Model-mediated Delay Compensation with Goal Prediction for Robot Teleoperation over Internet

Rolif Lima, Vismay Vakharia, Utsav Rai, Hardik Mehta, Vighnesh Vatsal and Kaushik Das

***Abstract*— Teleoperated robots have enabled humans to manipulate objects in remote environments without requiring physical presence. In this paper we focus on teleoperation of a robotic arm with shared control between the robot and the operator. A model-mediated approach is used to compensate for delays in the communication channel. Position information of the operator's arm is captured and processed to compute the states of a motion prediction model before transmission over a network to be used on the robot's side, allowing for compensation of transmission delays. Model Predictive Control (MPC) and a novel goal prediction algorithm is used to follow the operator's intended motion while reducing the cognitive loads arising from collision avoidance and fine manipulation in the remote environment. We evaluate the proposed method against a baseline pure teleoperation condition with an inverse kinematic controller and observe that the proposed approach improves the overall teleoperation performance in terms of task completion time.**

I. INTRODUCTION

Tele-robotics is a popular and continuously evolving paradigm in the robotics community. Even though machine learning applications have enabled robots with autonomous capabilities [1], [2], [3], teleoperation allows humans to apply their reasoning, creativity, and intuition to perform tasks that are difficult for autonomous agents. These systems are best suited for critical tasks where autonomous systems may have unacceptable rates of error, such as nuclear waste handling, robotic surgery, space, and deep-sea exploration [4], [5], [6], [7], [8].

Existing work in this domain focuses on bilateral teleoperation [9], [10], where haptic feedback is provided to the operator, allowing them to improve system stability using force sensing [11], [12]. Such systems need additional hardware and instrumentation, which affects the overall motion capabilities of the operator and adds to the overall cost of the system, including additional training requirements.

Though such systems are crucial to applications such as telerobotic surgery, for applications involving known environments with known objects, such as a warehouse or retail store with standard payloads, haptic feedback is not necessary. In such scenarios, visual feedback from the remote environment along with shared capabilities may be sufficient for performing tasks efficiently [13].

Teleoperation with only visual feedback requires the remote robot to possess autonomous capabilities [14], [15], such as avoiding collision and identifying the operator's task-specific intent, thus minimizing the cognitive loads that may otherwise arise due to the lack of haptic feedback [16], [17]. Model Predictive Control (MPC) provides an elegant way to plan collision-free paths by repeatedly solving a constrained optimal control problem for a finite time horizon to modify the operator's inputs. Additionally, predicting the operator's intent [18], [19] frees them from performing critical motion adjustments in domain-specific tasks such as picking and placing, involving the fine alignment of the end-effector with the target. This goal-intent prediction, coupled with MPC, allows for efficient remote manipulation.

A. Related Work

Tele-manipulation [20] has been used in several applications due to the safety and access it provides to the operator. A nuclear material handling teleoperation system is described in [21], along with associated challenges and solution strategies. A dual-arm teleoperation architecture with haptic and visual feedback specific to surface conditioning tasks is demonstrated in [22]. Due to the differences in manipulation needs, it uses two different motion capture systems with haptic feedback for each arm. The use of augmented reality (AR) and mixed reality setups are demonstrated in [23], [24] for an intuitive user interface through the projection of virtual objects on real environments and a gamepad to provide input for tele-operating the robot. Most of these works rely on using inverse kinematic solutions for computing joint reference angles to control the manipulator. These approaches lack the ability to alter the input from the operator in case of a possible collision.

In this respect, MPC has been used for various aspects of teleoperation, such as manipulation, delay compensation, and collision avoidance. The teleoperation of a humanoid robot with two different modes for locomotion is shown in [25]. MPC is used for computing zero moment point velocities for stable locomotion of the robot while constraining the same to prevent self-collision, satisfying the kinematic constraints and stability of the system. In [26], a prediction model was used on the operator's side to obtain the future reference for the remote robot, transmitting the same over a network, and tracking the commanded reference using MPC. Distributed MPC was used in [27], with controllers on both the operator and remote robot ends, aiming to maximize the system transparency while simultaneously maintaining the controller's and communication channel's passivity. Due to the ability of MPC to efficiently solve an optimal control problem while considering several constraints, it has been extensively used

The authors are with TCS Research & Innovation, Tata Consultancy Services, Bengaluru, India. e-mails: {rolif.lima, rai.utsav, vismay.vakharia, hr.mehta, vighnesh.vatsal, kaushik.da}@tcs.com

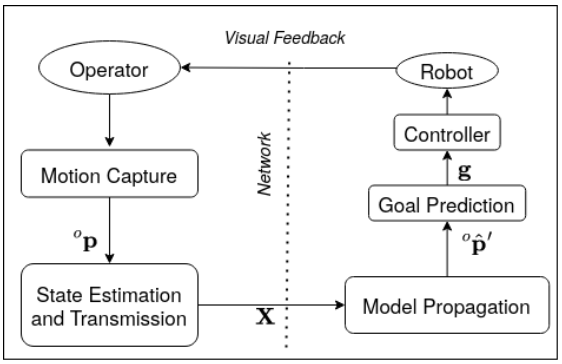


Fig. 1: System architecture

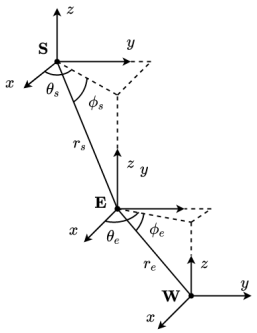


Fig. 2: The kinematic structure used for motion prediction

in trajectory optimization and tracking applications [22], [28].

Semi-autonomous teleoperation of manipulator without explicitly computing inverse kinematics was demonstrated in [29], using MPC with constraints to prevent collisions. We extend this work and propose a motion prediction algorithm to forecast human motion and handle variable time delays over a network. Additionally, we impose constraints to enforce a desirable robot configuration.

B. Contributions

This work contributes toward the development of a teleoperation system architecture with shared control between the operator and the remotely located robot with only visual feedback to the operator. Novelty in the proposed architecture can be summarised as follows:

1) A model-mediated approach for network delay compensation is developed, employing a simplified kinematic model to predict operator arm motion.
2) A goal prediction algorithm is proposed for predicting the operator's intent while approaching an object in a remote environment.
3) Seamless integration of the above methods with MPC to provide collision avoidance and shared manipulation capabilities.

The proposed architecture is evaluated by performing telemanipulation using UR5 manipulator and VR setup over a standard internet connection, and a pilot user study is performed to evaluate the efficacy of the proposed method in terms of task completion times.

A detailed description of the proposed system architecture is given in Sec. II, followed by evaluation in Sec. III. The pilot comparison study of the proposed controller is described in Sec. III-E, followed by a conclusion in Sec. IV.

II. SYSTEM ARCHITECTURE

The proposed architecture consists of a motion capture system and an Extended Kalman filter (EKF) to estimate model states on the operator's side. The robot's side consists of a motion prediction model along with delay compensation, goal identification, and MPC. Only stereoscopic visual feedback in terms of video captured using a fixed camera mounted in the remote environment is presented to the operator. The complete architecture of the system is shown in Fig. 1.

A. Motion Capture

A motion capture setup is used to extract position and orientation corresponding to the operator's arm movements, using extrinsic sensing such as RGB cameras, depth cameras, or virtual reality (VR) setup.

In this study, we use a VR setup (HTC Vive) to capture the operator's arm position and orientation with respect to the head-mounted display, and a binary toggle switch is used to trigger the gripping action.

B. Relative Kinematic Model

In order to mitigate the communication channel delay in the forward path, we employ a model-mediated strategy [30], where a state estimation mechanism on the operator's side is used to determine states of the kinematic model for predicting the operator's motion on the robot side. In this approach, we transmit only the state information along with the time stamp over the network to a remotely located robot, where it is used to reproduce the operator's trajectory using the robot's time reference to compensate for any delay in the network.

For this purpose we use a relative kinematic model as shown in Fig. 2, with frames $\{\mathbf{S},\mathbf{E},\mathbf{W}\}$ corresponding to shoulder, elbow and wrist respectively, and defined by Eq. (1) and (2), corresponding to the relative motion of the elbow with respect to the shoulder, and wrist with respect to elbow.

$$\begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} = \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} + \begin{bmatrix} r_s\cos(\phi_s)\cos(\theta_s) \\ r_s\cos(\phi_s)\sin(\theta_s) \\ r_s\sin(\phi_s) \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = \begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} + \begin{bmatrix} r_e\cos(\phi_e)\cos(\theta_e) \\ r_e\cos(\phi_e)\sin(\theta_e) \\ r_e\sin(\phi_e) \end{bmatrix} \tag{2}$$

where, $(\cdot)_s$, $(\cdot)_e$ and $(\cdot)_w$ correspond to quantities related to the shoulder, elbow and wrist joint frames $\{\mathbf{S},\mathbf{E},\mathbf{W}\}$ respectively, $\theta_{(\cdot)}$ and $\phi_{(\cdot)}$ corresponds the the azimuth and elevation angles respectively, $r_{(\cdot)}$ represents the link lengths.

The rate of change of $\theta_{(\cdot)}$ and $\phi_{(\cdot)}$ for individual pair of joints is determined by the kinematic equations defined by

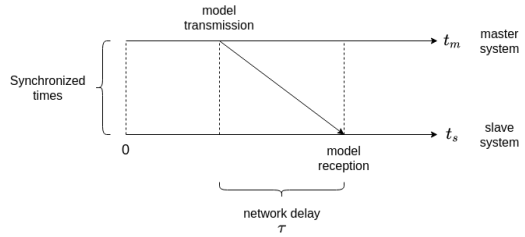


Fig. 3: Schematic of time delay

Eq. (3) and Eq. (4) respectively.

$$\begin{aligned} V_\theta =& r_{(.)}\dot{\theta}_{(.)}\cos(\phi_{(.)}) \\ =& V_B\cos(\eta)\sin(\mu-\theta_{(.)}) - V_A\cos(\alpha)\sin(\beta-\theta_{(.)}) \end{aligned} \tag{3}$$

$$\begin{aligned} V_\phi =& r_{(.)}\dot{\phi}_{(.)} \\ =& V_B\{-\cos(\eta)\sin(\phi_{(.)})\cos(\mu-\theta_{(.)}) + \sin(\eta)\cos(\phi_{(.)})\} \\ & - V_A\{-\cos(\alpha)\sin(\phi_{(.)})\cos(\beta-\theta_{(.)}) + \sin(\alpha)\cos(\phi_{(.)})\} \end{aligned} \tag{4}$$

where, V_A and V_B are the magnitudes of velocities of consecutive joints (that is for shoulder-elbow pair: V_A is the velocity of shoulder and V_B represents elbow velocity, and likewise for elbow-wrist pair), with (β,α) and (μ,η) being their respective azimuth and elevation angles.

C. State Estimation and Transmission

State estimates of the operator's position and velocity obtained from motion capture are used as measurements in an extended Kalman filter (EKF) with system model defined by Eq. (3) and Eq. (4), to obtain filtered estimates of model states $\mathbf{X}_T(t) = [\theta_s, \phi_s, \dot{\theta}_s, \dot{\phi}_s, \theta_e, \phi_e, \dot{\theta}_e, \dot{\phi}_e]^T$, which are further transmitted to the robot.

On the remotely located robot side, the model states $\mathbf{X}_T(t-\tau)$ are received after a delay of τ sec and are used to smoothly update the remote model states $\mathbf{X}_R(t-\tau)$ as $\mathbf{X}_R(t-\tau) := \mathbf{X}_R(t-\tau) + \lambda(\mathbf{X}_T(t-\tau) - \mathbf{X}_R(t-\tau))$, where $\lambda < 1$ prevents jerky motions which may otherwise arise if a hard update is used.

D. Model Propagation and Delay Compensation

Prediction of the operator's trajectory is generated by propagating the kinematic equations (3) and (4) with the newly updated states $\mathbf{X}_R$ and converting the obtained angular trajectories to wrist positions using Eq. (2) and (1). Since the reception of model parameters is inevitably affected by the latency of τ sec in the network, it is required to compensate the state for the network delay. We achive this compensation by propagating the delayed states from the delayed time to the robot's current time t.

$$\hat{\mathbf{X}}_R(t) = \int_{t-\tau}^{t} f(\mathbf{X}_R(t-\tau))dt \tag{5}$$

where, $f(.)$ denotes the state transition function for propagating the obtained states $\mathbf{X}_R(t-\tau)$. This state provides the estimate of the position where an operator would most likely reach at the current time. This process is depicted in Fig. 3. This delay compensated state is further supplied to the goal prediction algorithm to determine the operator's intended target.

E. Goal Prediction

A goal is considered the position the operator is trying to reach, and it can be the position of the object that the operator may try to pick or the position of a placement point. We also include the terminal position in the operator's predicted trajectory as one of the goal corresponding to teleoperation mode.

A scoring function is designed utilizing operator's distance and the instantaneous velocity with respect to the individual goals as follows:

$$G := \{\mathbf{g}_1,\ \mathbf{g}_2,\ \ldots,\ \mathbf{g}_N\} \tag{6}$$

$$s_i := \left(\frac{k}{(1 - {}^{\mathbf{o}}\hat{\mathbf{x}}.\overline{\mathbf{g}_i} + \delta)^2} \frac{1}{(1+e^{-\gamma(||{}^{\mathbf{o}}\hat{\mathbf{x}}||^2 - \beta))}} \right) e^{-w||\hat{\mathbf{x}} - \mathbf{g}_i||^2} \tag{7}$$

where, $\mathbf{g_i}$ is i^{th} goal in goal set G consisting of N goals, $[{}^o\hat{\mathbf{x}}, \hat{\dot{\mathbf{x}}}]$ represents the estimate of operators position and velocity computed using the estimated operator's states $\hat{\mathbf{X}}_R(t)$ in the kinematic model given in eq. 2 , k, w, γ and β are tuning parameters, $\delta << 1$ is a scalar constant used to prevent the denominator from going to zero, and s_i is the score for goal g_i. $\overline{(.)}$ is used to denote unit vectors of the respective vectors. The first term in the scoring function corresponds to the velocity contribution to the score, while the term $e^{-||{}^o\mathbf{x} - \mathbf{g}_i||^2}$ corresponds to the distance contribution. The velocity contribution is designed such that a higher score is allocated to the goal towards which the velocity vector is directed. The contribution of this score to total component is also adjusted based on the magnitude of the velocity, with it's value going to unity as velocity magnitude goes to zero, allowing only distance score to contribute to the final score.

Let $\mathbf{S}$ represent a set of scores for all the goals g_i in the current iteration.

$$\mathbf{S} := \begin{bmatrix} s_1 & s_2 & \ldots & s_N \end{bmatrix}^T$$

Due to the operator's motion and noise in measurements, the scores associated with the individual goal may vary in every iteration, which may lead to fluctuating goals if chosen purely based on maximum score. In order to prevent this, a confidence metric is computed for individual goals recursively as an expectation of goal occurrences.

For computing a goal's confidence, we select the goal with the highest score as the candidate, and a one-hot encoded vector with ones corresponding to the identified goal's index is used for updating the confidence using the following recursion:

$$\mathbf{H} := \begin{bmatrix} h_1 & h2 & \ldots & h_N \end{bmatrix} \qquad h_i = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

Confidence,

$$\mathbf{C}_0 = \begin{bmatrix} 0 & 0 & \ldots & 0 \end{bmatrix}$$
$$\mathbf{C}_t = \mathbf{C}_{t-1} + \alpha\,(\mathbf{H} - \mathbf{C}_{t-1})$$

where, $(.)_t$ corresponds to t^{th} iteration. When the confidence for any goal exceeds 80%, it is passed on to the control algorithm for an autonomous approach to its position.

There exists a special case when the operator's velocity is below a certain threshold κ, which causes the terminal state of predicted trajectory (operators goal) to lie very close to the current operator's position leading to high score. In such a situation, the operators goal score s_o is excluded from the score set $\mathbf{S}$

if $||^{o}\dot{\mathbf{x}}||^2 \geq \kappa$ **then**
  $j = \text{argmax}\{\mathbf{S}\}$
else
  $j = \text{argmax}\{\mathbf{S} \setminus s_o\}$
end if

F. Model Predictive Control

An optimal control problem is formulated in order to determine the robot's joint position and velocity trajectory that minimizes the error in the Cartesian position and orientation of the end-effector with respect to that of the operator's palm position and orientation. Additionally, constraints are added to avoid the collision of the robot arm with the table and to avoid entering singular configurations.

1) Model Used: The joint angle dynamics are modelled using a double integrator system representing damped system dynamics. This choice is made due to the presence of an inner loop controller in the robot for tracking the reference joint angles provided. The states for the system are taken as joint angles and angular rates $[\theta, \dot{\theta}]^T \in \mathbb{R}^{12\times 1}$ and angular accelerations are taken as the control input $\mathbf{u} = \ddot{\theta}$.

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{I} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix} \mathbf{u} \tag{8}$$

2) Cost Function: A quadratic cost function with the aim to minimize the error in position and orientation of the end-effector with respect to that of the wrist of the operator is used. Additionally, a cost corresponding to joint angular rate is included to minimize the speed of the robot motion, thus reducing possible damage to the robot. In order to penalise large control actions, a quadratic cost on control inputs is also added.

$$J(\theta, \mathbf{u}) = \sum_{j=0}^{N} \mathbf{e}_j^T \mathbf{Q_a} \mathbf{e}_k + \dot{\theta}_j^T \mathbf{Q_b} \dot{\theta}_j + \mathbf{u}_j^T \mathbf{R} \mathbf{u}_j \tag{9}$$

where, $\mathbf{e} = [\mathbf{x}_j - \hat{\mathbf{x}}(\theta)_j, \mathbf{q}_j - \hat{\mathbf{q}}(\theta)_j]^T \in \mathbb{R}^{7\times 1}$ represents error in position and orientation, with $\mathbf{x}$ representing the position and $\mathbf{q}$ representing quaternion orientation of the operator palm during pure teleoperation and goal when confidence in goal exceeds the threshold. $\hat{(.)}(\theta)$ represents the corresponding quantity obtained using the forward kinematics of the robot. $\mathbf{Q_a} \in \mathbb{R}^{7\times 7}$, $\mathbf{Q_b} \in \mathbb{R}^{6\times 6}$ and $\mathbf{R} \in \mathbb{R}^{6\times 6}$ are the weighing matrices for the errors, joint rates and control inputs respectively, with $\mathbf{Q_a}$ and $\mathbf{Q_b}$ being symmetric positive semidefinite and $\mathbf{R}$ being symmetric positive definite.

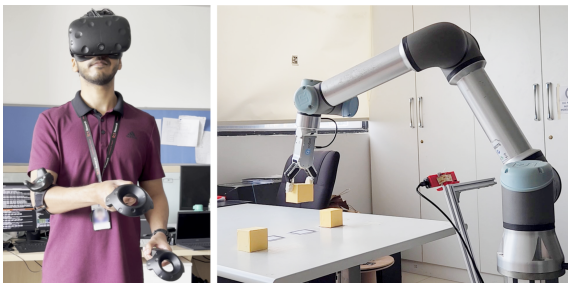


Fig. 4: Experiment setup and robot configuration

3) Singularity Constraint: At singularities, the Jacobian relating the Cartesian velocities and angular velocity of the end-effector to the joint angular velocities becomes singular, which can lead to significant joint rates causing unexpectedly fast motion of the robot. In order to prevent MPC from providing solutions that correspond to singular configurations, a constraint of the form

$$\det(J(\theta)J^T(\theta))^{1/2} > \varepsilon \tag{10}$$

is used, where, $J(\theta)$ is the Jacobian matrix and $\varepsilon < 1$ is an empirically chosen positive constant.

4) Collision Avoidance Constraint: In order to prevent collisions with large planar objects such as a table, an affine constraint of the form

$$\mathbf{An} + \mathbf{b} \geq 0 \tag{11}$$

is used. Where $\mathbf{n}$ is the unit normal to the plane with which collision is to be avoided, $\mathbf{A}$ is the matrix with rows as points of interest $\hat{\mathbf{x}}_\mathbf{i}(\theta)$ on the body of the robot that should avoid collision with the plane. Similarly, to avoid collision with smaller objects, an ellipsoidal constraint is used to restrict the motion of the point of interest $\hat{\mathbf{x}}_\mathbf{i}(\theta)$ from entering the enclosed ellipsoid around objects in the workspace. This constraint can also be used to prevent self-collision by appropriately choosing reference points on the robot's body.

$$(\hat{\mathbf{x}}_i(\theta) - \mathbf{x}_{ref})^T \mathbf{RMR}^T (\hat{\mathbf{x}}_i(\theta) - \mathbf{x}_{ref}) > \alpha \tag{12}$$

where $\mathbf{M}$ is a diagonal matrix and $\mathbf{R}$ is the rotation matrix defining the orientation of the ellipsoid, onstra$\mathbf{x}_{ref}$ is the reference position around which the ellipsoidal constraint is fitted, α is chosen empirically to enclose the object in the ellipsoid

5) Joint Constraint for Desired Robot Configuration: For a given end-effector position and orientation, there exist multiple joint angle configurations, and the solutions obtained on solving the optimal control problem, which depend on the initial condition provided to the solver and may correspond to undesirable robot postures. In order to make the robot take up a desirable posture, the joint angles of the robot are constrained to stay within specified limits using the bound constraints as,

$$\theta_i^{min} \leq \theta_i \leq \theta_i^{max} \tag{13}$$

where θ_i is the desired value for i^{th} joint.

G. Baseline: Inverse Kinematic Controller

As a baseline controller for comparing the performance, an inverse kinematic (IK) solver is used for computing reference joint angles for the manipulator. Depending on the position and orientation, IK provides multiple solutions. In order to maintain continuity, it is required to select a solution that is closest to the current joint configuration. It is also required for the robot to attain a suitable posture/configuration based on the task at hand, such as manipulation on the table. In order to achieve these goals, we use weighted least squares to select reference joint positions from the set of solutions obtained from the IK solver for desired position and orientation of the end-effector.

$$\theta = arg\min_{\theta} \sum_{i}^{N} ||\theta - \theta_l||^2_{\mathbf{W}_l} + ||\theta - \theta_d||^2_{\mathbf{W}_d} \qquad (14)$$

where, $||\cdot||_{\mathbf{W}_i}$ represents weighted norm of difference in the IK solution θ and the last solution θ_l and desired joint configuration θ_d respectively with weight matrix $\mathbf{W}_i$ and $\mathbf{W}_d$.

For autonomously traversing to the goal, a minimum jerk trajectory is planned from the end-effector to the goal at every iteration, and only the first element of the trajectory is used in IK during execution.

III. EVALUATION

For validating the proposed architecture, an evaluation of a block manipulation task was performed over an internet network.The task comprised picking and placing three square blocks in a pre-specified order at designated places. Universal Robot UR5e equipped with a Robotiq 2F-85 two-finger gripper was used to perform the task as shown in Fig. 4.

For motion capture, HTC Vive VR system was used. The operator's right arm's elbow and wrist positions were extracted using the elbow-mounted and handheld trackers.

Parameters used in the evaluation are as follows: soft-update constant α was set to 0.5, and the parameters β and γ were taken to be 0.2 and 30, respectively, k and w were set to 1 in the goal scoring function, δ for the operator terminal goal was set to 10^{-2}, while for other goals it was set to zero. $Q_a = diag([5,5,5,1,1,1,1])$, $Q_b = (0.001)I_6$, $R = (0.001)I_6$ in MPC, while ε was taken to be 10^{-3}. The joint limits were set to $\theta_{min} = [-1.2,-1.6,-3.14,-1.2,1.2,-3.14]$ and $\theta_{max} = [-1.4,-1.8,3.14,-1.4,1.4,3.14]$. All the constraints were treated as soft constraints to prevent solvers from failing due to infeasible solutions. MPC was implemented using the ACADO toolkit [31] with a time horizon of 2 sec and discritization step size of 0.1 sec.

A. Network Setup

The network architecture used the Robot Operating System (ROS) in a Master-Slave configuration deployed over an Amazon Web Services (AWS) cloud platform. To allow remote connection between ROS master and ROS slave securely, Virtual Private Network (VPN) setup is used by running an OpenVPN server on a Virtual Private Cloud (VPC). This architecture provides complete control over the network and eliminates the usage of public IP.

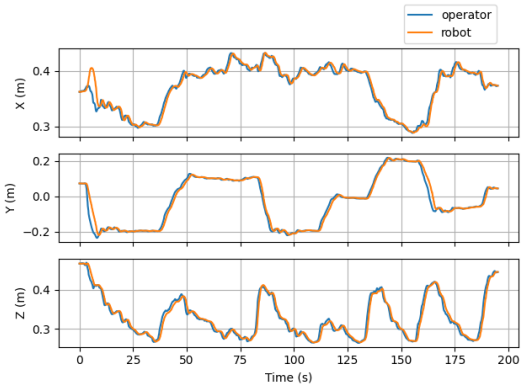


Fig. 5: Operator and robot trajectories for MPC with delay compensation

The operator is provided with real-time stereoscopic video captured on the robot side that is shared using a WebRTC-based video streaming solution. A GStreamer-based video streaming framework that takes a live camera input as the video source, which is further encoded into a WebRTC compatible format, allowing remote peers to receive the video in real-time.

B. Delay Compensation

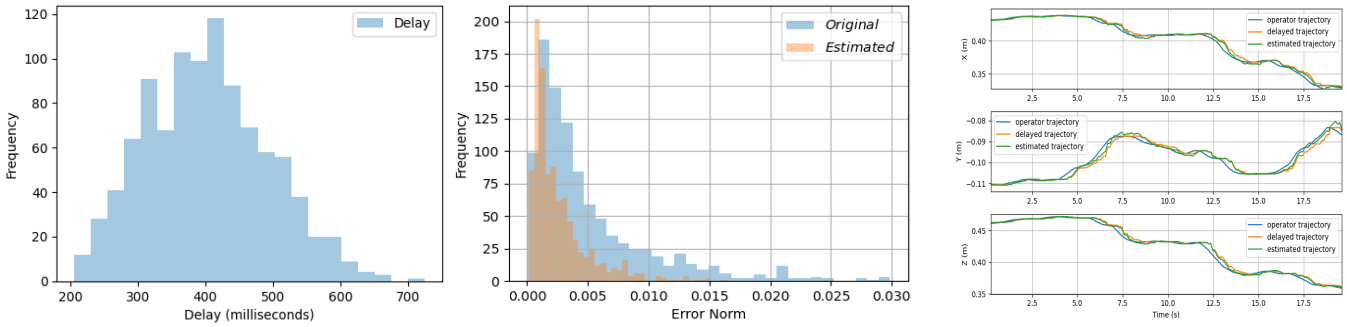
Fig. 5 shows the trajectory plots of commanded input and the trajectory followed by the robot using MPC. It can be seen that MPC is able to follow the commanded trajectories. The histogram plot in Fig. 6a shows the delay in the collected samples. It can be seen that a delays of magnitude of 400ms were experienced most frequently in the communication channel. Similarly, the histogram plot in Fig. 6b shows the norm of the error in the operator's wrist position across the network in the presence of delay (labelled as original), against the error in the predicted operator's position on the robot's side. It can be seen from the plot that the error corresponding to the estimated operator's position are less frequent at higher magnitudes, and this can also be seen in Fig. 6c, showing a zoomed-in view of the trajectory.

C. Constraint Satisfaction

For testing the collision avoidance constraint, we restricted the end-effector to stay above the table surface while simultaneously avoiding the obstacle located at $[0.4,0,0.1]^T$. The plot in Fig. 7 shows the trajectory corresponding to the back-and-forth motion of the operator along the y-axis across the obstacle. It can be seen from the z-axis trajectories that the end-effector successfully manoeuvres over the obstacle while safely avoiding it and also maintains a constant offset of approximately 0.05m above the table.

D. Effect of Shared Control

To analyse the efficacy of the proposed shared control approach, same experiment was performed using pure tele-operation mode. The corresponding operator commanded trajectory was further evaluated with the shared control approach in moving towards the goal. Fig. 8 shows the variation of the distance of the end-effector in both cases



(a) Experienced network delay **(b)** Error norm in delayed & estimated trajectory **(c)** Zoomed delayed & estimated trajectory

Fig. 6: Delay compensation using trajectory re-sampling

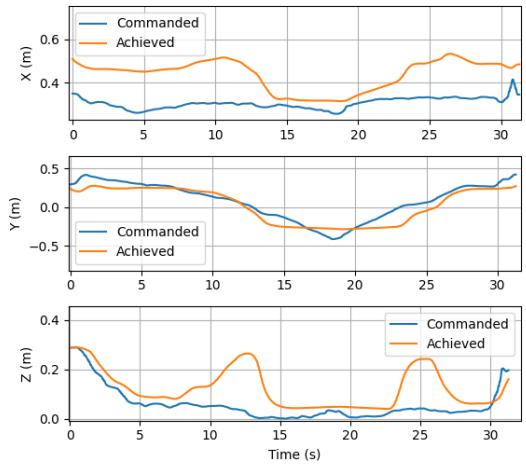


Fig. 7: Trajectory while satisfying the collision constraints

during the picking operation. It can be seen from the plot that the end-effector moves to the intended goal much earlier in the shared control case compared to the pure teleoperation mode, in which the operator has to move with fine motions to reach the goal. On an average it was observed that the pure teleoperation case required approximately 115 sec to complete the experiment while shared control took 74 sec.

For the shared control, the first approach to the goal occurs due to the operator's motion toward the goal, which can be observed from a steadily decreasing distance. When the operator's approach velocity towards the goal decreases, the confidence in the operator's terminal goal increases and the control switches back to the operator. But as the distance between the operator and the goal decreases, the confidence in the goal increases, and control switches back to the autonomous mode, taking the robot to the goal position. The difference in the trajectories of the two modes during operator tracking is because the shared control tracks the terminal state of the predicted trajectory, while pure teleoperation follows the current position of the operator.

E. Comparison User Study

The previous evaluation involved a trained operator using the complete proposed architecture. In order to validate the benefits of the proposed method over baselines, a pilot user study was conducted involving 10 novice operators. Each

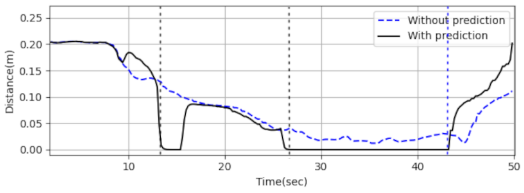


Fig. 8: Variation of the distance between the goal and end-effector with and without goal prediction

operator was asked to perform the block placement task (Fig. 4). The study had a 2×2 design, with the teleoperation system running one of the controllers (IK or MPC), along with model propagation for delay compensation being active or inactive (MP or NMP). Each operator's performance was evaluated using the time taken to complete the task.

TABLE I: Summary of Trial Times in Box Placement (s)

Condition	Mean	Std. Dev.
IK, NMP	70.81	12.26
IK, MP	70.53	12.80
MPC, NMP	78.92	21.67
MPC, MP	72.25	14.62

As shown in Table I, model propagation was slightly faster on average with both control methods. Though MPC was slower than IK, it incorporated collision avoidance constraints leading to safer operation. A two-way ANOVA with the Bonferroni correction was used to compare the factors of the control method (IK, MPC) and the presence of model propagation (NMP, MP). These factors were not found to be statistically significant in their effect on the reduction of trial times in the pilot study, with p = 0.48 for MPC vs IK and p = 0.33 for NMP vs MP. However, the trends indicate that a more extensive user study in the future may show greater effects of these factors on operator performance.

IV. CONCLUSION

In this work, we demonstrated a teleoperation architecture involving a model-mediated approach for delay compensation using operator motion prediction with a relative kinematic model and shared control through goal identification. It was verified through an evaluation procedure that the proposed motion prediction approach and shared control helped shorten the required task completion time.

An MPC formulation used was able to compute optimal joint angle references while satisfying the imposed constraints leading to collision-free motion. A pilot comparison indicated slightly better performance characteristics compared to a conventional IK solver-based approach, along with added benefits of collision avoidance.

For future work, we plan to improve the goal prediction algorithm using an adaptive technique for tuning the scoring function parameters and further validate the complete architecture with a larger user study. In addition, we plan to incorporate a predictive display feature to provide delay-free visual feedback to the operator.

REFERENCES

[1] M. Hwang, B. Thananjeyan, D. Seita, J. Ichnowski, S. Paradis, D. Fer, T. Low, and K. Goldberg, "Superhuman surgical peg transfer using depth-sensing and deep recurrent neural networks," arXiv preprint arXiv:2012.12844, 2020.

[2] K. Schmeckpeper, O. Rybkin, K. Daniilidis, S. Levine, and C. Finn, "Reinforcement learning with videos: Combining offline observations with interaction," arXiv preprint arXiv:2011.06507, 2020.

[3] A. Tung, J. Wong, A. Mandlekar, R. Martín-Martín, Y. Zhu, L. Fei-Fei, and S. Savarese, "Learning multi-arm manipulation through collaborative teleoperation," arXiv preprint arXiv:2012.06738, 2020.

[4] N. Marturi, A. Rastegarpanah, C. Takahashi, M. Adjigble, R. Stolkin, S. Zurek, M. Kopicki, M. Talha, J. A. Kuo, and Y. Bekiroglu, "Towards advanced robotic manipulation for nuclear decommissioning: A pilot study on tele-operation and autonomy," in 2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA). IEEE, 2016, pp. 1–8.

[5] G. Feng, W. Li, and H. Zhang, "Space robot teleoperation experiment and system evaluation method," in 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC). IEEE, 2018, pp. 346–351.

[6] J. Guo, C. Liu, and P. Poignet, "A scaled bilateral teleoperation system for robotic-assisted surgery with time delay," Journal of Intelligent and Robotic Systems, vol. 95, no. 1, pp. 165–192, 2019.

[7] T. Wang, Y. Li, J. Zhang, and Y. Zhang, "A novel bilateral impedance controls for underwater tele-operation systems," Applied Soft Computing, p. 106194, 2020.

[8] P. Duan, Z. Duan, S. Li, and Y. Chen, "Motion prediction and delay compensation for improved teleoperation of an exoskeletal robot," Advances in Mechanical Engineering, vol. 10, no. 2, p. 1687814018760353, 2018.

[9] Y. Deng, Y. Tang, B. Yang, W. Zheng, S. Liu, and C. Liu, "A review of bilateral teleoperation control strategies with soft environment," in 2021 6th IEEE International Conference on Advanced Robotics and Mechatronics (ICARM), 2021, pp. 459–464.

[10] S. N. F. Nahri, S. Du, and B. J. Van Wyk, "A review on haptic bilateral teleoperation systems - journal of intelligent and robotic systems," Dec 2021. [Online]. Available: https://link.springer.com/article/10.1007/s10846-021-01523-x

[11] R. J. Anderson and M. W. Spong, "Bilateral control of teleoperators with time delay," IEEE Transactions on Automatic Control, vol. 34, no. 5, pp. 494–501, 1989.

[12] G. Niemeyer and J.-J. Slotine, "Stable adaptive teleoperation," IEEE Journal of oceanic engineering, vol. 16, no. 1, pp. 152–162, 1991.

[13] D. P. Losey, C. G. McDonald, E. Battaglia, and M. K. O'Malley, "A review of intent detection, arbitration, and communication aspects of shared control for physical human–robot interaction," Applied Mechanics Reviews, vol. 70, no. 1, 2018.

[14] M. Laghi, M. Maimeri, M. Marchand, C. Leparoux, M. Catalano, A. Ajoudani, and A. Bicchi, "Shared-autonomy control for intuitive bimanual tele-manipulation," 11 2018.

[15] G. De Rossi, M. Minelli, A. Sozzi, N. Piccinelli, F. Ferraguti, F. Setti, M. Bonfé, C. Secchi, and R. Muradore, "Cognitive robotic architecture for semi-autonomous execution of manipulation tasks in a surgical environment," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019, pp. 7827–7833.

[16] F. Stroppa, M. Selvaggio, N. Agharese, L. H. Blumenschein, E. W. Hawkes, A. M. Okamura, et al., "Shared-control teleoperation paradigms on a soft growing robot manipulator," arXiv preprint arXiv:2108.00677, 2021.

[17] M. J. A. Zeestraten, I. Havoutis, and S. Calinon, "Programming by demonstration for shared control with an application in teleoperation," IEEE Robotics and Automation Letters, vol. 3, no. 3, pp. 1848–1855, 2018.

[18] H. J. Jeon, D. P. Losey, and D. Sadigh, "Shared autonomy with learned latent actions," arXiv preprint arXiv:2005.03210, 2020.

[19] S. Javdani, H. Admoni, S. Pellegrinelli, S. S. Srinivasa, and J. A. Bagnell, "Shared autonomy via hindsight optimization for teleoperation and teaming," The International Journal of Robotics Research, vol. 37, no. 7, pp. 717–742, 2018.

[20] C. Liu, Adaptive Control of Teleoperation Systems with Uncertainties: A Survey. New York, NY, USA: Association for Computing Machinery, 2021, p. 5–9. [Online]. Available: https://doi.org/10.1145/3475851.3475856

[21] O. Tokatli, P. Das, R. Nath, L. Pangione, A. Altobelli, G. Burroughes, E. T. Jonasson, M. F. Turner, and R. Skilton, "Robot-assisted glovebox teleoperation for nuclear industry," Robotics, vol. 10, no. 3, p. 85, 2021.

[22] V. Girbés-Juan, V. Schettino, Y. Demiris, and J. Tornero, "Haptic and visual feedback assistance for dual-arm robot teleoperation in surface conditioning tasks," IEEE Transactions on Haptics, vol. 14, no. 1, pp. 44–56, 2020.

[23] J. E. Solanes, A. Muñoz, L. Gracia, A. Martí, V. Girbés-Juan, and J. Tornero, "Teleoperation of industrial robot manipulators based on augmented reality," The International Journal of Advanced Manufacturing Technology, vol. 111, no. 3, pp. 1077–1097, 2020.

[24] E. Triantafyllidis and Z. Li, "Considerations and challenges of measuring operator performance in telepresence and teleoperation entailing mixed reality technologies," arXiv preprint arXiv:2103.12702, 2021.

[25] L. Penco, N. Scianca, V. Modugno, L. Lanari, G. Oriolo, and S. Ivaldi, "A multimode teleoperation framework for humanoid loco-manipulation: An application for the icub robot," IEEE Robotics & Automation Magazine, vol. 26, no. 4, pp. 73–82, 2019.

[26] H. Arai, H. Nagakura, and Y. Uchimura, "Variable-horizon based model predictive control for tele-operation with time-varying delay," in 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE). IEEE, 2020, pp. 586–591.

[27] N. Piccinelli and R. Muradore, "A passivity-based bilateral teleoperation architecture using distributed nonlinear model predictive control," in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020, pp. 11 466–11 472.

[28] S. Kleff, A. Meduri, R. Budhiraja, N. Mansard, and L. Righetti, "High-frequency nonlinear model predictive control of a manipulator," in 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021.

[29] M. Rubagotti, T. Taunyazov, B. Omarali, and A. Shintemirov, "Semi-autonomous robot teleoperation with obstacle avoidance via model predictive control," IEEE Robotics and Automation Letters, vol. 4, no. 3, pp. 2746–2753, 2019.

[30] X. Xu, B. Cizmeci, C. Schuwerk, and E. Steinbach, "Model-mediated teleoperation: toward stable and transparent teleoperation systems," IEEE Access, vol. 4, pp. 425–449, 2016.

[31] B. Houska, H. J. Ferreau, and M. Diehl, "Acado toolkit—an open-source framework for automatic control and dynamic optimization," Optimal Control Applications and Methods, vol. 32, no. 3, pp. 298–312, 2011.