper bound of the total delay, i.e., $H \geq T_1 + T_2$.

Finally, we employ MPC to track the predicted human-intended trajectory. Fig. 6 illustrates the human-intended trajectory (dashed line) and the mobile robot's actual trajectory (solid line). A distinct separation point $x_{h,s}(k)$ divides the intended trajectory into a past segment (in red) and a future segment (in blue), corresponding to an estimate of the mobile robot's state at step k. A sequence of intended states, denoted as $X_s = [x_{h,s+1}(k), x_{h,s+2}(k), \ldots, x_{h,H}(k)]$, is generated along the future segment of the intended trajectory. To track the sequence X_s, the following MPC scheme is employed:

$$\boldsymbol{u}_{opt}(k) = \arg\min_{\boldsymbol{u}(k)} \sum_{i=1}^{N} \Big(\left\| x_i(k) - x_{h,s+i}(k) \right\|_Q^2 + \|u_i(k)\|_R^2 + \|\Delta u_i(k)\|_G^2 \Big) \tag{15a}$$

$$\text{s.t.} \quad x_{i+1}(k) = f(x_i(k), u_i(k)) \tag{15b}$$

$$x_i(k) \in \mathbb{X} \tag{15c}$$

$$u_i(k) \in \mathbb{U} \tag{15d}$$

where $x(k)$ represents the robot's state, and $u(k)$ is the control input. Equation (15b) describes the dynamics of the mobile robot. Equations (15c) and (15d) are hard constraints on the state and control input. $x_h(k)$ is the intended state on the predicted trajectory. To improve the smoothness of control, the cost function includes penalty terms on both the control input $u(k)$ and its increment $\Delta u(k)$. The weighting matrices Q, R, and G are diagonal. N denotes the prediction horizon of the MPC. To ensure a sufficient number of intended states, the prediction horizon of the intended trajectory is explicitly defined as

$$H = T_1 + T_2 + N. \tag{16}$$

By minimizing the cost function, we obtain the optimal input sequence $\boldsymbol{u}_{opt}(k)$, whose first element is regarded as the reconstructed human input $u_h'(k)$

$$\boldsymbol{u}_{opt}(k) = \left[u_{opt}(k|k), u_{opt}(k+1|k), \ldots, u_{opt}(k+N-1|k)\right]$$
$$u_h'(k) = u_{opt}(k|k). \quad (17)$$

2) Authority Allocation: The authority allocation module adaptively adjusts the weights between the autonomous input $u_a(k)$ and the reconstructed human input $u_h'(k)$

$$u(k) = \lambda(k) u_a(k) + (1 - \lambda(k)) u_h'(k). \quad (18)$$

To comprehensively consider the smoothness and safety of the takeover process under communication delays, the weight adjustment mechanism $\lambda(k)$ is designed as follows:

$$\lambda(k) = \begin{cases} \mathrm{sat}(\lambda(k-1) - \Delta\lambda(k)), & p(x(k), v(k)) < \varepsilon_u \\ 0, & p(x(k), v(k)) \geq \varepsilon_u \end{cases}$$
$$\Delta\lambda(k) = K(b - D(k)) \quad (19)$$

where the saturation function $\mathrm{sat}(x) \triangleq \min(\max(x, 0), 1)$ constrains $\lambda(k)$ within $[0, 1]$. $D(k)$ represents the delay impact level. ε_u is the emergency switching threshold, serving as a hard constraint that forces the autonomous controller to exit the control loop. K and b represent the authority transfer rate and the operator's tolerance to delays, respectively.

The delay impact level $D(k)$ is quantified as the distance from the starting point of the human-intended trajectory to the mobile robot's current position. Increased robot speed and delay lead to a larger value of $D(k)$. The condition $D(k) < b$ indicates that the delay impact remains within the human-tolerable range. The value of ε_u is related to the inherent capabilities of the autonomous system and should satisfy $\varepsilon_u \geq \varepsilon$. Both K and b are positive parameters and are influenced by individual factors such as skill level and reaction time. For operators with high skill levels and rapid response times, higher values of K and b are typically more suitable.

Remark 3: A procedure is proposed to determine the initial reference values b^* and K^* for tuning b and K, as follows:

By tracking the intended trajectory, the teleoperation task is conducted under fixed velocity and delay conditions, which are gradually increased respectively until the task becomes impossible. Then the corresponding velocity v^* and communication delay τ^* are recorded, and $b^* = v^*\tau^*$.

By conducting pretests on the given task, we can estimate the average delay impact level D^*. The takeover stabilization time is defined as the time interval from takeover onset to the point at which control signal variability decreases substantially [38]. The average takeover stabilization time of participants can be estimated from their historical takeover performance and denoted as T^*. Accordingly, $K^* = 1/(T^*(b^* - D^*))$.

The study mainly focuses on tasks where the delay impact remains consistently within the human-tolerable range. As shown in Fig. 7, in such tasks, the authority transfer process can be divided into two phases.

1) When $p(x(k), v(k)) < \varepsilon_u$, the control authority is gradually and adaptively transferred based on the delay impact level $D(k)$. As $D(k)$ increases, the human cognitive workload also increases, even with the application of reconstruction [37]. Therefore, as $D(k)$ grows, the rate of authority transfer is gradually reduced, thereby facilitating the operator's adaptation to the collaboration with the mobile robot in authority transfer and improving the smoothness of the takeover process.
2) When $p(x(k), v(k)) \geq \varepsilon_u$, direct switching is used to promptly remove the autonomous controller to meet safety requirements.

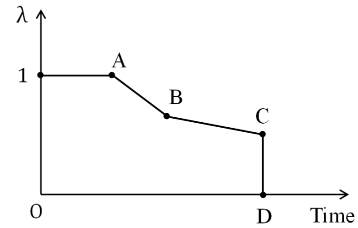

Fig. 7. Conceptual diagram of the authority transfer process. AC represents the progressive transfer phase: AB corresponds to lower $D(k)$, whereas BC corresponds to higher $D(k)$. CD denotes the switching phase.

Fig. 8. Human-in-the-loop remote driving simulation platform.

Let $D_{\max}$ denote the maximum value of $D(k)$ during authority transfer. In tasks where the delay impact is consistently tolerable by the human operator, we have $\Delta\lambda(k) \geq K(b - D_{\max}) > 0$. Consequently, even without entering the switching phase, $\lambda(k)$ converges to zero in finite steps.

For completeness, if $D(k) \geq b$, the takeover is immediately terminated, serving as a redundant safety strategy.

IV. Experiment

The remote driving system serves as a representative mobile robot teleoperation system, which is well-suited as an example to validate the effectiveness of the proposed method.

A. Simulation Platform

As illustrated in Fig. 8, a human-in-the-loop remote driving simulation platform was developed to evaluate the proposed method. The vehicle-road system was simulated using the CARLA simulator, which incorporated an RGB camera to provide the driver's perspective of the environment. The camera, the sensors used to collect human input, and the other components were all set to operate at a consistent update frequency of 20 Hz. Driver interaction was facilitated via a Logitech G29 steering wheel. The TOR was multimodal,

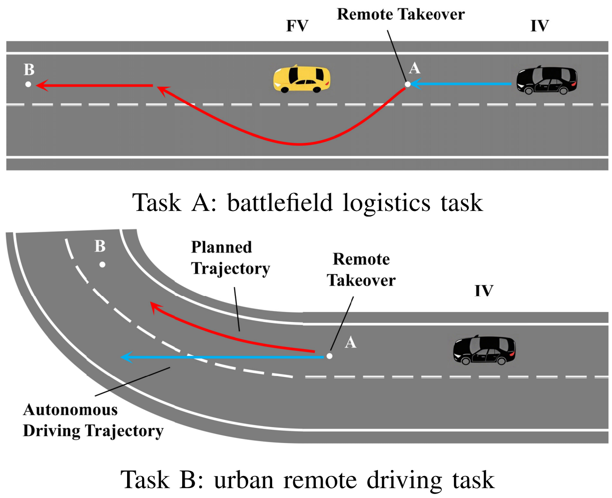

Fig. 9. Remote takeover scenarios in the experiment.

comprising an auditory signal and a visual message reading "Please take over!" to alert the driver when takeover was required. In addition, the virtual communication network was implemented using two first-in-first-out (FIFO) queues.

B. Delay Design

The generalized extreme value (GEV) distribution is widely used to model communication delays [1]. The probability density function of GEV(ξ, μ, σ) is expressed as

$$f_{\xi\neq 0,\mu,\sigma} = \frac{1}{\sigma}\left(1+\xi\frac{x-\mu}{\sigma}\right)^{-\frac{1}{\xi}-1} e^{-\left(1+\xi\frac{x-\mu}{\sigma}\right)^{-\frac{1}{\xi}}} \tag{20}$$

where x is a random delay variable, ξ is the shape parameter, μ is the location parameter, and σ is the scale parameter.

In this article, the random control delays and feedback delays were generated by GEV ($\xi = 0.29, \mu = 0.2, \sigma = 0.009$) and GEV ($\xi = 0.58, \mu = 0.4, \sigma = 0.018$), respectively, with the mean values approximating 0.2 and 0.4 s. It reflects the fact that feedback delays are typically higher than control delays due to the larger packet size of image signals.

To satisfy Assumption 1, the upper bounds for control delays and feedback delays were set to $T_1 = 0.3$ s and $T_2 = 0.5$ s, respectively. Furthermore, continuous delays were discretized by the system sampling period (0.05 s).

C. Task Design and Procedure

We design two tasks, as shown in Fig. 9. Task A depicts a battlefield logistics scenario in which an intelligent vehicle (IV) is assigned to transport supplies to point B. However, a faulty vehicle (FV) is stalled at a random position 25–35 m ahead of point B, a situation not anticipated in the IV's original design. This unexpected obstacle significantly hinders the IV's ability to complete the task autonomously, thereby necessitating remote human takeover from point A to maneuver around the obstacle and continue transporting the supplies to point B.

Task B involves an urban remote driving scenario where the IV follows a pre-planned route to maintain lane-keeping along the centerline. However, when the IV reaches point A, external disturbances cause the actual driving trajectory to deviate from the planned path, necessitating a remote takeover by a safety operator in the cloud. The operator then controls the IV to drive approximately 80 m to point B.

The evolution of the IV in both tasks is modeled using a bicycle model, formulated as follows:

$$\begin{cases} x_p(k+1) &= x_p(k) + T v_l \cos\theta(k) \\ y_p(k+1) &= y_p(k) + T v_l \sin\theta(k) \\ \theta(k+1) &= \theta(k) + \frac{T v_l}{L}\tan\delta(k) \end{cases} \tag{21}$$

where x_p and y_p denote the position coordinates, θ is the heading angle, L is the wheelbase of the vehicle, v_l is the longitudinal velocity, and δ is the steering angle. The sampling period is $T = 0.05$ s. The robot state and control input are represented by $x = [x_p, y_p, \theta]$ and $u = [v_l, \delta]$, respectively. To simplify the experiment, v_l is fixed at one of three values: 5, 8, or 10 m/s, which restricts human takeovers to steering angle control only.

The risk indicator of Task A is defined as the driving risk potential P

$$P = \frac{4000}{d(IV, FV)^2} \tag{22}$$

where $d(IV, FV)$ denotes the distance between the IV and the FV. The takeover demand threshold ε is set to 2, while the emergency switching threshold ε_u is designated as 10.

The risk indicator of Task B is defined as the minimum distance between the current position and the planned trajectory. The takeover demand threshold ε is set to 0.6, while the emergency switching threshold ε_u is designated as 3.

Both Task A and Task B employ an LSTM model with a Seq2Seq architecture, consisting of two stacked LSTM layers in the encoder and two stacked LSTM layers in the decoder, with the hidden state dimension set to 128. The input sequence length is 40, and the output sequence length is 20. The model is trained using Adam with an initial learning rate of 0.001 and mean squared error as the loss function.

For Task B, the input consists of 5-D features, including $x_p, y_p, \theta, v_l, \delta$. For Task A, the distance between the IV and FV is further incorporated as an environmental state and treated as an additional input feature.

The cost function of the MPC for tracking the intended trajectory is formulated as follows:

$$\begin{aligned} J = \sum_{i=1}^{4} \Big[& q_1 \left(x_{p,i}(k) - x^p_{h,s+i}(k)\right)^2 \\ & + q_2\left(y_{p,i}(k) - y^p_{h,s+i}(k)\right)^2 \\ & + r\delta_i(k)^2 + g\Delta\delta_i(k)^2\Big] \end{aligned} \tag{23}$$

where $x^p_{h,s+i}(k)$ and $y^p_{h,s+i}(k)$ denote the x-coordinates and y-coordinates of the human intended position, respectively. No strict constraints are imposed on the position states. The control constraint set $\mathbb{U}$ limits the steering angle $\delta_i(k)$ to lie within $[-0.5, 0.5]$ radians. The prediction horizon is set to 4, and the empirically tuned optimal parameters q_1, q_2, r, and g for different tasks and speeds are summarized in Table I.

To validate our approach, we recruited 12 volunteers (six male, six female) from multiple age groups, with an average

TABLE I
KEY PARAMETERS FOR MPC

Task	Speed (m/s)	q_1	q_2	r	g
Task A	5	6	4	1	0.5
	8	5	3	2	2
	10	8	4	5	3
Task B	5	4	4	2	2
	8	5	5	3	3
	10	4	4	6	6

age of 26 years, to participate in the experiment. All participants provided informed consent. They were invited to take over the vehicle in five modes.

1) *Mode A:* The proposed method.
2) *Mode B:* The robot generates TOR based on (4) and switches control authority immediately upon human input detection.
3) *Mode C:* The robot generates TOR based on (4) and adopts a delay-unaware progressive strategy to adjust control weights [14].
4) *Mode D:* The robot generates TOR based on (4) and uses the proposed authority transfer controller.
5) *Mode E:* The robot adopts the proposed TOR generator and switches control authority immediately upon human input detection.

Each participant was required to complete four takeover tasks at each speed, in each mode, and in each scenario. To eliminate potential subjective bias toward any specific mode, the tasks were randomized, and participants were unaware of the mode used in each task.

D. Evaluation Metrics

The study aims to improve the safety and smoothness of remote passive takeovers of mobile robots under variable communication delays. Six evaluation metrics were developed to systematically assess these two aspects.

The general metrics are as follows.

1) *TOR Response Time ($TRRT$):* The $TRRT$ is defined as follows:

$$TRRT = t_f - t_\varepsilon \tag{24}$$

where t_ε denotes the time at which the risk indicator exceeds the takeover demand threshold ε, and t_f is when the robot detects the first human input, marking the initiation of the authority transfer. A smaller $TRRT$ indicates a quicker human response to the TOR, indirectly reflecting takeover safety.

The metrics specific to Task A are as follows.

1) *Accident Number (AN):* An accident occurs when the IV collides with the FV or completely crosses a lane boundary. As a direct measure of safety, a lower AN indicates a higher level of takeover safety.
2) *Minimum Accident Distance (MAD): MAD* refers to the minimum Euclidean distance between the IV and either the lane boundary or the FV during takeover. It is set to zero in the event of an accident. This metric quantifies the vehicle's proximity to potential hazards, with a larger MAD indicating a greater safety margin.

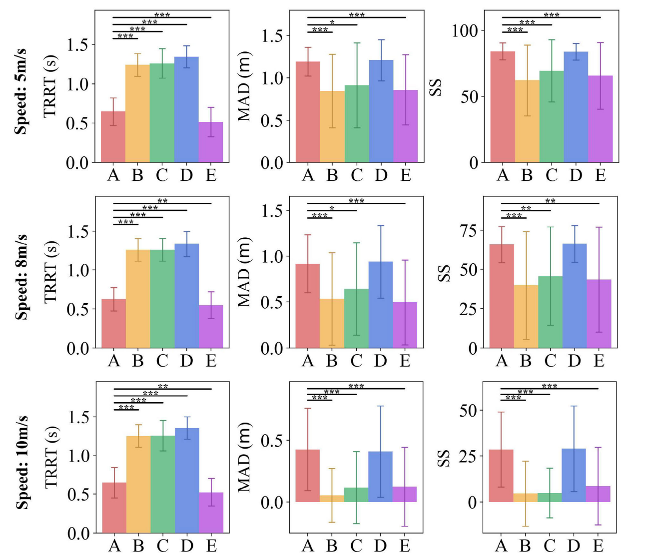

Fig. 10. Comparison of $TRRT$, MAD, and SS for Task A in different takeover modes.

TABLE II
AN FOR TASK A IN DIFFERENT TAKEOVER MODES

Speed	Mode A	Mode B	Mode C	Mode D	Mode E
5 m/s	0	6	4	0	5
8 m/s	0	17	13	0	16
10 m/s	10	45	42	11	42

3) *Smoothness Score (SS):* Accidents may disrupt data integrity and introduce bias, making the average steering angle (ASA) an unreliable metric for quantifying smoothness in Task A. Therefore, an SS is defined as

$$SS = 100 \times \frac{\bar{\delta}_{\max} - \bar{\delta}}{\bar{\delta}_{\max} - \bar{\delta}_{\min}} \tag{25}$$

where $\bar{\delta}$ denotes the ASA during the takeover. $\bar{\delta}_{\min}$ and $\bar{\delta}_{\max}$ represent the minimum and maximum ASAs among all accident-free trials, respectively. SS is set to zero if an accident occurs. A higher SS reflects better takeover smoothness.

The metric specific to Task B is as follows.

1) *Mean Squared Distance (MSD): MSD* is defined as the mean of squared minimum distances from actual vehicle positions to the planned trajectory centerline. A lower MSD indicates that the vehicle follows the planned trajectory more accurately, reflecting a safer takeover.
2) *ASA: ASA* captures the ASA observed during the takeover. A smaller ASA suggests that the human driver managed the vehicle more smoothly.

E. Experiment Results

The experimental results for Task A are presented in Fig. 10 and Table II, while the results for Task B are shown in Fig. 11.

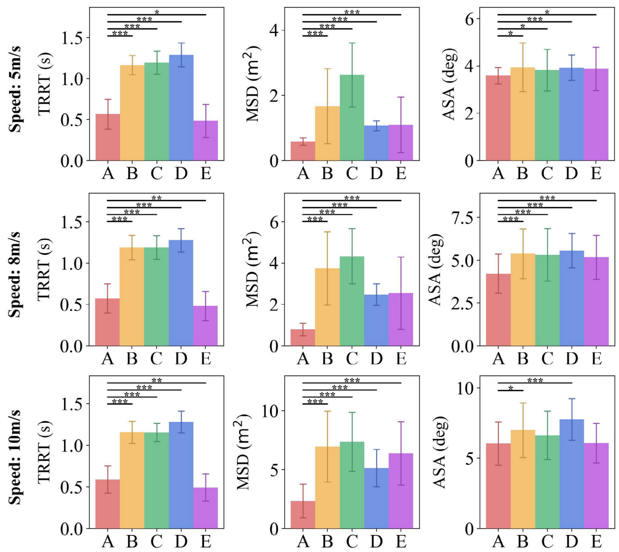

Fig. 11. Comparison of $TRRT$, MSD, and ASA for Task B in different takeover modes.

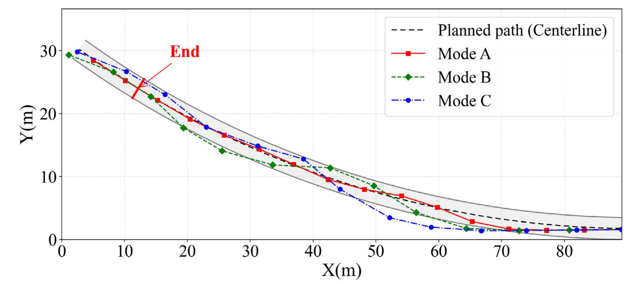

Fig. 12. Representative driving paths in different takeover modes.

We analyzed the evaluation metrics using the Mann–Whitney U test, a nonparametric test that does not assume normality. When $p < 0.05$, it indicates that there is a statistically significant difference in the corresponding metric between the two modes. We mark the p-values from the U test with significance indicators: * for $p < 0.05$, ** for $p < 0.01$, and *** for $p < 0.001$.

In the following section, we first conduct a baseline comparison of our method with other takeover baselines, followed by an ablation study to analyze the contribution of each component. Finally, we elaborate on the underlying mechanisms of our method through representative cases.

1) Baseline Comparison: Across all experiments, the $TRRT$ of our method is significantly lower than that of modes B and C. This suggests that the proposed framework effectively enhances the human operator's responsiveness to risk, indirectly ensuring takeover safety. It is worth noting that our method does not guarantee $TRRT < 0$ due to human reaction latency, which is outside the scope of this study. The $TRRT$ of our method is around 0.6 s, which falls within the reasonable range for human reaction time. Therefore, we focus solely on the relative values of $TRRT$.

As for other metrics, in Task A, our method avoids accidents at both 5 and 8 m/s, while accidents occur under modes B and C. In most cases, our method shows significantly higher MAD and SS for Task A, and significantly lower MSD and ASA for Task B, indicating the superiority of our method in ensuring the safety and smoothness of the takeover process.

However, the performance of our method degrades across many metrics as speed increases. Our method caused accidents in Task A at 10 m/s. Although Task B does not define accidents, the average MSD and ASA values increased by 35% and 17% from 5 to 8 m/s, but then surged by over 180% and 40% from 8 to 10 m/s, with the growth rates more than doubling compared to the lower-speed interval. This can be explained by two reasons: first, under a fixed prediction horizon, higher speeds amplify trajectory prediction errors. Second, high speeds tend to cause substantial deviations between the vehicle's actual position and its intended trajectory, making it harder for MPC to achieve safe and smooth trajectory tracking.

2) Ablation Study: We compared the metrics under modes A and D to analyze the contribution of the proposed TOR generator. Mode A demonstrated significantly lower $TRRT$ across all groups, highlighting the proposed TOR generator's core contribution—enabling faster human takeover responses.

Interestingly, whether modes A and D differ in other metrics depends on the task type. In Task A, no significant differences in MAD and SS were observed between the two modes. However, mode A significantly reduced MSD and ASA in Task B. The reason may be that the risk indicator in Task B is directly related to the deviation from the planned path. As $TRRT$ increases, the deviation grows, requiring more steering effort to follow the path and reducing safety and smoothness. In contrast, the risk indicator for Task A is more proactive, providing a time window for the vehicle to continue straight after warnings, so the TOR generator has less impact. Overall, we believe the TOR generator's contribution is closely tied to the design of the admissible set $\boldsymbol{S}$.

By comparing the metrics under modes A and E, we evaluate the effectiveness of the authority transfer controller. In Task A, mode A shows lower AN across all groups, with significant increases in MAD and SS. In Task B, excluding the 10 m/s group, mode A significantly reduces MSD and ASA. These results indicate that the proposed authority transfer controller plays an important role in improving safety and smoothness. Although the same TOR generator is used, our method generally results in a higher $TRRT$ due to the extension strategy for handling variable feedback delays. Considering that the $TRRT$ in mode A is also within a reasonable range, this does not imply that our method lacks the ability to compensate for communication delays.

3) Mechanism Interpretation With Representative Cases: Figs. 12 and 13 illustrate representative driving paths and the corresponding steering angle variations for Task B under modes A, B, and C, clearly demonstrating the improvements made by our approach through three key mechanisms:

First, the early-issued TOR provides additional preparation time for humans. Due to delays in TOR transmission, modes B and C maintain autonomous control for extended periods, limiting the time available for the driver to perceive, plan, and execute actions. Our approach issues the TOR in advance, enabling the driver to initiate the path change earlier and

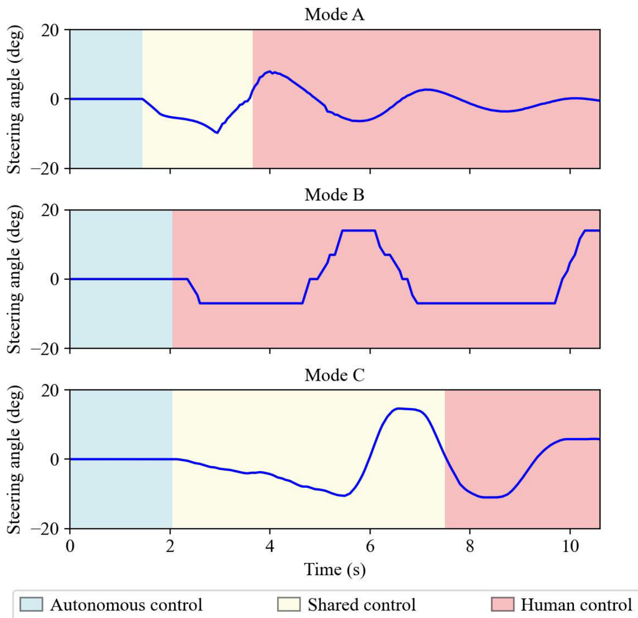

Fig. 13. Steering angle variations in different takeover modes.

thereby improving the safety and smoothness of the takeover process under communication delays.

Second, tracking the human-intended trajectory compensates for the lag in human inputs. In mode C, delayed human commands prevent timely planned path following, causing the driver to make abrupt steering corrections. In mode B, mismatches between sensing data and driving commands lead to an oscillatory vehicle trajectory. Our method addresses this issue by predicting the driver's intended trajectory, thereby ensuring that the vehicle accurately follows the centerline as intended.

Finally, the segmented control authority allocation strikes a balance between smoothness and safety. In mode B, the sudden transition from autonomous mode to human control leads to fluctuations in steering angles. In contrast, prolonged shared control in mode C may introduce safety risks. Our approach maintains smoothness by utilizing shared control during the early stages of the takeover while ensuring the autonomous driving controller disengages promptly under delayed conditions, thus safeguarding both smoothness and safety.

To examine the effect of human-intended trajectory prediction accuracy on our method, we conducted 20 trials in Task A at a speed of 8 m/s using a simplified kinematics-based predictor that assumes a constant turn rate. While no accidents occurred in the original experiment, six accidents were observed, indicating that accurate trajectory prediction is a prerequisite for our method. In real-world scenarios with unstable operator behavior and complex environments, prediction accuracy may degrade, leading to reduced takeover performance. Conversely, more advanced and multimodal trajectory prediction models are expected to further enhance the performance of our method in high-speed scenarios.

Additionally, we conducted a simple additional analysis on the personalized parameters K and b of the authority transfer controller for a subset of participants in Task A. With b fixed, we conducted experiments in which K was varied within $\pm 20\%$ of its final value. Although differences in subjective operational experience were reported, no statistically significant differences were observed in the performance metrics. With K fixed, perturbing b within $\pm 20\%$ led to similar results (the delay impact level remains below b). These results suggest that, within the proposed method, the system does not exhibit strong sensitivity to these parameters, allowing a certain degree of tolerance in their adjustment.

V. Conclusion

This article has proposed a trajectory prediction-based adaptive takeover method for mobile robot teleoperation systems. The TOR generator issues early warning signals by predicting the robot's future trajectory under autonomous control. The authority transfer controller reconstructs delayed human inputs by tracking the human-intended trajectory and adaptively transfers control authority from the autonomous input to the reconstructed human input. This method enhances the safety and smoothness of the remote passive takeover process under varying communication delays. The effectiveness of the proposed approach was validated in representative remote driving scenarios.

It is observed that the performance of the proposed method deteriorates as the robot speed increases. In addition, the current framework lacks formal theoretical guarantees and relies on assumptions such as bounded communication delays. Future work will focus on strengthening theoretical guarantees, relaxing these assumptions, and extending the applicability of the proposed approach to more dynamic environments and higher-speed conditions.

References

[1] Y. Zheng, M. J. Brudnak, P. Jayakumar, J. L. Stein, and T. Ersal, "Evaluation of a predictor-based framework in high-speed teleoperated military UGVs," *IEEE Trans. Hum.-Mach. Syst.*, vol. 50, no. 6, pp. 561–572, Dec. 2020.

[2] J. Teague, M. J. Allen, and T. B. Scott, "The potential of low-cost ROV for use in deep-sea mineral, ore prospecting and monitoring," *Ocean Eng.*, vol. 147, pp. 333–339, Jan. 2018.

[3] A. Birk, S. Schwertfeger, and K. Pathak, "A networking framework for teleoperation in safety, security, and rescue robotics," *IEEE Wireless Commun.*, vol. 16, no. 1, pp. 6–13, Feb. 2009.

[4] J. Di, K. Li, Y. Kang, Q. Dong, S. Chen, and W. Lv, "Semi-adaptive spectrally normalized identifier based model uncertainty online compensation for racing drones," *Sci. China Technol. Sci.*, vol. 68, no. 7, Jul. 2025, Art. no. 1720404.

[5] T.-C. Lin, A. U. Krishnan, and Z. Li, "Physical fatigue analysis of assistive robot teleoperation via whole-body motion mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 2240–2245.

[6] Y. Kang, J. Di, M. Li, Y. Zhao, and Y. Wang, "Autonomous multi-drone racing method based on deep reinforcement learning," *Sci. China Inf. Sci.*, vol. 67, no. 8, Aug. 2024, Art. no. 180203.

[7] S. He, X. Liu, J. Wang, H. Ge, K. Long, and H. Huang, "Influence of conventional driving habits on takeover performance in joystick-controlled autonomous vehicles: A low-speed field experiment," *Heliyon*, vol. 10, no. 11, Jun. 2024, Art. no. e31975.

[8] C. Sun, Z. Deng, W. Chu, S. Li, and D. Cao, "Acclimatizing the operational design domain for autonomous driving systems," *IEEE Intell. Transp. Syst. Mag.*, vol. 14, no. 2, pp. 10–24, Mar. 2022.

[9] C. W. Lee, N. Nayeer, D. E. Garcia, A. Agrawal, and B. Liu, "Identifying the operational design domain for an automated driving system through assessed risk," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 1317–1322.

[10] M. Bahram, M. Aeberhard, and D. Wollherr, "Please take over! An analysis and strategy for a driver take over request during autonomous driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2015, pp. 913–919.

[11] J. Wan and C. Wu, "The effects of lead time of take-over request and nondriving tasks on taking-over control of automated vehicles," *IEEE Trans. Hum.-Mach. Syst.*, vol. 48, no. 6, pp. 582–591, Dec. 2018.

[12] H. E. B. Russell, L. K. Harbott, I. Nisky, S. Pan, A. M. Okamura, and J. C. Gerdes, "Motor learning affects car-to-driver handover in automated vehicles," *Sci. Robot.*, vol. 1, no. 1, p. 5682, Dec. 2016.

[13] T. Wada, K. Sonoda, T. Okasaka, and T. Saito, "Authority transfer method from automated to manual driving via haptic shared control," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2016, pp. 002659–002664.

[14] T. Saito, T. Wada, and K. Sonoda, "Control authority transfer method for automated-to-manual driving via a shared authority mode," *IEEE Trans. Intell. Vehicles*, vol. 3, no. 2, pp. 198–207, Jun. 2018.

[15] Z. Zhang, C. Wang, W. Zhao, C. Xu, and G. Chen, "Driving authority allocation strategy based on driving authority real-time allocation domain," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8528–8543, Jul. 2022.

[16] K. Okada, K. Sonoda, and T. Wada, "Transferring from automated to manual driving when traversing a curve via haptic shared control," *IEEE Trans. Intell. Vehicles*, vol. 6, no. 2, pp. 266–275, Jun. 2021.

[17] C. Lv et al., "A novel control framework of haptic take-over system for automated vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1596–1601.

[18] M. Mostefa, L. K. El Boudadi, and J. Vareille, "Safe and efficient mobile robot teleoperation via a network with communication delay," *Int. J. Interact. Design Manuf. (IJIDeM)*, vol. 12, no. 1, pp. 37–47, Feb. 2018.

[19] T. Zhang, "Toward automated vehicle teleoperation: Vision, opportunities, and challenges," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11347–11354, Dec. 2020.

[20] S. Lu, M. Y. Zhang, T. Ersal, and X. J. Yang, "Workload management in teleoperation of unmanned ground vehicles: Effects of a delay compensation aid on human operators' workload and teleoperation performance," *Int. J. Hum.-Comput. Interact.*, vol. 35, no. 19, pp. 1820–1830, Nov. 2019.

[21] V. Melnicuk, S. Thompson, P. Jennings, and S. Birrell, "Effect of cognitive load on drivers' state and task performance during automated driving: Introducing a novel method for determining stabilisation time following take-over of control," *Accident Anal. Prevention*, vol. 151, Mar. 2021, Art. no. 105967.

[22] S. G. Vougioukas, "Reactive trajectory tracking for mobile robots based on non linear model predictive control," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2007, pp. 3074–3079.

[23] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," in *Proc. Amer. Control Conf. (ACC)*, May 2021, pp. 3882–3889.

[24] H. Zhao, Z. Zhao, D. Xu, and H. Yu, "Event-triggered automatic parking control for unmanned vehicles against DoS attacks," *IEEE Trans. Ind. Cyber-Phys. Syst.*, vol. 2, pp. 531–541, 2024.

[25] N. Wang, M. Marwan, Y. Yang, X. Chen, H. Ho-Ching Iu, and Q. Xu, "Designing nest-fold system via dual-route fractal process and application in chaotic mobile robot," *IEEE Internet Things J.*, vol. 12, no. 7, pp. 9206–9217, Apr. 2025.

[26] Y. Wang, H. Wang, Y. Liu, J. Wu, and Y. Lun, "6-DOF UAV path planning and tracking control for obstacle avoidance: A deep learning-based integrated approach," *Aerosp. Sci. Technol.*, vol. 151, Aug. 2024, Art. no. 109320.

[27] T. Li, B. Huang, C. Li, and M. Huang, "Application of convolution neural network object detection algorithm in logistics warehouse," *J. Eng.*, vol. 2019, no. 23, pp. 9053–9058, Dec. 2019.

[28] X. Li, L. Wang, Y. An, Q.-L. Huang, Y.-H. Cui, and H.-S. Hu, "Dynamic path planning of mobile robots using adaptive dynamic programming," *Expert Syst. Appl.*, vol. 235, Jan. 2024, Art. no. 121112.

[29] D. Ludovico, P. Guardiani, A. Pistone, L. De Mari Casareto Dal Verme, D. G. Caldwell, and C. Canali, "A dual forward–backward algorithm to solve convex model predictive control for obstacle avoidance in a logistics scenario," *Electronics*, vol. 12, no. 3, p. 622, Jan. 2023.

[30] Z. Qin, H. Jing, G. Xiong, L. Chen, B. Xu, and R. Ding, "An optimal obstacle avoidance method using reinforcement learning-based decision parameterization for autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 26, no. 7, pp. 1–15, Jul. 2025.

[31] L. Hu, J. Huang, S. Hao, S. Liu, J. Lu, and B. Chen, "Finite-time switching resilient control for networked teleoperation system with time-varying delays and random DoS attacks," *IEEE Trans. Ind. Cyber-Phys. Syst.*, vol. 2, pp. 232–243, 2024.

[32] B. Jing, J. Na, H. Duan, X. Wang, and Y. Huang, "A USDE-based control for bilateral teleoperation systems with time-varying delays," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 407–417, 2025.

[33] S. Kolekar, B. Petermeijer, E. Boer, J. de Winter, and D. Abbink, "A risk field-based metric correlates with driver's perceived risk in manual and automated driving: A test-track study," *Transp. Res. C, Emerg. Technol.*, vol. 133, Dec. 2021, Art. no. 103428.

[34] S.-K. Chiu, D. Araiza-Illan, and K. Eder, "Risk-based triggering of bio-inspired self-preservation to protect robots from threats," in *Towards Autonomous Robotic Systems*. Cham, Switzerland: Springer, 2017, pp. 166–181.

[35] A. Leite, A. Pinto, and A. Matos, "A safety monitoring model for a faulty mobile robot," *Robotics*, vol. 7, no. 3, p. 32, Jun. 2018.

[36] J. Jiang, A. Astolfi, and T. Parisini, "Robust traffic wave damping via shared control," *Transp. Res. C, Emerg. Technol.*, vol. 128, Jul. 2021, Art. no. 103110.

[37] Q. Zhang, Z. Xu, Y. Wang, L. Yang, X. Song, and Z. Huang, "Predicted trajectory guidance control framework of teleoperated ground vehicles compensating for delays," *IEEE Trans. Veh. Technol.*, vol. 72, no. 9, pp. 11264–11274, Sep. 2023.

[38] T. Gruden, S. Tomažić, and G. Jakus, "Post-takeover proficiency in conditionally automated driving: Understanding stabilization time with driving and physiological signals," *Sensors*, vol. 24, no. 10, p. 3193, May 2024.

Ruoshan Wang received the B.S. degree in automation from the Department of Automation, Xiamen University, Xiamen, China, in 2023. He is currently pursuing the M.S. degree with the Department of Automation, University of Science and Technology of China (USTC), Hefei, China.

His research interests include human–machine collaboration and teleoperation.

Pengfei Li received the Ph.D. degree in control science and engineering from the University of Science and Technology of China (USTC), Hefei, China, in 2020.

He is currently an Associate Research Fellow with the School of Information Science and Technology, USTC. His research interests include human–machine hybrid control systems, model predictive control, and learning-based control.

Yun-Sheng Zhao received the B.S. degree in automation from the Department of Automation, University of Science and Technology of China (USTC), Hefei, China, in 2021, where he is currently pursuing the Ph.D. degree in control science and engineering.

His research interests include human–machine collaboration, with a focus on shared control.

Yun-Bo Zhao received the Ph.D. degree in control theory and control engineering from the University of South Wales, Cardiff, U.K., in 2008.

He is currently a Professor with the University of Science and Technology of China, Hefei, China, and the Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei. His research interests mainly focus on AI-driven control and automation, specifically, human–machine intelligence, smart manufacturing, and AI-driven networked control systems.

Yu Kang (Senior Member, IEEE) received the Ph.D. degree in control theory and control engineering from the University of Science and Technology of China (USTC), Hefei, China, in 2005.

From 2005 to 2007, he was a Post-Doctoral Fellow with the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China. He is currently a Professor with Hefei University of Technology, Hefei, and the Department of Automation, USTC. His current research interests include monitoring of vehicle emissions, adaptive/robust control, variable structure control, mobile manipulator, and Markovian jump systems.