symmetry

MDPI

Article

Digital-Twin-Enabled Synchronous Teleoperation of Robotic Arms Using BiLSTM-KF Hybrid Model for Time Delay Compensation

Mengmeng Sun [1], Youquan Liu [1,*], Zhengxiong Liu [2] and Panfeng Huang [2]

[1] School of Information Engineering, Chang'an University, Xi'an 710064, China
[2] School of Astronautics, Northwestern Polytechnical University, Xi'an 710129, China
* Correspondence: youquan@chd.edu.cn; Tel.: +86-18792866033

Abstract

Communication delay in digital-twin teleoperation causes temporal mismatch between local virtual arm motion feedback and remote physical arm execution, particularly under time-varying public-network conditions. This study proposes a prediction-informed synchronous framework that combines one-step-ahead Bidirectional Long Short-Term Memory (BiLSTM) delay prediction with recursive Kalman Filter (KF) correction. Historical delay measurements are used to predict the subsequent communication delay, and the corrected estimate is incorporated into a feedforward speed-compensation law for the local virtual arm without modifying the low-level controller of the remote physical arm. The framework therefore adopts corrected delay estimation directly to virtual-arm motion adaptation within the digital-twin teleoperation system. On the evaluating test sequence, the BiLSTM-KF model achieved an RMSE of 5.6498 ms, an MAE of 3.5455 ms, and an MAPE of 7.84%. In teleoperation experiments, BiLSTM-KF-based compensation yielded an average normalized position deviation rate of 2.427 ± 0.569%, corresponding to reductions of 53.63% relative to no compensation, 46.42% relative to historical-mean delay compensation and 35.83% relative to LSTM-based compensation. These results indicate that prediction-informed virtual-side adaptation can reduce virtual–physical mismatch under the tested network and motion conditions.

Keywords: synchronous interaction; time-series forecasting; delay compensation; BiLSTM; Kalman filter

Check for updates

Academic Editor: Jie Yang

Received: 21 October 2026
Revised: 11 November 2026
Accepted: 13 November 2026
Published: 17 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Teleoperation enables human operators to perform remote manipulation in environments where direct intervention is difficult, hazardous, or impractical [1]. However, network-induced latency disrupts the temporal correspondence between local operation and remote execution, degrading tracking accuracy, operator perception, and task performance [2], which becomes worse in a public network environment [3]. Moreover, the measurement of time delay is inherently lagging. Therefore, prospective compensation cannot rely on the current measured delay and instead requires an estimate of the forthcoming delay.

Predictive displays and learned action models can mitigate the effects of communication delay by providing information about expected future motion before the delayed remote feedback becomes available [4,5]. More recently, recurrent neural networks have been applied to communication-delay prediction under dynamic network conditions [6]. However, purely data-driven predictions may remain sensitive to abrupt delay variations. Hybrid

learning and recursive estimation provide a complementary approach, as demonstrated by LSTM-assisted delay measurement [7] and learning-aided Kalman-type estimation in robotic systems [8].

Based on these observations, this study proposes one-step-ahead BiLSTM delay prediction with recursive KF correction and uses the corrected estimate for virtual-side feedforward speed compensation in a digital-twin teleoperation platform. This coupling is intended to reduce the temporal and positional mismatch between the virtual and physical arms.

Accordingly, the aim of this study is to develop a prediction-informed synchronous-interaction framework for robotic-arm teleoperation under time-varying communication delay. Specifically, the study aims to:

(1) Design a hybrid BiLSTM-KF predictor that combines BiLSTM-based temporal feature extraction with recursive Kalman-filter correction for one-step-ahead delay estimation;

(2) Establish a feedforward compensation mechanism that uses the corrected delay estimate to adjust the virtual arm manipulation, thereby reducing the position mismatch between the virtual and physical arms without modifying the physical-arm controller;

(3) Evaluate the delay-prediction performance and quantify the resulting improvement in virtual–physical endpoint coherence using normalized endpoint position deviation rates.

The remainder of this paper is organized as follows. Section 2 reviews related delay-compensation and prediction methods. Section 3 presents the proposed architecture, BiLSTM-KF predictor, and virtual-side compensation strategy. Section 4 presents the experimental results and discusses the prediction and compensation performance. Section 5 summarizes the limitations of the current study and outlines future work. Section 6 concludes the paper.

2. Related Work

2.1. Communication-Delay Compensation in Teleoperation

Communication delay in teleoperation has been addressed through controller design, force estimation, and state reconstruction. Guajardo-Benavides and Arteaga [9] developed a force-estimation and control framework for bilateral manipulators under time-varying delays, while Hutchison et al. [10] examined haptic interaction under variable latency. Black and Salcudean [11] further investigated stability and transparency in mixed-reality bilateral teleoperation using locally available interaction information. These approaches primarily mitigate the effects of an existing delay rather than predict the delay of a subsequent communication cycle. In contrast, the present study predicts the subsequent communication delay and uses the estimate to adjust the motion timing of the local virtual arm within a digital-twin teleoperation framework, without modifying the low-level controller of the physical robot.

2.2. Predictive Display and Operation Compensation

To reduce the effect of communication delay on operator feedback, a local virtual model can provide an immediate representation of robot motion while the response of the remote physical robot is still delayed. Aoki et al. [12] investigated transitions between real and virtual environments in delayed teleoperation. Penco et al. [13] anticipated future operator commands for humanoid-robot teleoperation, whereas Prakash et al. [14] generated predictive visual information for vehicle teleoperation under network latency. Different from these works, we slow down the local visual feedback with compensation to match the physical robot motion, since we realize the delay is inevitable and we cannot push the robot move faster or move ahead of time.

2.3. Learning-Based Delay Prediction and Hybrid Estimation

Learning-based methods have increasingly been applied to delayed teleoperation. Zhou et al. [15] proposed LSTM-based active delay estimation, and Slawinski et al. [16] incorporated LSTM into bilateral teleoperation under time-varying delay. Kim et al. [17] used stochastic state estimation to detect and compensate delay outliers. More recent studies include physics-informed LSTM compensation for bilateral mobile-robot teleoperation [18] and online predictive control for time-delayed robotic teleoperation [19].

These studies demonstrate the ability of learning-based models to capture nonlinear temporal behavior and of stochastic estimators to suppress short-term irregularities. Hybrid recurrent-learning and Kalman-type estimation has also been explored for delay measurement and robotic state estimation [7,8]. Existing studies have generally evaluated delay prediction, state estimation, or compensation control as separate tasks. In the present framework, the predicted delay is first corrected recursively and then used directly to modify the virtual-arm speed. The contribution therefore lies in linking delay estimation to synchronization compensation at the system level.

3. Methods

3.1. Digital-Twin-Enabled Teleoperation Architecture and Delay Measurement

The proposed system consists of a local virtual robot arm and a remote physical robot arm connected through a public network, as shown in Figure 1. The local virtual arm can respond to operator input without waiting for the remote physical arm, whereas the physical arm executes the same command after network transmission. Without compensation, this communication delay causes the virtual motion to lead the physical response. The proposed method reduces this temporal lead by adjusting the virtual-arm speed.

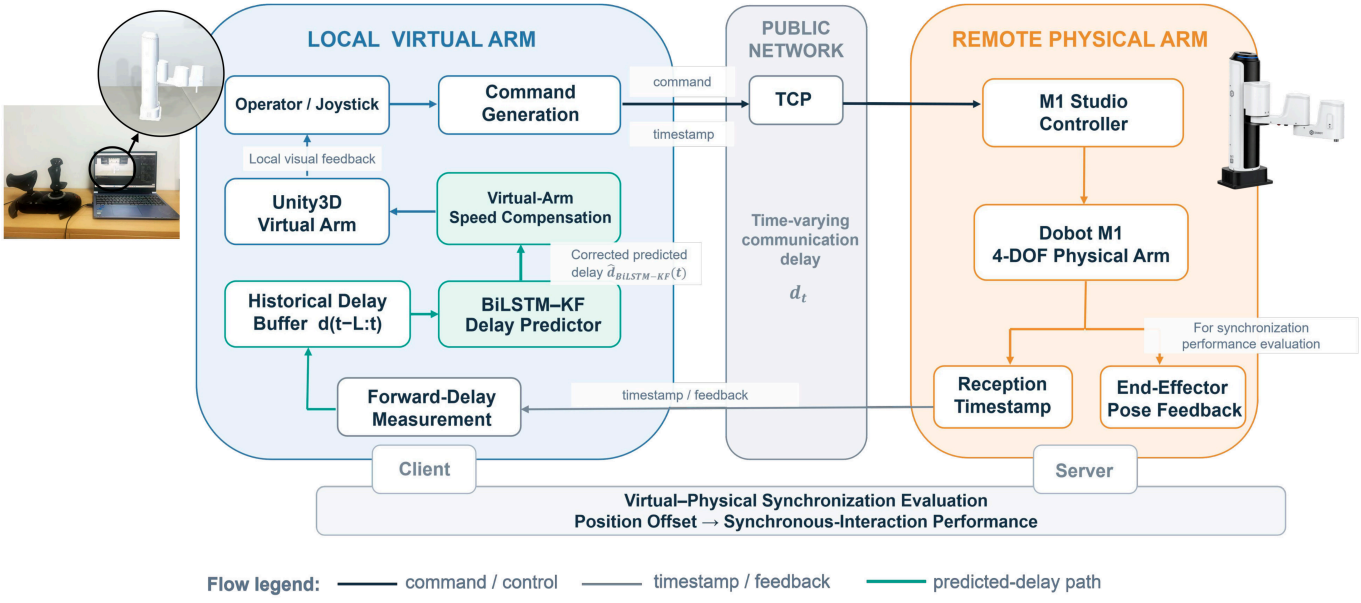


Figure 1. Overall architecture and data flow of the digital-twin teleoperation framework.

The digital-twin framework sends each operator command to both the local virtual arm and the remote physical arm. The remote controller returns the reception timestamp together with the physical-arm end-effector pose, providing the information required for delay measurement and endpoint evaluation. Each measured forward-delay sample is appended to the historical delay sequence, and the most recent L samples are used to generate a one-step-ahead BiLSTM prediction. The prediction is subsequently corrected

by the KF and supplied to the virtual-side speed-compensation module. The command transmitted to the physical arm remains unchanged.

For the t-th command packet, let the local transmission timestamp be $t_{s,t}$ and the corresponding remote reception timestamp be $t_{r,t}$. The forward communication delay is calculated as

$$d_t = t_{r,t} - t_{s,t} \quad (1)$$

Before delay measurement, the clocks of the local and remote computers are synchronized using the Network Time Protocol (NTP) through the Windows Time Service. The transmission timestamp is generated by the local computer, whereas the reception timestamp is generated by the remote computer upon receiving the command packet. The robot controller is not involved in timestamp generation. The forward communication delay for each command packet is calculated as the difference between the remote reception timestamp and the corresponding local transmission timestamp, as defined in Equation (1).

The experimental digital-twin platform consists of a remote physical robot and its local virtual counterpart. The remote physical robot is a 4-axis SCARA manipulator (Dobot M1), with three rotary joints (J1, J2, and J4) and one vertical prismatic joint (J3). The corresponding local virtual arm is implemented in Unity3D 2021.3 LTS. End-effector pose feedback is obtained directly from the Dobot M1 controller without external sensors. The returned robot-state information is transmitted to the local computer for subsequent virtual–physical endpoint evaluation.

3.2. BiLSTM-KF Hybrid Delay Prediction Model

The delay-prediction module comprises two stages. A BiLSTM first generates a one-step-ahead delay prediction from the historical delay sequence, after which a KF recursively corrects the preliminary estimate.

3.2.1. BiLSTM-Based Delay Prediction

The measured communication delay is dynamic, nonlinear, and time-varying. Unlike unidirectional LSTM, which processes the input sequence only chronologically, BiLSTM encodes the same historical window in both forward and backward directions and fuses the resulting representations. It is therefore used here to generate the preliminary one-step-ahead delay prediction.

The delay samples were collected from the teleoperation platform described in Section 3.1 during public-network communication. The complete delay dataset contained 3736 sequential samples. The first 2988 samples (80%) are assigned to the training set and the remaining 748 samples (20%) the test set. The training set was further divided chronologically into 2390 training samples (80%) and 598 validation samples (20%).

The training subset is used for model fitting, whereas the validation subset is used for hyperparameter selection, including the BiLSTM hidden dimension and the KF covariance ratio. The test set is used for the final performance evaluation reported in Section 4.2.

Candidate lookback-window length $L = 10$ was selected as it is consistent with established practices in LSTM-based forecasting [20]. For the hidden dimension H, values of 16, 32, and 64 were evaluated. $H = 32$ yielded the lowest validation error while avoiding unnecessary model complexity, consistent with commonly used LSTM configurations [21]. The final BiLSTM comprised two stacked recurrent layers with 32 hidden units in each direction, followed by fully connected and output layers.

Let the measured delay sequence be $\{d_1, d_2, \ldots, d_{t-1}\}$. To predict the delay at time t, an input vector of length $L = 10$ is constructed:

$$\boldsymbol{X_t} = [d_{t-L}, d_{t-L+1}, \ldots, d_{t-1}]^{\mathrm{T}} \in \mathbb{R}^L \quad (2)$$

The first object of the proposed study is to predict the measured delay d_t of the subsequent communication cycle. For each prediction, the input consists exclusively of the 10 most recent delay samples available at the current prediction time. Thus, we propose a one-step-ahead prediction, which is evaluated against the corresponding measured value. Although the BiLSTM processes the input window in both forward and backward directions, the backward branch only reverses the order of the historical samples within this window; it does not access future delay measurements beyond time t. During sequential evaluation, each newly measured delay is incorporated into the input window after it becomes available for the subsequent prediction.

The forward and backward hidden states $\vec{\boldsymbol{h}}_t$ and $\overleftarrow{\boldsymbol{h}}_t$ encode the same input window in chronological and reverse order, respectively:

$$\vec{\boldsymbol{h}}_t = LSTM_{fwd}(\boldsymbol{X}_t,\ \vec{\boldsymbol{h}}_{t-1};\ \boldsymbol{W}_{fwd}) \tag{3}$$

$$\overleftarrow{\boldsymbol{h}}_t = LSTM_{bwd}(\boldsymbol{X}_i,\ \overleftarrow{\boldsymbol{h}}_{t+1};\ \boldsymbol{W}_{bwd}) \tag{4}$$

where $\boldsymbol{W}_{fwd}$ and $\boldsymbol{W}_{bwd}$ denote the forward and backward LSTM parameters. The two hidden states are concatenated as

$$\boldsymbol{h}_t = \left[\vec{\boldsymbol{h}}_t;\ \overleftarrow{\boldsymbol{h}}_t\right] \in \mathbb{R}^{2H} \tag{5}$$

where $H = 32$ is the hidden unit dimension of a single-direction LSTM. The concatenated vector $\boldsymbol{h}_t$ is then passed through a fully connected layer and an output layer to obtain the delay prediction:

$$\boldsymbol{z}_t = \mathrm{ReLU}(\boldsymbol{W}_c\boldsymbol{h}_t\ +\ \boldsymbol{b}_c) \tag{6}$$

where $\boldsymbol{W}_c \in \mathrm{R}^{\mathrm{H}\times 2\mathrm{H}}$ and $\boldsymbol{b}_c \in \mathrm{R}^{\mathrm{H}}$ are the parameters of the fully connected layer

$$\hat{d}_{BiLSTM}(t) = \boldsymbol{W}_o\boldsymbol{z}_t\ +\ b_o \tag{7}$$

where $\boldsymbol{W}_o$ and $\boldsymbol{b}_o$ are the weight matrix and bias of the output layer, respectively. The predicted delay $\hat{d}_{BiLSTM}(t)$ is then used as the observation input to the KF correction module described in Section 3.2.2.

The BiLSTM was trained using mean squared error (MSE) loss and the Adam optimizer with an initial learning rate of 0.001. The batch size was 32, with a maximum of 50 epochs and early stopping after 20 epochs without validation-loss improvement. Xavier uniform initialization was used for the model parameters.

3.2.2. Kalman-Filter-Based Dynamic Correction

At each prediction cycle, the latest measured delay d_t is appended to the historical delay sequence and included in the BiLSTM input window to generate a one-step-ahead prediction. To refine the BiLSTM output, the KF then operates on this BiLSTM prediction $\hat{d}_{BiLSTM}(t)$, which is treated as a pseudo-observation of the next-step delay state, rather than directly using the latest raw delay measurement d_t as its observation. Since no deterministic transition law is assumed for public-network delay, its short-term evolution is represented by a scalar random-walk model. Accordingly, the state-transition and pseudo-observation equations are formulated as follows:

$$d_t = d_{t-1} + w_{t-1}, w_{t-1} \sim \mathcal{N}(0, Q) \tag{8}$$

$$\hat{d}_{BiLSTM}(t) = d_t + v_t, v_t \sim \mathcal{N}(0, R) \tag{9}$$

where w_{t-1} and v_t denote the process and observation noise, with covariance terms Q and R, respectively. Their relative weighting is characterized by the covariance ratio R/Q, whose sensitivity is evaluated in Section 4.1. The selected ratio is subsequently used in the prediction and compensation experiments.

Here, newly measured delays affect subsequent predictions through the BiLSTM input window, whereas the KF operates directly on the BiLSTM-generated pseudo-observations.

Equations (8) and (9) are rewritten in the standard state-space form as:

$$\begin{aligned} \boldsymbol{x}_t &= \boldsymbol{F}\boldsymbol{x}_{t-1} + \boldsymbol{w}_{t-1} \\ \boldsymbol{z}_t &= \boldsymbol{H}\boldsymbol{x}_t + \boldsymbol{v}_t \end{aligned} \tag{10}$$

where $\boldsymbol{x}_t$ is the delay state, $\boldsymbol{z_t}$ is the BiLSTM-derived observation, and $\boldsymbol{Q}$ and $\boldsymbol{R}$ denote the process- and observation-noise variances, respectively.

The KF recursively updates the delay estimate through prediction and correction steps. Let $\hat{\boldsymbol{x}}_{t-1}$ and $\boldsymbol{P}_{t-1}$ denote the updated state estimate and its associated error covariance at the previous time step. The prediction step yields the prior state estimate $\hat{\boldsymbol{x}}_t'$ and covariance $\boldsymbol{P}_t'$:

$$\hat{\boldsymbol{x}}_t' = \boldsymbol{F}\hat{\boldsymbol{x}}_{t-1} \tag{11}$$

$$\boldsymbol{P}_t' = \boldsymbol{F}\boldsymbol{P}_{t-1}\boldsymbol{F}^T + \boldsymbol{Q} \tag{12}$$

The correction step then incorporates the current BiLSTM-derived observation z_t, computes the Kalman gain $\boldsymbol{K}_t$, and updates the prior estimate $\hat{\boldsymbol{x}}_t$ to obtain the updated state estimate and covariance $\boldsymbol{P}_t$:

$$\boldsymbol{K}_t = \boldsymbol{P}_t'\boldsymbol{H}^T\left(\boldsymbol{H}\boldsymbol{P}_t'\boldsymbol{H}^T + \boldsymbol{R}\right)^{-1} \tag{13}$$

$$\hat{\boldsymbol{x}}_t = \hat{\boldsymbol{x}}_t' + \boldsymbol{K}_t\left(z_t - \boldsymbol{H}\hat{\boldsymbol{x}}_t'\right) \tag{14}$$

$$\boldsymbol{P}_t = (\boldsymbol{I} - \boldsymbol{K}_t\boldsymbol{H})\boldsymbol{P}_t' \tag{15}$$

Because the state vector contains only the communication-delay state, the corrected delay estimate is obtained from its single component as:

$$\hat{d}_{BiLSTM-KF}(t) = \hat{\boldsymbol{x}}_t[0] \tag{16}$$

For the selected covariance ratio (R/Q = 10), the steady-state Kalman gain of the scalar random-walk model is approximately 0.27. This means that the KF update assigns approximately 27% weight to the new BiLSTM-derived pseudo-observation and 73% to the prior filtered estimate. This behavior not only smooths short-term prediction fluctuations but also attenuate abrupt delay peaks. And it is considered when interpreting the prediction performance under abrupt network conditions.

As summarized in Figure 2, the historical delay window is first processed by the BiLSTM. The resulting BiLSTM-derived observation $\hat{d}_{BiLSTM}(t)$ is incorporated into the KF update to obtain the corrected delay estimate $\hat{d}_{BiLSTM-KF}(t)$ used by the compensation strategy in Section 3.3.

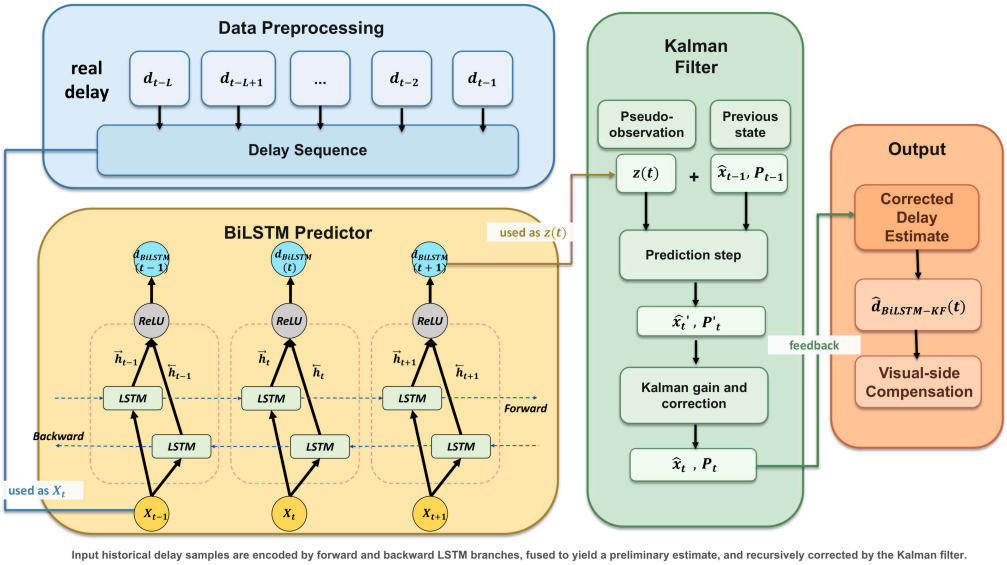


Figure 2. Workflow of the BiLSTM-KF hybrid model.

3.3. Prediction-Informed Virtual-Side Speed Compensation

Network delay causes the remote physical arm to execute the operator command later than the local virtual arm. To reduce this temporal mismatch, the corrected delay is used only to modify the virtual-arm speed; the operator command transmitted to the physical arm remains unchanged. The compensation is implemented at the command level.

3.3.1. Compensation Law

For the compensation calculation, let v_r denote the reference speed generated from the operator input during the current control interval. No constant-speed assumption is imposed on the complete trajectory; v_r is treated as locally constant only within one update interval. The corrected delay estimate is then used to reduce the virtual-side reference speed for that interval. This equivalent constant-speed assumption is used only for the compensation law; its limitations with respect to acceleration and deceleration are discussed in Section 5.

For a single operation command, let $\boldsymbol{p}_{target}$ denote the target position and $\boldsymbol{p}_{virt}(t)$ the current virtual-arm position at control cycle t. The vector $\boldsymbol{e}(t) = \boldsymbol{p}_{target} - \boldsymbol{p}_{virt}(t)$ represents the remaining displacement of the virtual arm to the target. Its magnitude is denoted by $D(t) = \|\boldsymbol{e}(t)\|$. In the current control cycle, the nominal remaining motion duration is estimated as $T_r(t) = D(t)/v_r$, where v_r is the reference speed generated from the operator input. The corrected delay estimate $\hat{d}_{BiLSTM-KF}(t)$ is then added to this nominal remaining duration to obtain the compensated virtual-side duration $T_v(t) = T_r(t) + \hat{d}_{BiLSTM-KF}(t)$.

The virtual-arm speed is adjusted so that the current remaining displacement is traversed over the compensated duration, as given in Equation (17). This adjustment reduces the temporal lead of the local virtual motion and is therefore intended to reduce the resulting virtual–physical mismatch.

Thus, the compensated velocity of the virtual arm is

$$\boldsymbol{v}_{virt}(t) = \frac{D(t)}{T_r(t)+\hat{d}_{BiLSTM-KF}(t)} \cdot \frac{\boldsymbol{e}(t)}{\|\boldsymbol{e}(t)\|} = \frac{v_r}{1+\frac{v_r \hat{d}_{BiLSTM-KF}(t)}{D(t)}} \cdot \frac{\boldsymbol{e}(t)}{\|\boldsymbol{e}(t)\|} \tag{17}$$

The duration relation is applied locally at each control cycle rather than fixed for the entire trajectory. At control cycle t, the current virtual-arm position and BiLSTM-KF delay estimate are used to update the remaining distance and compensated speed. After one control interval, these quantities are recalculated for the next cycle.

3.3.2. Compensation Workflow

At each control cycle, the virtual arm receives the target position $\boldsymbol{p}_{target}$, calculates the current remaining displacement $\boldsymbol{e}(t)$, and reads the current BiLSTM-KF delay estimate $\hat{d}_{BiLSTM-KF}(t)$. Equation (17) is then used to calculate the compensated virtual-arm velocity for the current control interval. At the next cycle, the remaining displacement and delay estimate are updated and the calculation is repeated. The complete workflow is shown in Figure 3.

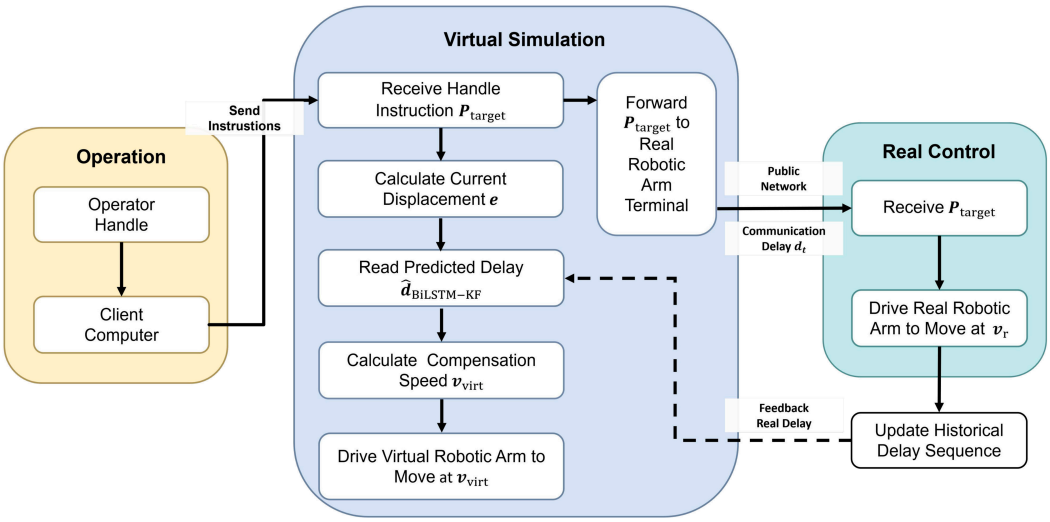


Figure 3. Compensation workflow of the arm based on the BiLSTM-KF hybrid model.

4. Experiments and Results

Experiments were conducted on the digital-twin teleoperation platform described in Section 3 to evaluate delay prediction and virtual–physical endpoint coherence. Prediction performance was assessed on the held-out test sequence, while the virtual and physical end-effector positions were recorded for endpoint-deviation evaluation. The evaluation covered the sensitivity of the KF covariance ratio, prediction performance relative to the KF and LSTM baselines, and the effectiveness of prediction-informed virtual-side compensation.

4.1. Sensitivity Analysis of the KF Noise-Covariance Ratio

The covariance ratio R/Q controls the relative weighting of the recursive state estimate and the BiLSTM-derived observation. Its influence on prediction performance is therefore evaluated before the subsequent experiments.

With the BiLSTM configuration fixed, five R/Q values (100, 10, 1, 0.1, and 0.01) were evaluated on the validation sequence using cumulative RMSE, MAE, and MAPE. As shown in Figure 4, $R/Q = 10$ yielded the lowest overall validation errors while maintaining relatively stable performance and was therefore selected for the subsequent test evaluation.

Temporary increases in the cumulative errors occur around samples 25–30 and 90–95, coinciding with intervals of larger instantaneous prediction deviations. Similar increases are observed for several R/Q settings, indicating that these changes are mainly associated with abrupt variations in the measured delay sequence rather than with a single parameter setting.

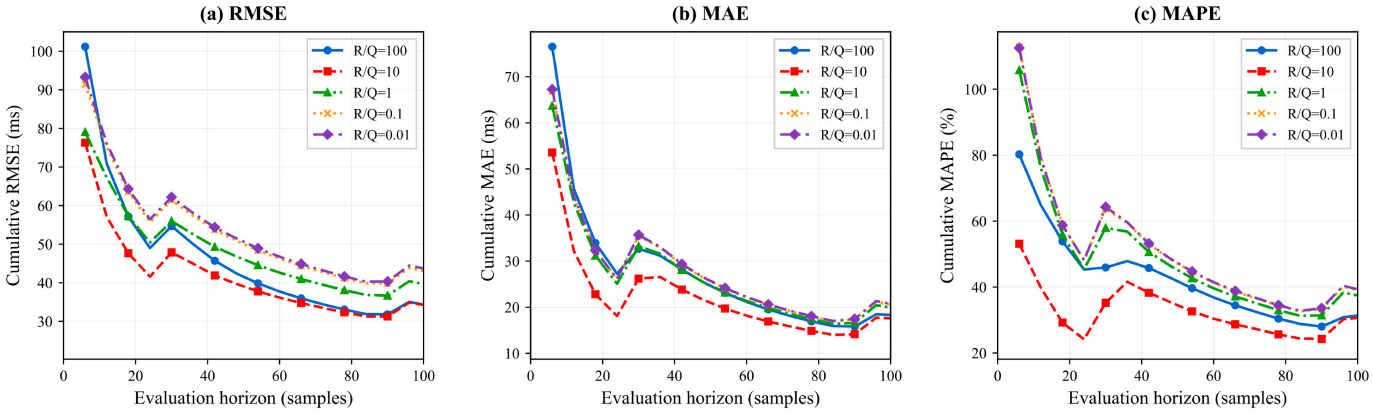


Figure 4. Cumulative validation errors for different R/Q ratios: (**a**) RMSE, (**b**) MAE, and (**c**) MAPE.

4.2. Comparative Analysis of Delay Prediction Performance of Different Methods

The standalone KF, LSTM, and BiLSTM-KF models were evaluated on the same aligned test sequence. For comparability, the unidirectional LSTM used the same recurrent-layer number (2), hidden dimension (H = 32), optimizer, learning rate (0.001), batch size (32), and early-stopping patience (20 epochs) as the BiLSTM. The standalone KF used $R/Q = 10$, and an initial error covariance of P = 1.

A representative segment of the test sequence is given in Figure 5. Figure 5a compares the measured delay with the corresponding one-step-ahead estimates from the three prediction methods. The KF follows the general delay trend but attenuates several abrupt peaks, whereas the LSTM captures more short-term variations but retains noticeable deviations during rapid changes. The BiLSTM-KF prediction shows closer overall coherence with the measured sequence over the displayed test interval. However, this visual coherence should not be interpreted as accurate prediction of every abrupt peak. The residual plots in Figure 5b–d provide a more direct assessment of prediction errors, particularly during rapid delay fluctuations. The KF and LSTM exhibit larger errors during abrupt delay changes, whereas the BiLSTM-KF residuals remain more concentrated around zero over most of the sequence.

In Figure 5, the standalone KF estimate is obtained by filtering the measured-delay sequence, whereas the BiLSTM-KF prediction is obtained by filtering the BiLSTM-generated one-step-ahead predictions treated as pseudo-observations. Although both use the same covariance ratio, their different observation inputs may result in different responses to abrupt delay peaks.

Table 1 gives the prediction errors calculated over all 748 samples in the test set. The BiLSTM-KF model achieved the lowest errors, with an RMSE of 5.6498 ms, an MAE of 3.5455 ms, and an MAPE of 7.84%, better than both reference methods. These results confirm the improved one-step-ahead prediction accuracy of the hybrid model on the evaluated test sequence.

Table 1. Quantitative comparison of delay-prediction performance of different methods.

Method	RMSE↓(ms)	MAE↓(ms)	MAPE↓(%)
KF	13.1915	7.1007	14.94
LSTM	8.7451	4.8720	10.65
BiLSTM-KF	5.6498	3.5455	7.84

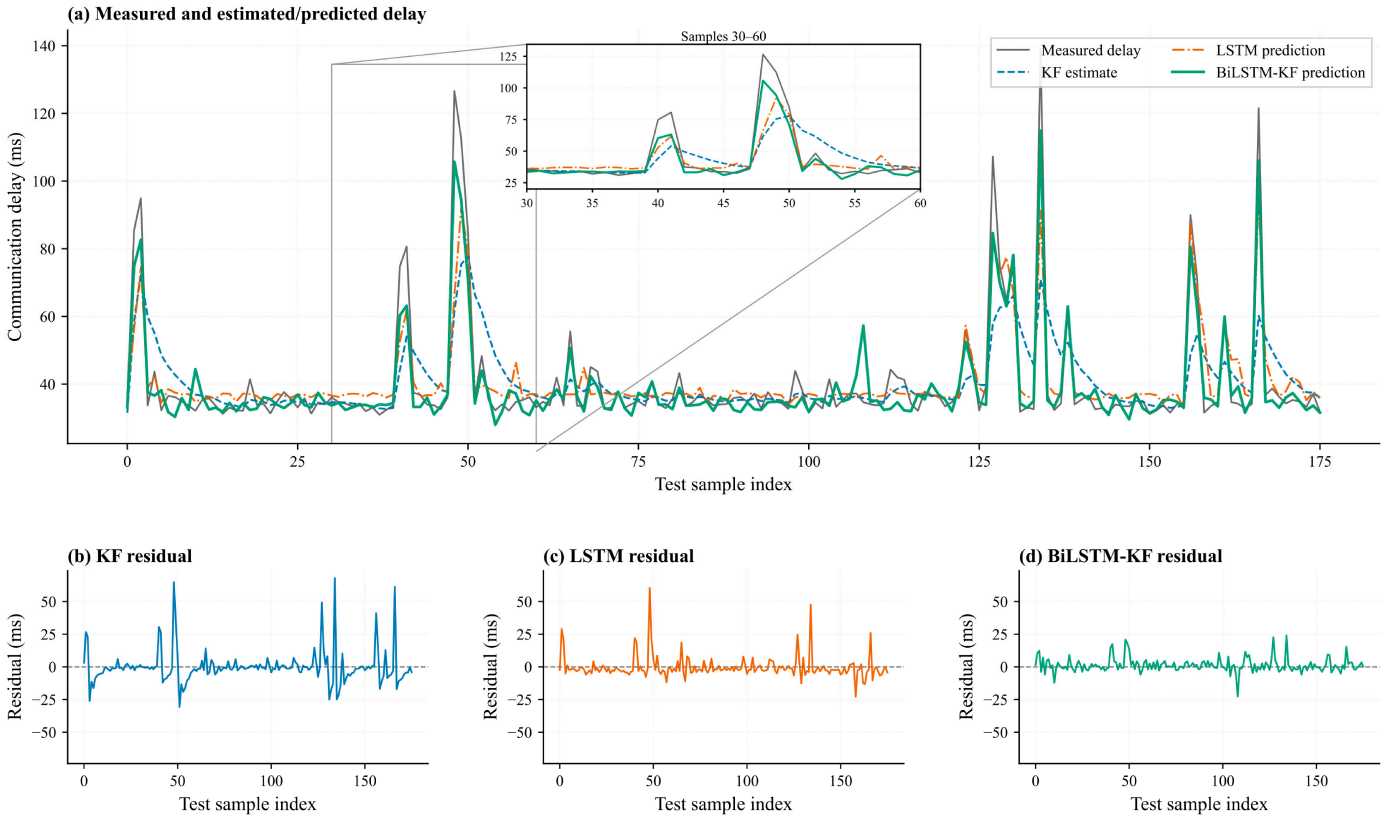


Figure 5. Measured and estimated/predicted communication delays and corresponding residuals: (**a**) overall comparison with a zoomed view of samples 30–60; (**b**) KF residual; (**c**) LSTM residual; and (**d**) BiLSTM-KF residual.

4.3. Effectiveness Validation of the Compensation Strategy

Endpoint discrepancy between the virtual and physical arms was evaluated under four conditions: no compensation, historical-mean delay compensation, LSTM-based compensation, and BiLSTM-KF-based compensation. Each execution comprised 0.1 m translations along the X, Y, and Z axes, a diagonal XY motion of 0.1 m along each axis, and a quarter-circular trajectory with a radius of 0.1 m. All motion segments were performed at a nominal speed of 0.05 m/s and lasted approximately 2–4 s, depending on the trajectory. For each compensation condition, the complete motion sequence was executed consecutively six times.

For the historical-mean delay condition, a fixed delay value $\bar{d}_{his}$ calculated from the training delay samples is used throughout the compensation experiment,

$$\bar{d}_{his} = \frac{1}{N_{train}} \sum_{i=1}^{N_{train}} d_i \tag{18}$$

where N_{train} is the number of training samples.

This value replaced the time-varying delay estimate in Equation (17), while the remaining compensation procedure and motion protocol were unchanged. The historical-mean condition therefore served as a simple baseline for evaluating the additional benefit of predicting time-varying delay.

After each execution, one pair of virtual- and physical-arm end-effector coordinates was recorded. Endpoint discrepancy was quantified using the normalized endpoint position deviation rate R_{offset} defined in Equation (19), where $\boldsymbol{p}_v$ and $\boldsymbol{p}_r$ denote the virtual and physical-arm endpoint positions recorded after the same execution, respectively. The same endpoint sampling procedure was used for all compensation conditions. A smaller value indicates closer endpoint coherence.

The performance observed under the corresponding experimental network conditions was compared with the same motion protocol and endpoint sampling procedure. Since the conditions were tested in separate public-network, identical delay realizations were not imposed across them.

$$R_{offset} = \frac{\|p_v - p_r\|}{\|p_v\|} \tag{19}$$

As shown in Figure 6, the deviation ranged from 3.953% to 6.480% without compensation, from 3.480% to 5.400% with historical-mean delay compensation, from 2.890% to 4.747% with LSTM-based compensation, and from 1.592% to 3.031% with BiLSTM-KF-based compensation. The BiLSTM-KF condition produced the lowest endpoint deviation in all six executions.

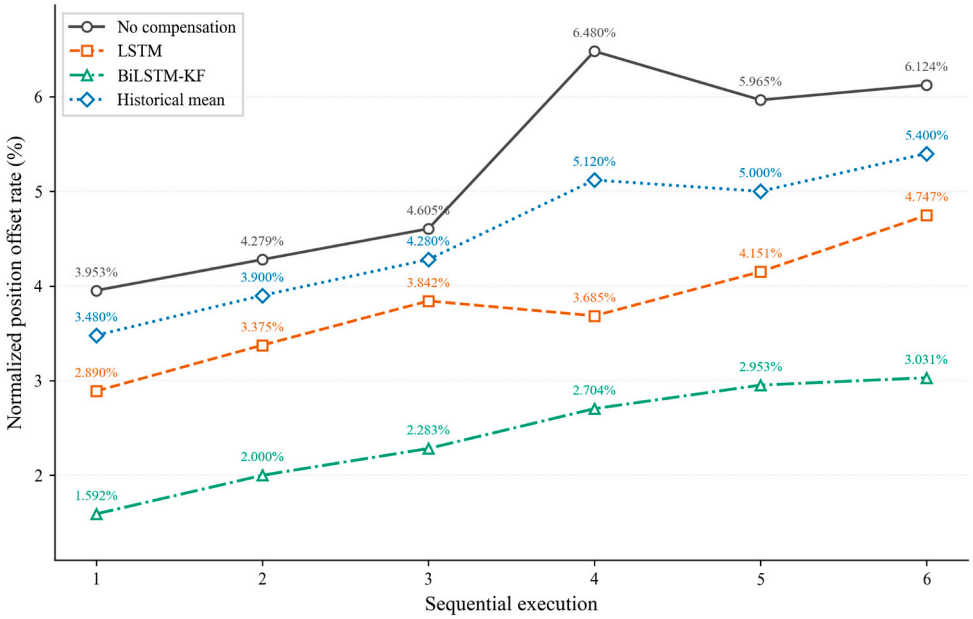


Figure 6. Normalized endpoint position deviation rates across six sequential executions under different compensation conditions.

As summarized in Table 2, the mean endpoint deviation rate decreased from 5.234 ± 1.080% without compensation to 4.530 ± 0.760% with historical-mean delay compensation, 3.782 ± 0.639% with LSTM compensation and 2.427 ± 0.569% with BiLSTM-KF compensation. The BiLSTM-KF strategy reduced the mean deviation by 53.63% relative to no compensation, 46.42% relative to historical-mean delay compensation, and 35.83% relative to LSTM-based compensation.

Table 2. Normalized endpoint deviation rates across six sequential executions under different compensation conditions.

Strategy	Mean ± SD (%)	Improvement vs. No Compensation (%)	Improvement vs. Historical Mean (%)	Improvement vs. LSTM (%)
No compensation	5.234 ± 1.080	-	-	-
Historical mean	4.530 ± 0.760	13.46		
LSTM	3.782 ± 0.639	27.74	16.52	-
BiLSTM-KF	2.427 ± 0.569	53.63	46.42	35.83

Overall, the BiLSTM-KF compensation condition produced the lowest virtual–physical position deviation rates under the test motion and network conditions.

4.4. Discussion

The BiLSTM-KF model achieved lower prediction errors than the standalone KF and LSTM on the test sequence and captured both the overall delay trend and much of its short-term variation. However, the model did not accurately anticipate every abrupt delay peak. Because each prediction is based only on the preceding 10 delay samples, a sudden spike without an informative preceding pattern cannot be predicted reliably. Consistent with this limitation, the largest absolute prediction error was approximately 24.2 ms and occurred during an abrupt delay fluctuation.

The observed performance is consistent with the complementary roles of the two model components: the BiLSTM captures nonlinear temporal dependencies in the historical delay sequence, whereas the KF recursively smooths the resulting one-step-ahead predictions. The KF can be understood as a recursive smoothing stage rather than a direct fusion of raw measured delays and BiLSTM predictions. However, the individual contributions of bidirectional feature extraction and Kalman smoothing are not quantified separately in our paper. The improvement should therefore be attributed to the complete BiLSTM-KF framework rather than to either component alone.

As mentioned above, BiLSTM-KF-based compensation produced the lowest normalized position deviation rates among the four tested conditions. These results indicate improved virtual–physical endpoint coherence under the tested protocol. Since continuous position trajectories were not used in this evaluation, the present experiment does not establish time-resolved dynamic synchronization throughout the motion.

5. Limitations and Future Work

The proposed method has several limitations. First, the virtual-side compensation is derived from an equivalent constant-velocity approximation within each control interval and therefore does not explicitly account for acceleration, deceleration, or rapid changes in motion direction. Such simplification may result in the actual motion duration deviating from the estimated value during highly dynamic movements. Future work will therefore incorporate time-varying motion models to improve compensation accuracy during transient motion phases.

Second, the current compensation directly reduces the virtual-arm velocity as the predicted delay increases. Under unusually large communication delays, this mechanism may lead to a slow virtual response and consequently degrade operator-side interactivity. Future work will investigate adaptive compensation gains, velocity saturation, and prediction-confidence-based switching mechanisms to maintain a balance between virtual–physical synchronization and responsive local interaction.

Third, NTP synchronization introduces uncertainty into the absolute one-way delay measurement. The time error was not continuously monitored during the original experiments. Future work will employ more precise clock synchronization and continuous deviation monitoring.

Fourth, the BiLSTM-KF model predicts the next delay from a finite historical window of 10 samples. Abrupt delay spikes without recognizable preceding patterns therefore remain difficult to anticipate. Additional network-state information and adaptive prediction strategies will be investigated to improve robustness under rapidly changing network conditions.

Finally, the current digital-twin framework uses the end-effector pose returned by the physical robot to update and evaluate the local virtual representation. Future work will enrich the state exchange between the physical and virtual systems and investigate tighter physical–virtual synchronization under more complex manipulation conditions. Future experiments will also be considered to evaluate whether the reduction in objective

position mismatch is accompanied by improvements in perceived synchronization and operator workload.

6. Conclusions

This study developed and evaluated a prediction-informed digital-twin teleoperation framework for reducing the virtual–physical motion mismatch caused by time-varying communication delay. Historical delay measurements are used for one-step-ahead BiLSTM prediction, the preliminary estimate is recursively corrected by a KF, and the corrected delay is incorporated into virtual-side speed compensation without modifying the low-level controller of the physical robot.

The main contribution is the direct coupling of corrected delay estimation with virtual-side motion adaptation in a digital-twin teleoperation framework. By using the corrected delay estimate to adjust the local virtual-arm motion while leaving the command transmitted to the physical robot unchanged, the proposed method links delay prediction directly to virtual–physical synchronization compensation.

Our experiments demonstrate the effectiveness of the BiLSTM-KF model in delay prediction and show that the proposed compensation strategy reduces the virtual–physical endpoint position deviation. These findings indicate that prediction-informed virtual-side adaptation can improve endpoint coherence between the virtual and physical arms under the tested network and motion conditions. The study therefore provides a practical framework for linking communication-delay prediction with virtual-side compensation in digital-twin-enabled robotic teleoperation.

Author Contributions: Conceptualization, Y.L.; methodology, M.S. and Y.L.; validation, M.S.; investigation, Z.L. and P.H.; data curation, M.S.; writing—original draft preparation, M.S. and Y.L.; writing—review and editing, Y.L., Z.L. and P.H.; visualization, M.S.; supervision, Y.L.; project administration, Y.L.; funding acquisition, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Research and Development Program of Shaanxi (Key Project), grant number 2026CY-ZDGG-95.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors upon reasonable request.

Acknowledgments: During the preparation of this manuscript, the authors used DeepSeek (web-based version; no specific model version was displayed) for translating certain sections of the manuscript from Chinese to English and refining the language for clarity and readability. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Darvish, K.; Penco, L.; Ramos, J.; Cisneros, R.; Pratt, J.; Yoshida, E.; Ivaldi, S.; Pucci, D. Teleoperation of Humanoid Robots: A Survey. *IEEE Trans. Robot.* **2023**, *39*, 1706–1727. [CrossRef]
2. Black, D.G.; Andjelic, D.; Salcudean, S.E. Evaluation of Communication and Human Response Latency for (Human) Teleoperation. *IEEE Trans. Med. Robot. Bionics* **2024**, *6*, 53–63. [CrossRef]
3. Kamtam, S.B.; Lu, Q.; Bouali, F.; Haas, O.C.L.; Birrell, S. Network Latency in Teleoperation of Connected and Autonomous Vehicles: A Review of Trends, Challenges, and Mitigation Strategies. *Sensors* **2024**, *24*, 3957. [CrossRef] [PubMed]
4. Zhu, Y.; Fusano, K.; Aoyama, T.; Hasegawa, Y. Intention-reflected predictive display for operability improvement of time-delayed teleoperation system. *ROBOMECH J.* **2023**, *10*, 17. [CrossRef]
5. Lu, Z.; Si, W.; Wang, N.; Yang, C. Dynamic Movement Primitives-Based Human Action Prediction and Shared Control for Bilateral Robot Teleoperation. *IEEE Trans. Ind. Electron.* **2024**, *71*, 16654–16663. [CrossRef]

6. Kang, H.; Kim, E.; Hwang, M.; Byeon, J.; An, J.; Cha, H. IPT-DCD: Interpolation Predictor for Teleoperation Under Dynamic Communication Delay Using Deep Learning Approach. *Sensors* **2025**, *25*, 4118. [CrossRef] [PubMed]
7. Dong, J.; Wang, Z.; Ding, S.; Wang, B.; Yu, S.; Luo, B.; Guo, H. Delay Measurement Enhancement by the LSTM Embedded Kalman Filtering in Fiber-Optic Time Distribution System. *IEEE Trans. Instrum. Meas.* **2025**, *74*, 5505508. [CrossRef]
8. Yeaser, A.; Tung, J.; Huissoon, J.; Hashemi, E. Learning-aided state estimation for robotic rollators with experimental validation. *Robot. Auton. Syst.* **2025**, *194*, 105140. [CrossRef]
9. Guajardo-Benavides, E.J.; Arteaga, M.A. On the finite time force estimation for bilateral teleoperation of robot manipulators with time varying delays. *Control Eng. Pract.* **2023**, *138*, 105622. [CrossRef]
10. Hutchison, C.; Hewlett, J.; Arbatani, S.; Weill-Duflos, A.; Kovecses, J. Haptic Interactions Subject to Variable Latency. *IEEE Trans. Haptics* **2024**, *17*, 66–71. [CrossRef] [PubMed]
11. Black, D.G.; Salcudean, S.E. Stability and Transparency in Mixed-Reality Bilateral Human Teleoperation. *IEEE Trans. Robot.* **2025**, *41*, 5800–5815. [CrossRef]
12. Aoki, J.; Sasaki, F.; Yamashina, R.; Kurazume, R. Teleoperation by seamless transitions in real and virtual world environments. *Robot. Auton. Syst.* **2023**, *164*, 104405. [CrossRef]
13. Penco, L.; Mouret, J.-B.; Ivaldi, S. Prescient Teleoperation of Humanoid Robots. In Proceedings of the 2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids), Austin, TX, USA, 12–14 December 2023; pp. 1–8.
14. Prakash, J.; Vignati, M.; Vignarca, D.; Sabbioni, E.; Cheli, F. Predictive Display with Perspective Projection of Surroundings in Vehicle Teleoperation to Account Time-Delays. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 9084–9097. [CrossRef]
15. Zhou, X.; Bai, W.; Ren, Y.; Yang, Z.; Wang, Z.; Lo, B.; Yeatman, E.M. An LSTM-based Bilateral Active Estimation Model for Robotic Teleoperation with Varying Time Delay. In Proceedings of the 2022 7th IEEE International Conference on Advanced Robotics and Mechatronics (ICARM), Guilin, China, 9–11 July 2022; pp. 725–730.
16. Slawiński, E.; Rossomando, F.; Chicaiza, F.A.; Moreno-Valenzuela, J.; Mut, V. LSTM network in bilateral teleoperation of a skid-steering robot. *Neurocomputing* **2024**, *602*, 128248. [CrossRef]
17. Kim, E.; Hwang, M.; Lim, T.; Jeong, C.; Yoon, S.; Cha, H. Communication Delay Outlier Detection and Compensation for Teleoperation Using Stochastic State Estimation. *Sensors* **2024**, *24*, 1241. [CrossRef] [PubMed]
18. Abubakar, A.; Zweiri, Y.H.; Haddad, A.G.; Yakubu, M.; Alhammadi, R.; Seneviratne, L.D. Physics-informed LSTM-based delay compensation framework for high-fidelity teleoperation of wheeled mobile robots. *Robot. Auton. Syst.* **2026**, *197*, 105265. [CrossRef]
19. Ebrahimian, M.; Pourmokhtari, M.; Ghiyasi, M.; Yazdankhoo, B.; Beigzadeh, B. Online Bilateral Predictive Control for Time-Delayed Teleoperation of Snake-like Robots. *J. Intell. Robot. Syst.* **2024**, *110*, 80. [CrossRef]
20. Bérot, O.S.; Canot, H.; Durand, P.; Hassoune-Rhabbour, B.; Acheritobehere, H.; Laforet, C.; Nassiet, V. Choice of parameters of an LSTM network, based on a small experimental dataset. *Eng. Lett.* **2024**, *32*, 59–71.
21. Thakur, N.; Karmakar, S.; Soni, S. Time series forecasting for uni-variant data using hybrid GA-OLSTM model and performance evaluations. *Int. J. Inf. Technol.* **2022**, *14*, 1961–1966. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.