2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Deep Learning-based Delay Compensation Framework For Teleoperated Wheeled Rovers on Soft Terrains

Ahmad Abubakar, Yahya Zweiri, Mubarak Yakubu, Ruqayya Alhammadi, Mohammed Mohiuddin, Abdel Gafoor Haddad, Jorge Dias, Lakmal Seneviratne

***Abstract*—The difficulties posed by terrain-induced slippage for wheeled rovers traversing soft terrains are critical to ensuring safe and precise mobility. While bilateral teleoperation systems offer a promising solution to this issue, the inherent network-induced delays hinder the fidelity of the closed-loop integration, potentially compromising teleoperator system controls, and resulting in poor command-tracking performance. This work introduces a new model-free predictor framework based on deep learning designed to improve prediction performance and effectively compensate for large network delays in teleoperated wheeled rovers. Our approach employs the Recurrent Neural Network (RNN) to achieve a significant improvement in modeling complexity and prediction accuracy. Particularly, our framework consists of two distinct predictors, each tailored to the forward and backward coupling variables of the teleoperated wheeled rover. Human-in-the-loop experiments were conducted to validate the effectiveness of the developed framework in compensating for the delays encountered by teleoperated wheeled rovers coupled with terrain-induced slippage. The results confirm the improved prediction accuracy of the framework. This improvement is evidenced by improved performance and transparency metrics, which lead to better command-tracking performance. A supplementary video is available at https://youtu.be/-06UGumQ0tA.**

***Index Terms*—Bilateral teleoperation, deep learning, delay compensation, wheeled rovers, longitudinal slippage.**

I. Introduction

Teleoperated wheeled rovers have become increasingly important in many high-risk environments, including military operations, space exploration, and the mining sectors, due to their ability to be remotely controlled by human operators [1]. This remote operation capability allows these rovers to perform complex tasks in challenging conditions, which is essential for ensuring human safety in several sectors [2]. Real-time situational awareness, through visual and tactile information, enhances the telepresence of the operator and the precision of rover operations [3]. This could be achieved through a "Bilateral teleoperator" [4].

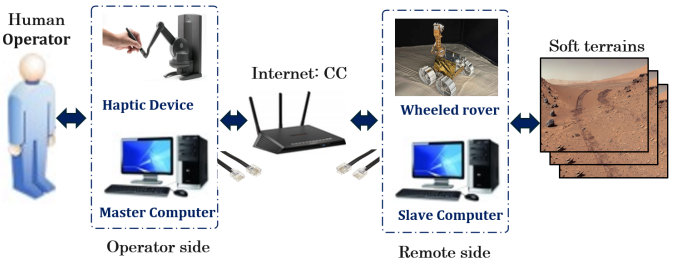

Fig. 1. Overview of bilateral teleoperation system with haptic feedback for wheeled rover traversing soft terrains.

A significant challenge encountered by teleoperated wheeled rovers during planetary exploration involves terrain traversability. Specifically, sloppy soft terrains induce longitudinal slippage as a result of wheel-terrain interactions when traversed by the wheeled rovers [4]. This slippage can substantially compromise navigation, leading to poor command-tracking performance and the possibility of tip-over [5]. To address these challenges, the induced slippage is also sent to the operator, as part of the situational awareness, to actively take corrective measures accordingly [6]. However, identifying the slippage through video feedback alone proves challenging, especially under the low-lighting conditions prevalent in space environments. In this context, integrating haptic feedback could offer a viable solution for rendering slippage awareness, with some researchers advocating for discrete haptic feedback [7], while others recommend the use of continuous haptic feedback [5]. This provides operators with a more intuitive means of detecting and responding to navigation issues encountered by teleoperated wheeled rovers [9]. The overview of the teleoperated wheeled rovers with terrain-induced slippage awareness is illustrated in Fig .1. The operator sends motion commands and receives slippage-based haptic feedback.

Nevertheless, delays in the network within the communication link between the operator and the rover can markedly impact teleoperation performance by reducing the fidelity of the closed-loop integration essential for quick responses to slippage detection [3]. Such delays can cause a decline

This research publication was funded by the Khalifa University Center for Autonomous Robotic Systems (KUCARS) under Award No. RC1-2018-KUCARS.

A. Abubakar, M. Yakubu, R. Alhammadi, M. Mohiuddin, A. Haddad, J. Dias, and L. Seneviratne are with the Khalifa University Center for Autonomous and Robotic Systems (KUCARS). Y. Zweiri is associated with the Advanced Research and Innovation Center (ARIC) and the Department of Aerospace Engineering. M. Mohiuddin is also with the School of Engineering and Technology, University of New South Wales, Canberra, Australia. J. Dias is also with the Department of Electrical Engineering. L. Seneviratne is also with the Department of Mechanical Engineering at Khalifa University, Abu Dhabi, United Arab Emirates (Corresponding author's email: 100059792@ku.ac.ae).

in performance and transparency measures and might lead to system instability, especially when the channel becomes unstable, despite minimal time delays [10]. In this study, we introduce a new control-based delay compensation framework that incorporates deep learning techniques, to enhance the fidelity of closed-loop integration.

A. Related work

Numerous studies have been dedicated to addressing the issue of network delays in bilateral teleoperator systems, to ensure high fidelity of closed-loop integration. These research efforts have investigated a variety of methodologies, including the Passivity Approach of Wave Variable (WV) and Time Domain Passivity Approach (TDPA) [2], [4]. A work in [10] has implemented the WV technique in combination with a derivative controller to stabilize and improve the performance of teleoperated wheeled rovers. However, it introduced a conservative nature, leading to a severe trade-off between performance and transparency within the system. Another study [11] implements the TDPA approach to compensate for large delays, accommodating varying delay lengths of rovers navigating dynamic environments. Yet, the approach faces difficulties in accurately estimating energy flow and dynamically adjusting to delays in real-time operations. In addition to WV and TDPA, other compensation techniques include model-based control approaches [12], [13], model-free control approaches [14], and hybrids of the two approaches [15].

The original model-free predictor framework proposed by Yingshi Zheng et al. [14] introduced a single regularization parameter for compensating delays in general closed network systems. Over time, this framework has seen various adaptations across multiple applications, demonstrating a notable capability in mitigating the effects of delays through promising prediction performance. Despite these successes, limitations in handling larger delays and the inherent uncertainty of real network delays were identified [16], indicating a need for further refinement. Subsequent research efforts sought to enhance the framework's adaptability to uncertain network delays. A study [17] refined the original model by integrating an adaptively varying parameter that responds to fluctuations in delay, aiming to offer a more dynamic and responsive solution. While this modification improved the framework's robustness, it introduced a new challenge of oscillatory error gain, which compromised the overall effectiveness of the predictor [4]. To overcome these challenges, a recent study [18] further modified the model-free predictor by adding a degree of freedom. This approach was designed as a potential remedy for the uncertainties associated with network delays and the issue of increased oscillatory error gain. Nevertheless, in our recent work, we implemented the predictor to compensate for large delays in [19], however, our study revealed that significant prediction errors remain in compensating for large delays. In our effort to overcome this limitation, we employed deep neural networks modeling technique, to develop predictors for compensating large delays, like those observed in long-distance Earth-to-Lunar networks [20].

B. Contribution

The modified first-order time-delay (FOD) predictor in [18] demonstrates significant prediction error for large delay compensation. However, the application of nonlinear modeling techniques, such as neural networks, to reduce this prediction error has not been previously explored. This article addresses this gap through the following original contributions:

1) We first establish a real-time test platform that can replicate real-world longitudinal slippage of the wheeled rover on soft terrains, to facilitate data generation.
2) We then introduce a deep learning-based (DL) model-free predictor framework designed to enhance prediction performance and effectively compensate for large delays.
3) Lastly, we validate the developed framework for its effectiveness in compensating for the large network delays observed in teleoperated wheeled rovers combined with terrain-induced slippage. Also compare with the state-of-the-art FOD predictor [18].

II. Preliminaries

In this section, we discuss the proposed bilateral teleoperation system depicted in Fig. 2, detailing its components that include the master robot, the slave robot, the remote environment, the communication channel, and the proposed predictor framework. The operator sends motion commands from the master robot (haptic device) through the communication channel to the forward predictor, which compensates for uplink delays before executing on the slave robot (wheeled rover). Conversely, terrain-induced slippage is computed from the remote environment (soft terrains) and sent back through the communication channel to the backward predictor, which compensates for downlink delays, and is then rendered to the operator as haptic feedback.

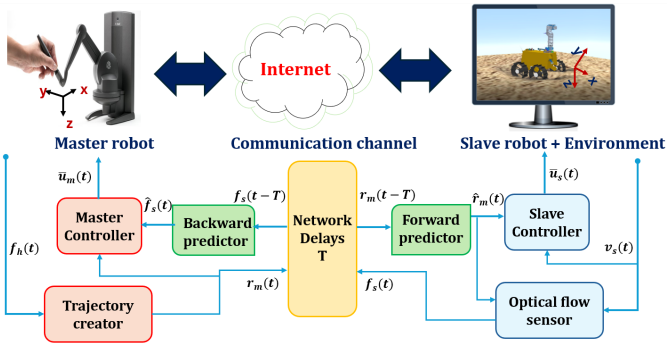

Fig. 2. Proposed bilateral teleoperation system with two predictors, forward and backward, to compensate for up-link and down-link delays, respectively.

A. Master robots

In our study, we employed the Phantom haptic device as the master robot (haptic interface), leveraging its Two-Degree Of Freedom (DOF)—joint x and joint y—to encode the non-holonomic motion commands, which include linear and angular velocities of wheeled rover, demonstrated in Fig. 2. This method is a common approach as detailed in [8]. The approximated linear dynamics equation of the master robot is presented as in (1), which is derived in [9].

$$M_m\ddot{q}_m + C_m\dot{q}_m = u_m + f_h \tag{1}$$

M_m represent the mass matrix of the robot, C_m denote the centrifugal matrix, $q_m = \begin{bmatrix} q_{mx} & q_{my} \end{bmatrix}^T$ is the joint position variable, $u_m = \begin{bmatrix} u_{mx} & u_{my} \end{bmatrix}^T$ is the joint control force, and $f_h = \begin{bmatrix} f_{hx} & f_{hy} \end{bmatrix}^T$ represents the external force exerted by human operator.

To tackle the possible instability issue of the master robot, a new dynamic variable was introduced, $r_m = \lambda\dot{q}_m + q_m \quad (0 < \lambda < 1)$. Consequently, the robot controller is formulated as $u_m = \bar{u}_m + u_m^*$, where $\bar{u}_m$ is the teleoperator controller, and u_m^* is the local controller, given by $u_{mi}^* = B_{Lv}\dot{q}_m$, as derived in [5]. The modified linear dynamics yield a first-order system as in (2).

$$\bar{M}_m\dot{r}_m + \bar{C}_m r_m = \bar{u}_m + f_h \tag{2}$$

where $\bar{M}_m = M_m/\lambda$ and $\bar{C}_m = C_m/\lambda - B_{Lv}$ are the equivalent mass and centrifugal, respectively. $r_m = \begin{bmatrix} r_{mx} & r_{my} \end{bmatrix}^T$ is the new joints velocity variable.

B. Slave robot

The slave robot studied in this paper is modeled after the Boogie drive of the UAE Rashid rover, as depicted in Fig. 2. Its kinematic model dictates the relationship between wheel speeds and the rover's velocity, while the dynamic model relates the applied torques to the rover's acceleration. In this paper, however, we will concentrate exclusively on the kinematic model, as our applications involve low-speed maneuvers. We assume, in our rover mobility concept, that the velocities of the front and rear wheels are approximately the same when traversing flat terrains. This assumption encapsulates the kinematic model of differential drive denoted by (3) in the robot frame as obtained from [5].

$$\begin{bmatrix} v_{sx} \\ v_{sy} \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 \\ -1/(2b) & 1/(2b) \end{bmatrix} \begin{bmatrix} v_r \\ v_l \end{bmatrix} = \begin{bmatrix} u_{sx} \\ u_{sy} \end{bmatrix} \tag{3}$$

where $v_s = \begin{bmatrix} v_{sx} & v_{sy} \end{bmatrix}^T$ represents the measured velocity of the rover, the longitudinal and angular velocity, respectively. $u_s = \begin{bmatrix} u_{sx} & u_{sy} \end{bmatrix}^T$ represents the desired velocities of the rover, the longitudinal and angular velocity, respectively. b is the distance between the right and the left wheels, v_r and v_l are the right and the left wheel linear velocities, respectively. The transformation matrix, denoted as $E(b)$, is detailed in (4).

$$E(b) = \begin{bmatrix} 1/2 & 1/2 \\ -1/(2b) & 1/(2b) \end{bmatrix} \tag{4}$$

Equation (3) is only valid on hard terrains where pure rolling occurs. On soft terrains, such as loose soil, the relationship between each wheel's linear velocity and the product of its angular velocity and radius might not be maintained due to longitudinal slippage from wheel-terrain interactions [21]. We apply (5) to quantify the degree of wheel slippage [1]. Consequently, this leads to a substantial alteration from the original kinematic model outlined in (3) to a modified model that accounts for wheel slippage, as derived in [10] and detailed in (6).

$$s_r = \frac{v_{rd} - v_r}{v_r}, \; s_l = \frac{v_{ld} - v_l}{v_l} \tag{5}$$

$$\begin{bmatrix} v_{sx} \\ v_{sy} \end{bmatrix} = \begin{bmatrix} u_{sx} \\ u_{sy} \end{bmatrix} - E(b) \begin{bmatrix} v_{rd} - v_r \\ v_{ld} - v_l \end{bmatrix} \tag{6}$$

Where v_{rd} and v_{ld} are the desired linear velocities of the right and left wheels, respectively. $s_r(t)$ and $s_l(t)$ are the measured wheel slip on the right and left, respectively. In addition, we focus on the case of $v_{rd} \geq v_r$ in (5), corresponding to $s_r \geq 0$, to avoid non-passivity issues due to negative slippage [5]. Thus, based on (5) and (6), the slippage-induced loss of velocity on the rover can be defined as in (7).

$$\begin{bmatrix} f_{sx} \\ f_{sy} \end{bmatrix} = E(b) \begin{bmatrix} v_{rd} - v_r \\ v_{ld} - v_l \end{bmatrix} = E(b) \begin{bmatrix} s_r v_r \\ s_l v_l \end{bmatrix} \tag{7}$$

Where the $f_s = \begin{bmatrix} f_{sx} & f_{sy} \end{bmatrix}^T$ represents the environmental force (haptic feedback), which is assumed to be the friction force applied by the soft terrains on the wheel.

To improve the poor command-tracking performance of the kinematic model in (6), an acceleration level controller is necessary to activate the embedded velocity motor controller. Based on the work in [5], we can modify the kinematic model (6), which is only coupled with the longitudinal slippage, with an acceleration level controller expressed by (8).

$$\begin{bmatrix} \dot{v}_{sx} \\ \dot{v}_{sy} \end{bmatrix} = \begin{bmatrix} \bar{u}_{sx} \\ \bar{u}_{sy} \end{bmatrix} - \begin{bmatrix} \dot{f}_{sx} \\ \dot{f}_{sy} \end{bmatrix} \tag{8}$$

Where $\bar{u}_s = \begin{bmatrix} \bar{u}_{sx} & \bar{u}_{sy} \end{bmatrix}^T$ is the new control input. Refer to [5], [10] for more details.

C. Bilateral teleoperator controls

The bilateral teleoperator controllers are designed to facilitate velocity-velocity mapping coordination between the master robot (haptic device) and slave robot (wheeled rover), as depicted in Fig. 2. The control laws for the master and slave presented in (9) and (11) correspond to cases with delays but without the use of predictors, while the control laws in (10) and (12) incorporate delays with our proposed predictors. The former results in poor performance and transparency (indicating low-fidelity closed-loop integration) in the presence of large delays T. Conversely, the latter aims to achieve superior performance and transparency (indicating high-fidelity closed-loop integration), even with

large delays T. The control schemes utilize basic proportional and derivative controllers, as described below:

Master Controller: To achieve perfect coordination between (r_{mx}, v_{sx}) and (r_{my}, v_{sy}) on the operator side. The joint control input u_m in (2) is computed based on the predicted feedback $(\hat{f}_{sx}(t),\ \hat{f}_{sy}(t))$:

$$\begin{bmatrix} \bar{u}_{mx}(t) \\ \bar{u}_{my}(t) \end{bmatrix} = \begin{bmatrix} -k_{mx}\ f_{sx}(t-T) - d_{mx}\ r_{mx}(t) \\ -k_{my}\ f_{sy}(t-T) - d_{my}\ r_{my}(t) \end{bmatrix} \quad (9)$$

$$\begin{bmatrix} \bar{u}_{mx}(t) \\ \bar{u}_{my}(t) \end{bmatrix} = \begin{bmatrix} -k_{mx}\ \hat{f}_{sx}(t) \\ -k_{my}\ \hat{f}_{sy}(t) \end{bmatrix} \quad (10)$$

Slave Controller: To achieve perfect coordination between (r_{mx}, v_{sx}) and (r_{my}, v_{sy}) at the remote side. The control input $\bar{u}_s$ in (6) is computed based on predicted motion commands $(\hat{r}_{mx}(t),\ \hat{r}_{my}(t))$:

$$\begin{bmatrix} \bar{u}_{sx}(t) \\ \bar{u}_{sy}(t) \end{bmatrix} = \begin{bmatrix} k_{sx}\ r_{mx}(t-T) - d_{sy}\ v_{sy}(t) \\ k_{sy}\ r_{my}(t-T) - d_{sy}\ v_{sy}(t) \end{bmatrix} \quad (11)$$

$$\begin{bmatrix} \bar{u}_{sx}(t) \\ \bar{u}_{sy}(t) \end{bmatrix} = \begin{bmatrix} k_{sx}\ \hat{r}_{mx}(t) \\ k_{sy}\ \hat{r}_{my}(t) \end{bmatrix} \quad (12)$$

To satisfy the absolute stability condition of the closed-loop integration, the following conditions must be met [19]: $d_m \geq k_m,\ d_s \geq 0,\ k_m, k_s > 0$.

III. Framework Design

The DL framework comprises two distinct predictors, each Primarily structured around Long Short-Term Memory (LSTM) networks. The forward predictor is tailored for predicting the motion commands, while the backward predictor is tailored for predicting the force feedback. The input-output architecture of each predictor is depicted in Fig. 3, where x is any variable of interest among the coupling variables.

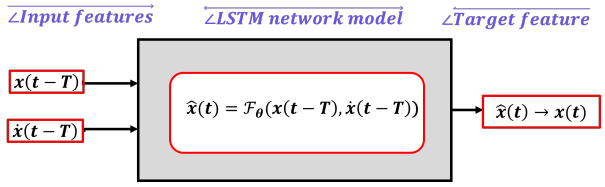

Fig. 3. Illustration of the mapping relationship between delayed and undelayed variables

A. Data generation

The dataset was created through human-in-the-loop experiments with a teleoperated wheeled rover navigating soft terrains, as elaborated in Section IV. The collection process began with the comprehensive recording of coupling variables that includes the motion command data, $r_m(t)$, and force feedback signals, $f_s(t)$ from the track-following task. The dataset comprises three key variables for each of the coupling variables including the undelayed actual value $x(t)$, the delayed actual value $x(t-T)$, and the time derivative of the delayed actual value $\dot{x}(t-T)$. These variables are presumed to encapsulate the critical information needed to accurately model the complexities inherent in delayed system.

The training dataset The training dataset was gathered at a sampling rate of 100 Hz over a total simulation time of 300 s (yielding about 30,000 data points). The testing datasets maintain the same sampling rate as the training set but with a shorter simulation duration of 100 s. Moreover, the training dataset is structured with the input features $x(t-T)$, and $\dot{x}(t-T)$; and the target output feature $x(t)$, representing the actual values that the model is designed to learn and predict. Furthermore, the input features underwent preprocessing, which involved normalizing and scaling [22].

B. Network Architecture

The design of the LSTM network is tailored to handle temporal data marked by complex nonlinearities and temporal dependencies. The network consists of an input layer, a fully connected layer, an activation layer, multiple LSTM layers, and an output layer as shown in Fig. 4. Each LSTM layer is specifically captured to extract various levels of detail from the temporal data.

The input layer is tasked with receiving the input sequence, which consists of a fixed length feature vector, represented as $(T_1, T_2, \ldots, T_L)$, where each T_i is defined as $T = [x(t-T), \dot{x}(t-T)]$. The distinct feature of the LSTM layer lies in its feedback loop, in which the output from each cell is repeatedly looped back into the same cell during the next time step (Fig. 4). The output layer produces single-step predicted values, denoted by $\hat{Y}_L$, where $\hat{Y}_i$ is defined as $\hat{Y} \longrightarrow Y = \hat{x}(t) \longrightarrow x(t)$.

The two predictors adhere to a consistent architectural framework, yet they exhibit variations in the number of layers and units.

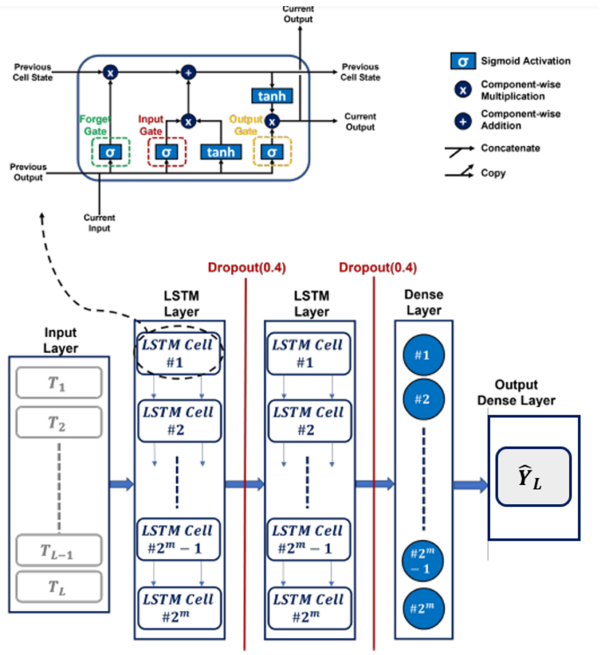

Fig. 4. LSTM architecture for the proposed DL predictor consists of the following: the fixed-length input features L, the hidden units in both dense and LSTM layer m, the number of LSTM layers k, and a single-step target output at L.

C. Optimization Process

In our regression tasks, a Mean Square Error (MSE) loss function ($\mathcal{L}_{MSE}$) is the most commonly used for optimization cost as given by (13).

$$\mathcal{L}_{MSE}(Y, \hat{Y}) = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2 \tag{13}$$

where N represents the total number of samples in a run. The loss function is chosen because it effectively maximizes the likelihood when prediction errors follow a normal (Gaussian) distribution. This assumption of Gaussian errors is a good fit for many real-world scenarios, making MSE a suitable choice for these problems. Given that $\hat{Y}_i = \mathcal{F}(T;\theta)$, with θ as the network parameters, the optimization problem to be solved is expressed in (14).

$$\theta^* = \min \mathcal{L}_{MSE}(Y, \mathcal{F}(T;\theta))) \tag{14}$$

Upon solving (14), it is deduced that the necessary condition for optimality is the differentiability of the loss function with respect to the network parameters.

D. Network training

The training protocol for each predictor begins with the input of a sequence of 100 prior data points, representing 1 s of historical data. As for the architecture of the two predictors, the optimal network hyperparameters (LSTMDepth, HiddenUnits) are identified through Bayesian optimization. This approach systematically explores the hyperparameter space to find the set that maximizes performance and minimizes error. Specifically, for the forward predictor, (LSTMDepth = 2, HiddenUnits = 157) with a vector of output $\hat{r}$, whereas for the backward predictor, (LSTMDepth = 3, HiddenUnits = 246) with a vector of output $\hat{f}_s$.

Using a form of stochastic gradient descent (Adam optimizer), the parameters θ updates are computed based on the loss gradient using back-propagation. This involves an iterative process where a forward pass computes: $\hat{Y}_i = \mathcal{F}(T;\theta)$ and a backward pass computes, $\frac{\partial \mathcal{L}_{MSE}(Y,\mathcal{F}(T;\theta))}{\partial \theta}$. The network parameters, θ are updated as $\Delta\theta = -\eta\nabla\theta$, where η represents a learning rate, which may vary over time. Upon identifying the optimal parameters that minimize loss, the network successfully learns a model, $\mathcal{F}_\theta$, accurately representing the data.

E. Prediction performance

After training each predictor, we validate the models using the testing dataset to demonstrate their generalization ability and effectiveness in extrapolating insights from new, unseen data. Consequently, the performance of the designed predictors is quantified by Root Mean Square Error (RMSE), which is mathematically defined in (15).

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2} \tag{15}$$

where N represents the total number of samples within the dataset. For the forward predictor, the validation RMSE is 25.76×10^{-3} and the test RMSE is 30.41×10^{-3}. Whereas for the backward predictor, the validation RMSE is 47.22×10^{-3} and the test RMSE is 69.71×10^{-3}. Analysis of these error values reveals that the generalization error for the forward predictor (18.05%) is consistently lower than that for the backward predictor (47.63%). This discrepancy could indicate that the predictors are more effective at predicting motion commands rather than force feedback.

IV. Experiment

Human-in-the-loop experiments as in [10], [16], [19] were conducted on a test platform to assess the performance of a teleoperated wheeled rover on soft terrain, both with and without the DL predictor framework, under large delays.

A. Test Platform

A real-time test platform has been established, combining a Torch-X haptic device with MATLAB/Simulink and the Vortex Studio simulator to produce real-world longitudinal slippage and tipping-over of the wheeled rover on soft terrains, as depicted in Fig. 5. Vortex Studio was selected based on prior research [5], [21], which confirmed its high fidelity and validated its ability to simulate wheel-terrain interaction dynamics experimentally.

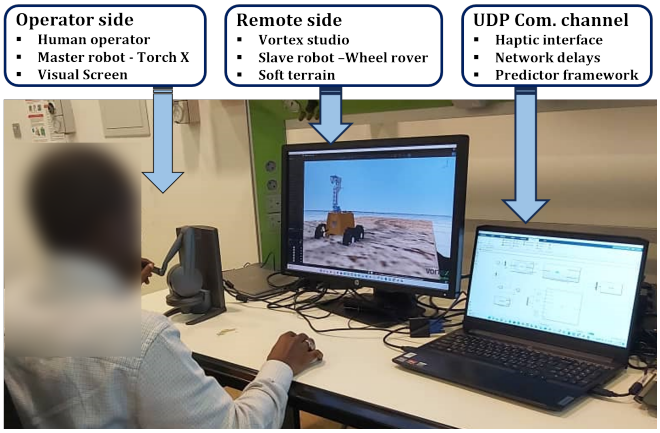

Fig. 5. Human-in-the-loop test platform for a wheeled rover on soft terrain with slippage awareness.

The operator side utilizes a Torch-X haptic device (master robot) for motion command generation and force feedback perception. This device is integrated with the simulator through Simulink, serving as a control interface. A first-order low-pass filter, set with a cutoff frequency of 0.8 Hz, was used to reduce high-frequency fluctuations in the variables. On the other hand, the remote side receives motion commands from the operator side to operate the wheeled rover (slave robot). This rover's simulation, conducted in Vortex Studio, incorporates a wheel-terrain interaction model developed in [21], to provide the real-world operation of rovers on soft terrains.

B. Experimental Design

A roadmap to a test track was developed, taking inspiration from lunar landscapes for the generated test platform, as illustrated in Fig. 6. This complex track, depicted in gray, is crafted to mimic soft terrain (loose soil), starting from the green mark and ending at the red mark. It includes straight paths, right turns, and left turns, to represent a real navigation map of planetary rovers [19]. The track 10 m in length and 2 m in width, and 6° in slope angle. The track is designed to exhibit position-dependent variable slippage, characterized by varied internal friction angle $\varphi(x)$, which changes as a function of the track's position, x [10]. The area outside the track, depicted in brown, acts as a visual cue, affording operators an awareness of their proximity to upcoming turns. Moreover, the network delays from the operator side to the remote side are assumed to be 1.5 s [20]. To simulate this, we implement variable delays obtained on a real network of $1.3 \sim \mathcal{U}(-0.2, 0.2)$s using the User Datagram Protocol (UDP) to emulate both forward and backward delays.

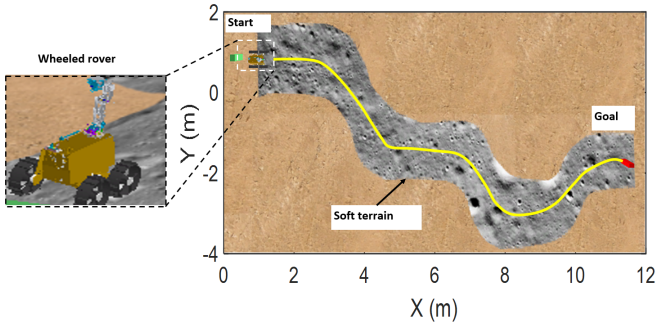

Fig. 6. Designated soft terrain tracks and non-track zone in the virtual environment, with a slope angle of 6°.

The operator is supposed to execute a track-following task under three distinct conditions: (i) the ideal case, without any delays, establishes a baseline for performance; (ii) the delayed case, incorporating round-trip delays without a predictor framework, to evaluate the impact of delays on task execution; and (iii) the predicted case, with the same delays and the predictor framework, to evaluate performance improvement due to delay compensation. Hence, the operator is subjected to operating the rover; to complete the track as fast as possible, while following the track's center closely, and avoiding rover tip-over; under the three conditions.

C. Evaluation Metrics

The experiment, delineating operations of all conditions, was subjected to individual assessments employing common teleoperation metrics [5], [10]: namely, performance, and transparency. The performance metric - velocity tracking, is quantified by the discrepancies between the commanded motion r_m and the rover measured states v_s. In contrast, transparency, force feedback tracking, is quantified by the deviation between the force perceived by the operator f_h and the environmental force-induced slippage f_s. These metrics serve as direct indicators of the closed-loop system's fidelity; smaller metric values correspond to higher fidelity, while larger values indicate reduced fidelity [18]. The mathematical description of these metrics is depicted in (16).

$$\text{Performance} := \|r_m - v_s\|_2, \text{Transparency} := \|f_h - f_s\|_2 \tag{16}$$

V. Result and Discussion

We compare the delay compensation performance between the developed DL predictor framework and FOD predictor [19], with benchmarking with the ideal and delayed case. The bilateral teleoperator controller gains for the master and the slave robots were set to $k_m = 10$, $d_m = 10.5$, $k_s = 10$, and $d_s = 1.0$, respectively [10]. Fig. 7 illustrates the path traveled by the teleoperated wheeled rover, depicting the ideal, predicted, and delayed cases.

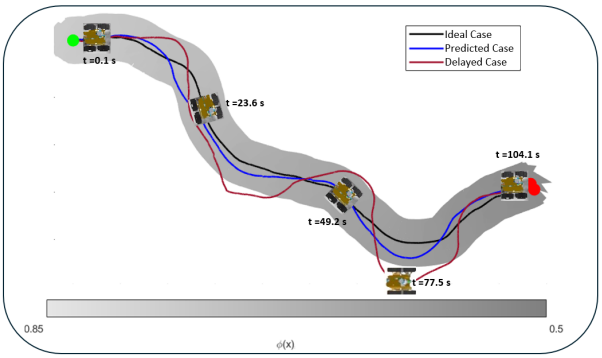

Fig. 7. Execution reconstruction, highlighting the three distinct cases.

We first validate the terrain-induced slippage experienced by the wheeled rover against the specified terrain properties to ensure its realism. In our analysis, we compare the measured slippage of the right s_r and the left s_l wheels of the rover, as observed on the test platform, with the slippage predicted by a theoretical model. This model is founded on an empirical relationship with the specified terrain properties, as detailed in [1], [9], and has been experimentally validated. Two trials were conducted to ensure fair validation. The performance comparison of the slippage for the trials is shown in Fig. 8. A closed alignment was observed between both the measured slippage s_r and s_r and the theoretically predicted slippage s_{thr}, in both instances. Slippage increases due to changes in terrain parameter $\varphi(x)$. This concordance underscores the repeatability of the real-world behavior of the rover when it encounters various terrain-induced resistances.

Figure 9 illustrates the performance in the ideal case of the track-following task. The operator successfully offsets the inherent discrepancy between the rover's states and the motion commands caused by longitudinal slippage. Consequently, command-tracking performance approaches perfection, as demonstrated by velocity-tracking and force-feedback tracking metrics, aligning with findings from [8]. However, upon adding the delays in the ideal case -delayed case,

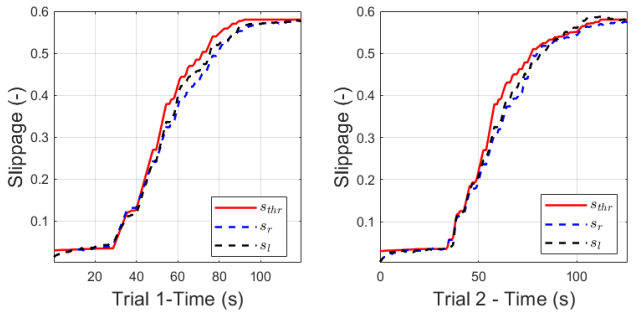

Fig. 8. Wheel-terrain slippage for the rover traversing the soft terrain tracks.

performance markedly degraded, as depicted in Fig. 10. This deterioration is due to reduced closed-loop integration fidelity, where delayed force feedback leads to delayed operator responses. Specifically, at delays of 1.37 and 1.5 seconds at joints 1 and 2, respectively.

Upon recognizing the performance deterioration induced by delays, we implemented our DL predictor framework and then compared it with the FOD framework in [18] to compensate for the delayed—predicted case. This implementation significantly improved the overall teleoperation performance, as demonstrated by the velocity and force feedback tracking metrics presented in Fig. 11 and Fig. 12, respectively. Both frameworks effectively compensate for some amount of delays between the commands and the actual rovers's state, as well as the environmental force and the force perceived by the operator, evident by the pointed delays. However, our DL framework outperformed the FOD framework in [18] in delay compensation, encountering only a maximum delay of 0.69 s compared to their 1.12 s, against the original delay of 1.5 s. This improvement is attributed to the ability of our deep learning framework to learn nonlinear and temporal dynamics behavior. As a result, the command-tracking performance neared the stable levels typical of the ideal case, suggesting a close recovery of high-fidelity closed-loop integration.

The bar graph in Fig. 13 compares the performance and transparency metrics across the above-discussed four cases. While the ideal case still performs the best, our DL-predicted case significantly reduces this gap in all the metrics compared with the FOD-predicted and delayed.

VI. Conclusion

In this paper, a deep learning-based delay compensation framework is introduced for a teleoperated wheeled rover subjected to terrain-induced slippage. This framework aimed to improve the prediction performance of the delayed systems by learning nonlinear and temporal dynamics behavior. Our findings demonstrate that the developed predictor framework surpasses the performance of the existing framework in compensating for large delays, with significant prediction performance across the various experiments of the teleoperated wheeled rover. This enhancement leads to a more stable closed-loop system with better command-tracking performance. However, our developed framework lacks generalization to unknown data from different track patterns, thus our future expansion may explore a hybrid of deep learning and model-based approaches for lower prediction error and better generalization in large delay compensation.

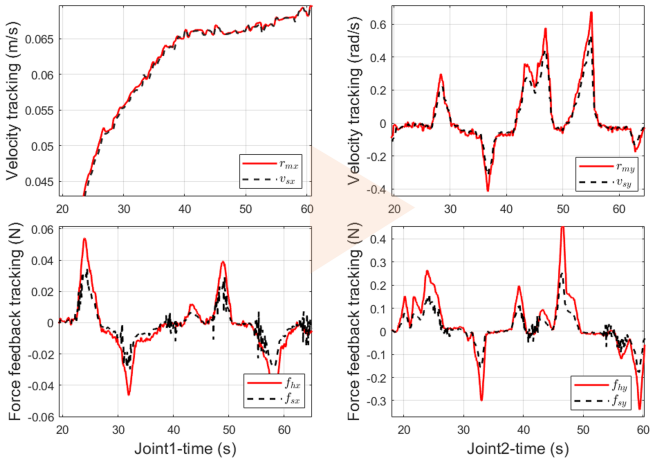

Fig. 9. Zoomed ideal case: Comparison of the performance and transparency of motion commands (top) and force feedback (bottom). Achieved perfect command tracking performance.

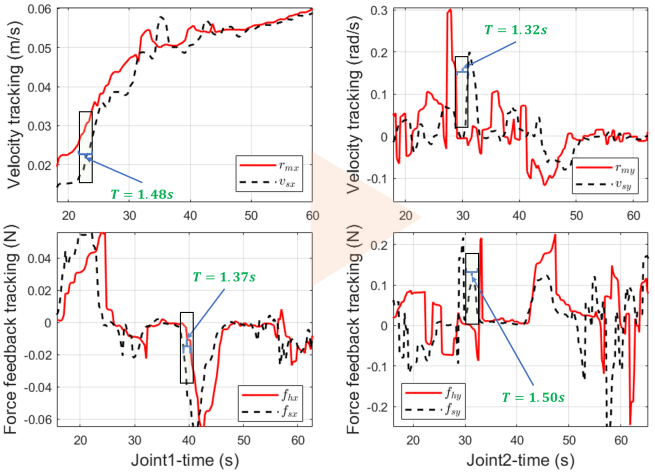

Fig. 10. Zoomed delayed case: Comparison of the performance and transparency of motion commands (top) and force feedback (bottom). This leads to poor command tracking performance due to the conservative derivative-term.

References

[1] R. Alhammadi, Y. Zweiri, A. Abubakar, M. Yakubu, L. Abuassi and L. Seneviratne, "Event-based Slip Estimation Framework for Space Rovers Traversing Soft Terrains," in *IEEE Access*, Access-2024-26979, July 2024.

[2] P. Arm et al., "Scientific exploration of challenging planetary analog environments with a team of legged robots," *Sci. Robot.*, vol. 8, no. 80, p. eade9548, Jul. 2023.

[3] S. N. F. Nahri, S. Du, and B. J. Van Wyk, "A Review on Haptic Bilateral Teleoperation Systems," *J. Intell. Robot. Syst. Theory Appl.*, vol. 104, no. 1, 2022.

[4] S. Opiyo, J. Zhou, E. Mwangi, W. Kai, and I. Sunusi, "A Review on Teleoperation of Mobile Ground Robots: Architecture and Situation Awareness," *Int. J. Control. Autom. Syst.*, vol. 19, no. 3, pp. 1384–1407, 2021.

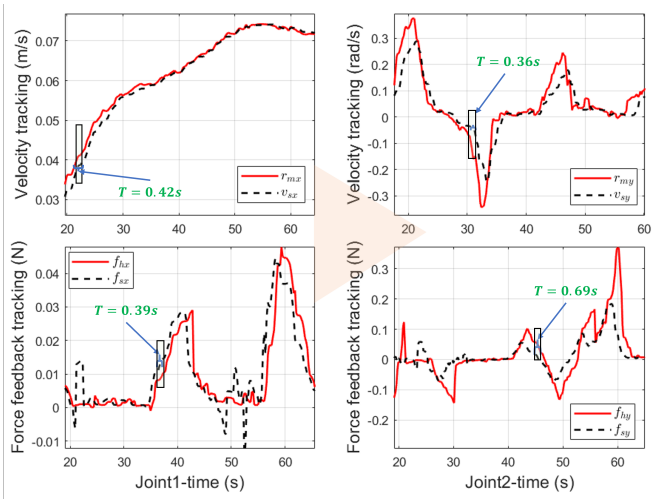

Fig. 11. Zoomed predicted case - our DL framework: Comparison of the performance and transparency of motion commands (top) and force feedback (bottom). Compensate delays by 78.2%. Thus, leads to better command tracking performance than the delayed case.

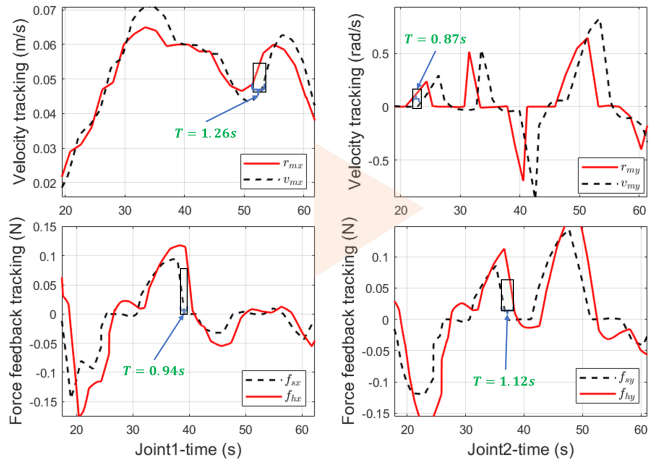

Fig. 12. Zoomed predicted case - FOD framework [19]: Comparison of the performance and transparency of motion commands (top) and force feedback (bottom). Compensate delays by only 33.6%. While slightly better than the delayed case, remains inferior to that of our predicted framework in command tracking performance.

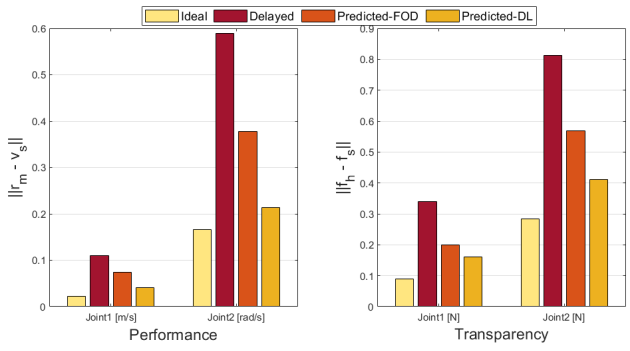

Fig. 13. L2-norm error comparison of the velocity and force feedback tracking metrics among all the cases.

[5] W. Li, N. Yang, J. Wang, and Z. Deng, "Teleoperation of wheeled mobile robots subject to longitudinal slipping and lateral sliding by time-domain passivity controller." *Mechatronics.*, 81, 2022.

[6] A. Abubakar, K. I. Dahiru, S. H. Sulaiman, and H. Mustapha, "Robust polytopic LPV based Adaptive Cruise Control design for Autonomous Vehicle System," *International Journal of Scientific and Engineering Research*, vol. 10, pp. 1367–1373, 2019.

[7] R. Luz, J. Corujeira, J. L. Silva and R. Ventura, "Traction Awareness Through Haptic Feedback for the Teleoperation of UGVs," *27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Nanjing, China, pp. 313-31, 2018.

[8] W. Li et al., "Semi-autonomous bilateral teleoperation of six-wheeled mobile robot on soft terrains," *Mech. Syst. Signal Process.*, vol. 133, p. 106234, 2019.

[9] W. Li, J. Guo, L. Ding, J. Wang, H. Gao, and Z. Deng, "Teleoperation of Wheeled Mobile Robot With Dynamic Longitudinal Slippage," *IEEE Trans. Control Syst. Technol.*, vol. 31, no. 1, pp. 99-113, 2023.

[10] W. Li, L. Ding, H. Gao, and M. Tavakoli, "Haptic Tele-Driving of Wheeled Mobile Robots under Nonideal Wheel Rolling, Kinematic Control and Communication Time Delay," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 50, no. 1, pp. 336-347, 2020.

[11] H. V. Quang, I. Farkhatdinov and J. -H. Ryu, "Passivity of delayed bilateral teleoperation of mobile robots with ambiguous causalities: Time Domain Passivity Approach," *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura-Algarve, Portugal, 2012, pp. 2635-2640.

[12] A. Abubakar, I. K. Dahiru, S. H. Sulaiman and A. B. Kunya, "Robust H-Infinity Control for Magnetic Levitation System," in *2019 2nd International Conference of the IEEE Nigeria Computer Chapter (NigeriaComputConf)*, Zaria, Nigeria, 2019, pp. 1-6.

[13] A. Abubakar, M. B. Mohiuddin, O. A. Hay, M. Yakubu and R. Alhammadi, "Robust Control for Autonomous Vehicle Lateral Dynamics: A Comparative Study of Gain-Schedule LPV and Non-linear MPC*," *8th International Conference on Robotics, Control and Automation (ICRCA)*, Shanghai, China, 2024, pp. 311-317.

[14] Y. Zheng, M. J. Brudnak, P. Jayakumar, J. L. Stein, and T. Ersal, "A Predictor-Based Framework for Delay Compensation in Networked Closed-Loop Systems," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 5, pp. 2482-2493, 2018.

[15] Y. Zheng, M. J. Brudnak, P. Jayakumar, J. L. Stein, and T. Ersal, "A Delay Compensation Framework for Predicting Heading in Teleoperated Ground Vehicles," *IEEE/ASME Trans. Mechatronics*, vol. 24, no. 5, pp. 2365-2376, 2019.

[16] 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. Human-Machine Syst.*, vol. 50, no. 6, pp. 561–572, 2020.

[17] N. Sridhar, R. Lima, U. Rai, V. Vakharia, K. Das, and B. P., "Transparency Enhancement in Teleoperation: An Improved Model-Free Predictor for Varying Network Delay in Telerobotic Application," in *2021 European Control Conference*, Delft, Netherlands, 2021.

[18] S. Guo, Y. Liu, Y. Zheng, and T. Ersal, "A Delay Compensation Framework for Connected Testbeds," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 52, no. 7, pp. 4163-4176, 2022.

[19] A. Abubakar, Y. Zweiri, R. Alhammadi, M. B. Mohiuddin, M. Yakubu and L. Seneviratne, "Predictor-Based Control for Delay Compensation in Bilateral Teleoperation of Wheeled Rovers on Soft Terrains," in *IEEE Access*, vol. 12, pp. 111593-111610, August 2024.

[20] S. Timman, M. Landgraf, C. Haskamp, S. Lizy-destrez, and F. Dehais, "Effect of time-delay on lunar sampling tele-operations: Evidence from cardiac, ocular and behavioral measures," *Appl. Ergon.*, vol. 107, no. December 2020, p. 103910, 2023.

[21] A. Abubakar, R. Alhammadi, Y. Zweiri, and L. Seneviratne, "Advance Simulation Method for Wheel-Terrain Interactions of Space Rovers: A Case Study on the UAE Rashid Rover," in *21st International Conference on Advanced Robotics*, no. December, pp. 526–532, 2023 Abu Dhabi, UAE, 2023.

[22] Y. Zou, L. Ding, H. Zhang, T. Zhu, and L. Wu, "Vehicle Acceleration Prediction Based on Machine Learning Models and Driving Behavior Analysis," *MDPI Appl. Sci.*, vol. 2022.