

Model-Reference Model-Mediated Control for Time-Delayed Teleoperation Systems

Leonam S. D. Pecly¹, Marcelo L. O. Souza², and Keyvan Hashtrudi-Zaad³

Abstract—Time delays in bilateral teleoperation systems can degrade the transparency and result in instability. For large delays, model predictive control or impedance reflective methods in which the operator interacts with the slave and environment model, can help overcome this issue. An important key to the success of these systems is to have the local virtual slave model synchronized with the slave. This is especially important when the slave gets into contact with hard environments, resulting in rapid contact force build up and slave chattering. In this paper, we propose a model-reference model-mediated teleoperation control architecture that provides enhanced synchronization between the slave and virtual slave models and keeps both the master and slave responsive. The proposed architecture is evaluated against through simulations and experiments carried out on a 1-DOF master-slave testbed.

I. INTRODUCTION

Teleoperation systems are designed to extend the manipulation capability of users to remote or dangerous environments. Every teleoperation system experience some delays due to the processing time and the communication medium. Delay can degrade task performance, distort transparency and cause instability, especially when dealing with hard environments [1], [2], [3], [4].

These poor performance situations usually are not experienced in systems operating in the same room or in a short distance using a dedicated transmission link, such as in surgeries. However, the round-trip delays in the communications through the Internet or in orbital space applications can amount to hundreds of milliseconds. The source of instability has traced back to the non-passivity of the communication channel [5]. In order to minimize the undesirable effects of delay, passivity-based methods such as wave variables or time domain passivity have been proposed. In the former, wave variables instead of power variables are transmitted [5], [6]. In time-domain passivity and its variants, a passivity observer on the fly determines the energy flow through the master-slave network and dissipates the excess of energy through dampers that the passivity controller places at both side of the network [7], [8].

¹Leonam da Silva Direito Pecly is a former student from Space Engineering and Technology, Division of Space Mechanics and Control, National Institute for Space Research-INPE, São José dos Campos, São Paulo, Brazil. leonam.pecly@gmail.com

²Marcelo Lopes de Oliveira e Souza is with Space Engineering and Technology, Division of Space Mechanics and Control, National Institute for Space Research-INPE, São José dos Campos, São Paulo, Brazil. marcelosouzabr@ig.com.br

³Keyvan Hashtrudi-Zaad is with the Department of Electrical and Computer Engineering, Queen's University, Kingston, ON K7L3N6, Canada. khz@queensu.ca

Model-mediated teleoperation (MMT) control architectures have been proposed to address large amount of delays by making the operator interacts locally with virtual slave and environment, providing force feedback without delay. The dynamic parameters of the environment model are transmitted from the remote to local sites instead of force data. MMT covers architectures with similar concepts such as predictive control, model-based or impedance reflecting [9], [10], [11], [12]. It also includes the exchange of alternative type of information than model parameters [13].

In this paper, we introduce a model-reference model-mediated control architecture for systems with significant amount of constant time delays. In this architecture, not only the master is in interaction with local virtual slave and environment, but also the slave receives control command from a remote virtual slave and environment that is local to the slave. In other words, the virtual slave acts a reference model for the slave. In this way, both the master and slave are responsive. We also propose the use of environment rest position at the virtual environments that is available through proximity sensor such a laser rangefinder. The proposed architecture is evaluated through simulations and preliminary experimental results obtained from a 1-DOF master-slave testbed interacting with an environment made of a bank of springs.

In Section II, a brief review of the conventional two channel position-force control architecture and the impedance reflecting benchmark control architecture [11] will be provided. The proposed control architecture will be presented next. Section III describes the simulation and experimental procedures, and the analysis of results. Finally, conclusions will be drawn in Section IV.

II. METHODOLOGY

In this section, we will first present and discuss the impedance reflecting bilateral control architecture presented in [14]. Then we will propose our methodology in order to mitigate the issue of synchronization between the local and the remote sides.

A. Impedance Reflecting Position-Force Architecture

Consider the two-channel position-force (P-F) bilateral control architecture in Figure 1-a, where the operator's hand position and environment contact force are exchanged through the communication channel, and X_h , X_s , F_e and F_h denote the Laplace transforms of the master and slave positions, the environment force measured by the slave, and the operator force on the master. Here the master and slave

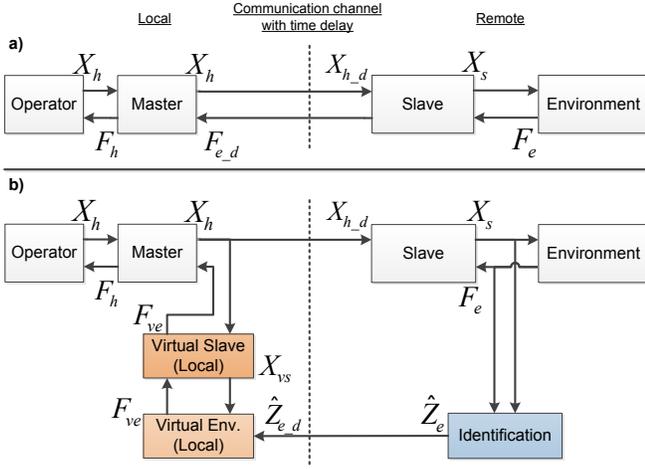


Fig. 1: Block diagrams of a) conventional P-F; and b) impedance reflecting P-F control architecture.

dynamics are modeled by a linear-time-invariant (LTI) mass-damper system as

$$\begin{aligned} F_{cm} + F_h &= (M_m s^2 + B_m s) X_h \\ F_{cs} - F_e &= (M_s s^2 + B_s s) X_s \end{aligned} \quad (1)$$

where M_m , M_s , B_m , B_s are the master and slave mass and damping, and F_{cm} and F_{cs} are the master and slave control commands, defined as

$$\begin{aligned} F_{cm} &= K_F F_{e,d} \\ F_{cs} &= K_P (X_{h,d} - X_s) - K_D X_s s \end{aligned} \quad (2)$$

The variables $X_{h,d}$ and $F_{e,d}$ are the delayed operator's hand position and environment force, and K_P , K_D and K_F denote the controller gains.

As the time delay (T_d) present in the communication channel increases, the system becomes less responsive due to the delayed force feedback until it reaches an inoperable point where the operator experiences undesired oscillations.

Figure 1-b shows an impedance reflecting control architecture proposed to overcome the above problem by implementing the dynamic model of the slave and environment at the master (local) side. In this way, the operator is in agile interaction with the local virtual environment by receiving the virtual environment force F_{ve} instead of the delayed environment force $F_{e,d}$. To this purpose, the environment impedance is identified and transmitted from the remote to the local side. Considering that the variation of environment impedance is much less than the force, this architecture in certain way opens the delay loop in the force feedback for the operator, which makes this type of architecture more suitable for teleoperation systems under significant time delays. We will consider this architecture as a benchmark in this paper.

The environment dynamic model can cover a broad range of dynamics as long as it is linearly parameterized for identification. In this paper, we consider the Kelvin-Voigt (KV) model

$$F_e = \begin{cases} (K_e + B_e s) X_e & X_s \geq X_{e_0}, \\ 0 & X_s < X_{e_0}. \end{cases} \quad (3)$$

where $X_e = X_s - X_{e_0}$ is the slave end-effector contact penetration and X_{e_0} , K_e and B_e denote the environment surface rest position, stiffness and damping, respectively. A widely used time-domain identification method is Recursive Least Squares (RLS) formulated as

$$\begin{aligned} \mathbf{G}_k &= \mathbf{P}_{k-1} \Phi_k^T [\Phi_k \mathbf{P}_{k-1} \Phi_k^T + 1]^{-1} \\ \mathbf{P}_k &= \mathbf{P}_{k-1} - \mathbf{G}_k \Phi_k \mathbf{P}_{k-1} \\ \hat{\Theta}_k &= \hat{\Theta}_{k-1} + \mathbf{G}_k (\mathbf{F}_k - \Phi_k \hat{\Theta}_{k-1}) \end{aligned} \quad (4)$$

where \mathbf{G}_k and \mathbf{P}_k are the gain and covariance matrices, respectively, Φ_k is the regression vector, and $\hat{\Theta}_k$ is the vector of dynamic parameters to be estimated at every time step k . Some variations such as the Exponentially-Weighted-RLS (EWRLS) [15] and Self-Perturbing RLS (SPRLS) [16] have been proposed to deal with the time-varying nature of the dynamic parameters. Block LS with floating window is another method for rapid estimation in which the adjustment of the window length enables a trade-off between stability and fast convergence [17]. Since the simulation and experimental validation conducted in Section III do not consider time-varying systems, the online identification performed in this paper uses the original RLS method.

B. Proposed Control Architecture

Two major challenges that significantly affects the stability and transparency in the impedance-reflective control architecture (Figure 1-b) are: (i) the lack of synchronization between the virtual slave and the slave at local side, and (ii) delay in impedance transition in the virtual slave during intermittent contact with hard environments [11]. The latter has been mitigated by implementing a collision prediction algorithm that would switch between impedances one delay ahead of the collision. Although improved, the control architecture showed satisfactory performance for time delays of up to 200 ms. Above this limit, since the slave would only receive delayed command from the master, it would not react rapidly enough to changes in the environment.

Figure 2 shows the proposed architecture which is based on the impedance reflective control architecture, illustrated in Figure 1-b [11]. At the local side the two architectures are the same in that the operator interacts with the local virtual environment and the master is in a closed-loop with the local virtual slave. The master position is also sent to the remote side to be used as reference trajectory for the slave.

To improve on the issue of synchronization and the performance for larger time delays, the proposed control architecture places the slave in closed-loop with a virtual slave in interaction with a virtual environment at the remote side. Therefore, the slave tracks both the delayed operator command ($X_{h,d}$) and the virtual slave position (X_{vsr}) according to the control law

$$F_{cs} = K_P (X_{h,d} - X_s) - K_D X_s s + K_{PV} (X_{vsr} - X_s) \quad (5)$$

where K_{PV} represents the proportional gain. In other words, the slave follows the remote virtual slave, thus implementing a model-reference control on the slave. The controller of the remote virtual slave follows (2).

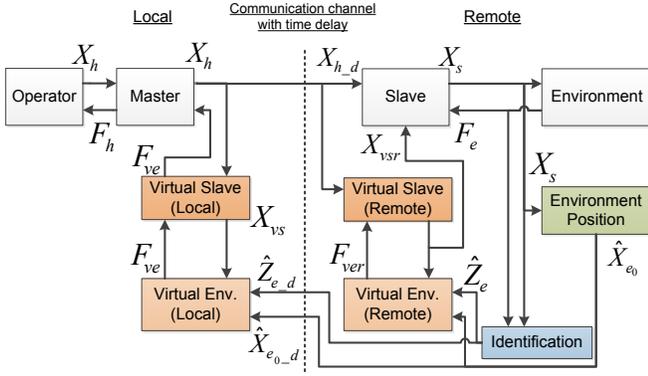


Fig. 2: The proposed model-reference model-mediated control architecture.

The remote virtual environment receives local updates with the environment dynamic parameters, resulting in an agile response of the remote virtual slave and thus slave to the changes in the environment. As an added benefit, this architecture helps to minimize the errors of modeling and identification of the slave robot since the slave tracks the virtual model position and, in a certain way, mimics its behavior.

In this architecture, it is also proposed to hold position correspondence between the environment and the virtual environments. As a result, the rest position of the environment (X_{e_0}) is computed at the remote side and sent to the local side with the help of a distance measuring sensor, such as laser rangefinder sensor. In this way, the "Environment Position" block computes the position (X_{e_0}) prior to contact using the data from the laser rangefinder (X_L), position of the slave robot (X_s), the last environment rest position computed (X_{e_0last}) and the distance between the end-effector and the laser (D_{ee}) according to (6), where $\varepsilon = 2 \times 10^{-3}$ m defines the distance from the environment that the computation stops.

$$X_{e_0} = \begin{cases} X_L + X_s - D_{ee} & , X_s \geq (X_{e_0last} - \varepsilon) \\ X_{e_0last} & , X_s < (X_{e_0last} - \varepsilon) \end{cases} \quad (6)$$

Figure 3 illustrates the schematic diagram by which the environment rest position is computed.

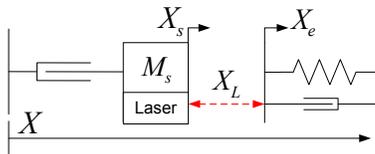


Fig. 3: Schematic diagram for environment rest position computation.

It is important to note that the virtual environment impedance must have a saturation limit to avoid destabilizing the interaction between the master and the local virtual slave. As at the onset of contact the environment impedance is

unknown, to protect the slave and environment and improve the system transparency it is proposed to use the impedance of saturation.

If more agility is required at the remote side, the remote virtual slave can be directly fed by the measured environment force F_e , rather than the virtual environment force F_{ver} . This modification (mode 1) is illustrated in Figure 4, where the red line highlights the difference from the originally proposed scheme. However, one should note that in this case during the convergence of environment identification, the movement of the remote virtual slave can be noticeably different from that of the local virtual slave.

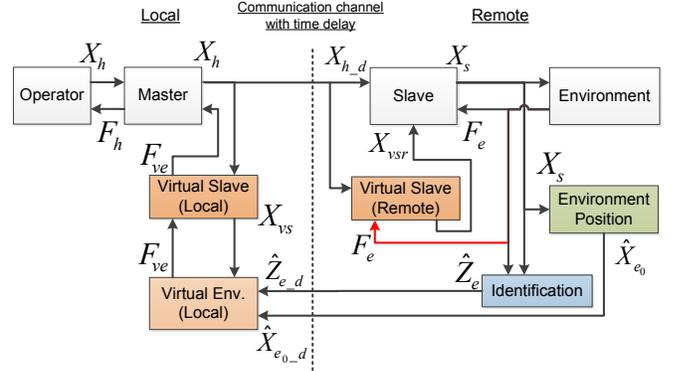


Fig. 4: The proposed model-reference model-mediated control architecture (mode 1).

After removal of the remote virtual environment, a question may arise with regard to the possibility of the removal of the remote virtual slave by feeding the local virtual slave position to the slave, as shown in Figure 5. This way the slave tracks the local virtual slave movement. The potential problem with this modified architecture (mode 2) is that both commands to the slave are delayed and thus the slave response will be delayed. The advantages and disadvantages of the proposed architecture and its modifications will be discussed through simulations in Section III-C.

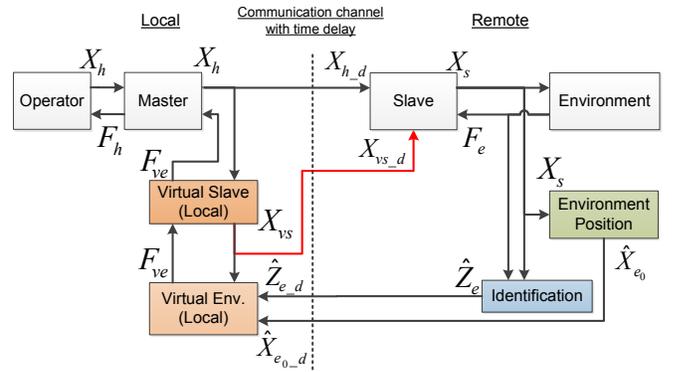


Fig. 5: The proposed model-reference model-mediated control architecture (mode 2).

III. SIMULATION AND EXPERIMENTAL VALIDATION

A. Experimental Setup and Simulation Procedure

In this section, the proposed architecture and its modified versions will be evaluated through simulations and preliminary experimental results. The impedance reflecting control architecture (Figure 1-b) will also be simulated as the benchmark.

Figure 6 shows the master-slave experimental setup consisting of a Quanser 3-DOF planar Twin-Pantograph master haptic interface and a 1-DOF linear stage as the slave robot. The Twin-Pantograph master is equipped with a 6-DOF ATI Nano-25 force/torque sensor at the end-effector and provides an approximate position resolution of 1.2×10^{-5} m around home position. In our experiments we only utilized the robot Y axis by locking the other axes with a tight PD controller.

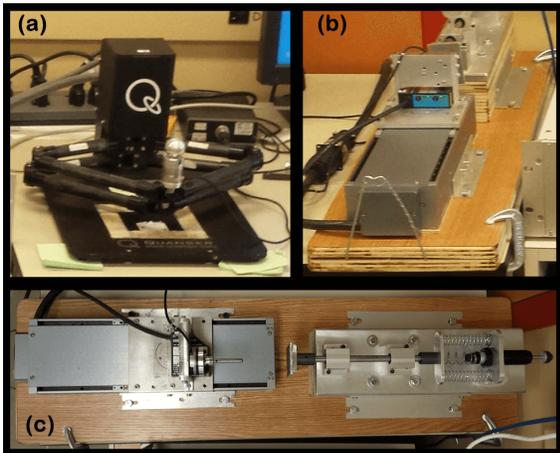


Fig. 6: Picture of the 1-DOF testbed used in the experiments.

The slave robot, as shown in the Figure 6-b, is made of a brushless linear motor (Primates PLG110) with a high resolution (1×10^{-6} m) linear and incremental encoder. The device is equipped with an 6-DOF ATI Mini-40 force/torque sensor to measure the contact force between the slave and environment. An Acuity AccuRange 200-50 laser proximity sensor is also installed in the carriage plate close to the robot end-effector to measure its distance from the environment. This rangefinder has a span of 50.9 mm with 42 mm standoff and offers an accuracy of $\pm 101 \mu\text{m}$ with normally distributed noise, measured during the experiments, of $\mathcal{N}(0, 0.015 \text{ mm})$. The data are obtained through serial communication (RS-232) and is configured to sample at the rate of 1 kHz. The laser output signal properties are also considered in the simulations. The experimental setup runs on Quanser QUARC[®] real-time control software running at rate of 1 kHz. Each robot is installed on a separate computer and the communication is established through a UDP protocol.

The numerical simulations were based on the 1-DOF experimental setup and were carried out on Simulink[®] using a fourth-order Runge-Kutta solver for a period of 10 seconds. The parameter values of the robots models and controllers used in both the simulations and experiments are shown in the Table I. The dynamic parameters used for virtual slaves

in both the simulations and experiments were chosen slightly different from the slave dynamic parameters to highlight possible modeling error.

TABLE I: The parameter values of robots model and controller.

Parameter	Master	Slave	Virtual Slaves
K_P	-	800	800
K_D	-	40	40
K_F	1	-	-
K_{PV}	-	4000	-
Mass (M)	0.27	1.4	1.57
Damping (B)	0.23	0.15	0.2

The 1-DOF environment module with a bank of linear springs is shown in Figure 6-c(right). The environment with specific setup springs is modeled by the stiffness $K_e = 2750 \text{ N m}^{-1}$ and damping $B_e = 36 \text{ N s m}^{-1}$, that were identified experimentally and used in all simulations. The impedance of saturation used in the virtual environments, as described in Section II-B, is set to $K = 1 \times 10^4 \text{ N m}^{-1}$ and $B = 1 \times 10^3 \text{ N s m}^{-1}$ since the master robot becomes unstable for higher local virtual slave impedance. As mentioned previously, the identification method used in this paper is the regular RLS method. In order to speed up the convergence, the initial covariance matrix is set to high value. The identification becomes active when the slave position X_s is greater than the environment resting position X_{e_0} . For the simulations, we used $X_{e_0} = 4 \text{ mm}$.

The task given to the operator was to make the slave get into contact with the environment and leave the environment after a few seconds, and then repeat the same pattern again. The operator was modeled by an LTI mass-damper-spring according to

$$F_h = F_h^* - Z_h X_h \quad (7)$$

where F_h^* represents the exogenous force generated by the arm muscles, and $Z_h := F_h/(X_h s) = M_h s + B_h + K_h/s$ denotes the arm impedance. To mimic a similar movement pattern, the exogenous force input $F_h^* = 7.5 \sin(1.2t)$ was used. The operator's hand dynamic parameters reported in [18] were used, that is $M_h = 0.5 \text{ kg}$, $B_h = 70 \text{ N s m}^{-1}$ and $K_h = 1000 \text{ N m}^{-1}$.

B. Simulation Results and Analysis

The position and force results of the simulation with a time delay of 500 ms are shown in Figure 7, where plots (a)-(b) are from the impedance reflecting; (c)-(d) from the proposed control architecture; (e)-(f) from modification mode 1, and (g)-(h) from the modification mode 2. In this figure, the local side position X_h and X_{vs} and force F_{ve} signals were shifted by the time delay to match the signals from the remote side for better visualization and easier evaluation of the transparency performance. The stiffness and damping identification results are displayed in Figure 8.

As one may notice, the operator trajectory (X_h) allows two contacts with the environment. Since before the first

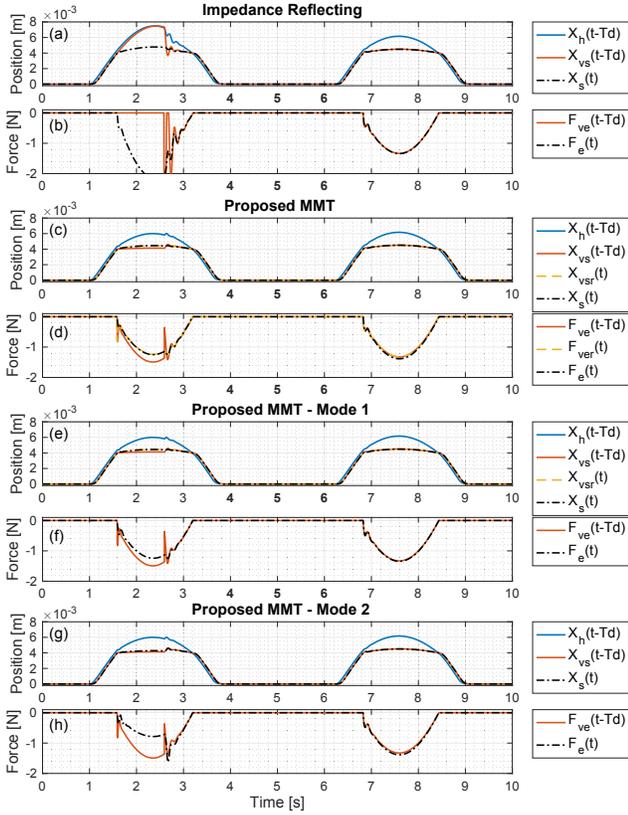


Fig. 7: Simulation results of position and force using (a)-(b) the impedance reflecting architecture; (c)-(d) the proposed model-reference model-mediated control architecture; (e)-(f) the proposed architecture mode 1; and (g)-(h) the proposed architecture mode 2.

contact the environment impedance is unknown at the local side, the first contact shows the architectures performance when the environment impedance is unknown or changing. As can be seen from Figure 8, the stiffness and damping converge quickly and the identified values stay close after the transition time. In all the plots in Figure 7, it is possible to see a change in all the signals around the time 2.5 s, that is when the environment impedance is received at the local side. The second contact shows the performance when the impedance of the environment is known and does not change or changes slightly. As can be observed from Figure 8, in this case the results are similar since the environment impedance has already been identified.

From the Figure 7-b, it is seen that using the impedance reflecting architecture, the operator does not have any feeling that the slave gets into contact with the environment until the change in environment impedance is received at local side. In fact, looking at the position plots from Figure 7-a, one may notice that the position of virtual slave X_{vs} follows the operator hand X_h closely after the environment resting position (4 mm). However after the impedance is received, the operator experiences severe oscillations as the virtual slave chatters on the virtual environment, resulting in the

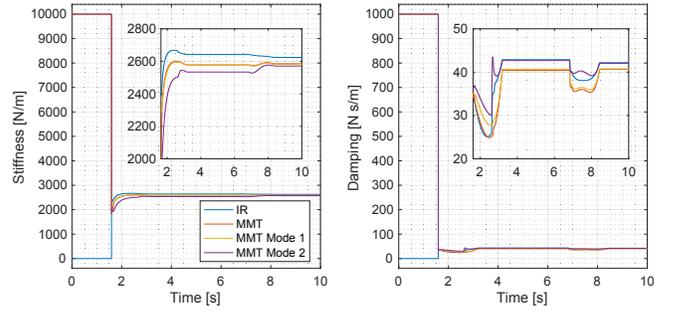


Fig. 8: Simulation identification results using the architectures shown in Figure 7.

virtual environment force F_{ve} oscillating substantially.

Such poor performance is not observed in the proposed model-reference control architecture, as can be seen in the Figure 7-c-d. Here, as the slave gets into contact with the environment, the operator feels the resistance as the virtual environment is preloaded with the saturation impedance. As a result, the local virtual slave position and force X_{vs} and F_{ve} to good extend follow the environment position X_e and force F_e . Since the remote virtual slave is updated locally the slave and the virtual slave closely follow each other. Due to the fast parameter convergence rate in comparison to the transportation delay, the modification mode 1 results in the same response as the original method, as shown in Figure 7-e-f.

Looking for the proposed architecture mode 2, one may say see that the position tracking is slightly better than the original proposed method. However, the force signals F_e and F_{ve} are farther apart. Once the change in the position of local virtual slave, which acts as a position command for the slave, reaches the slave side with a delay the slave force leaps to a much higher value to synchronize with the virtual environment force. This results in an undesired behavior that might damage the environment or instability.

Regarding the second contact with the environment, one may see that all results are similar since the environment impedance has already been identified, as can be observed from Figure 8.

C. Experimental Results and Analysis

Figure 9 displays the preliminary experimental results obtained from the proposed model-reference control architecture, in which the first and second contacts are shown. One may notice that the position and force results are similar with those obtained with the simulations (Figure 7-c-d), where the operator can feel the saturation environment at -16.3 mm until the actual environment impedance is received at local side around 12.75 s. At that moment, there is a force oscillation felt by the operator; however, this oscillation is significantly attenuated at the slave and remote virtual slave as shown in Figure 9-b. After the adjustment, the task is performed smoothly and the slave and virtual slave remain synchronized. Figure 10 shows the identification results, where one may see that the stiffness (K_e) converged rapidly

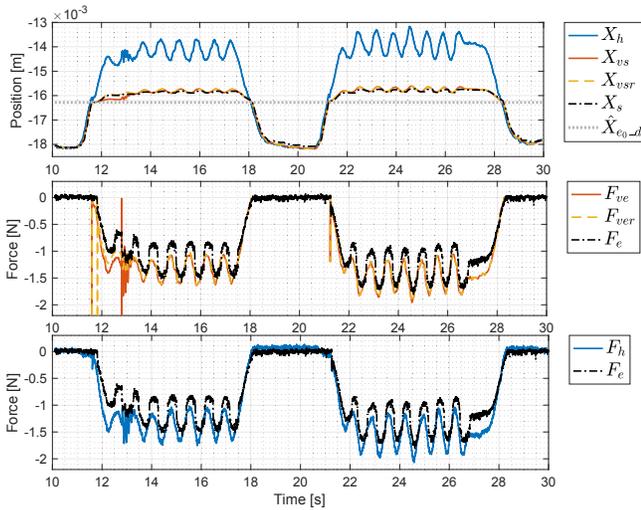


Fig. 9: Position and force results using the proposed model-reference model-mediated control architecture with time delay of 500 ms.

close to the values shown in Section III-A. The damping (B_e) did not converge to the expected value due to the low velocity during the contact. However, this did not influence the validation of the control architecture as the contact force is mainly defined by the stiffness component.

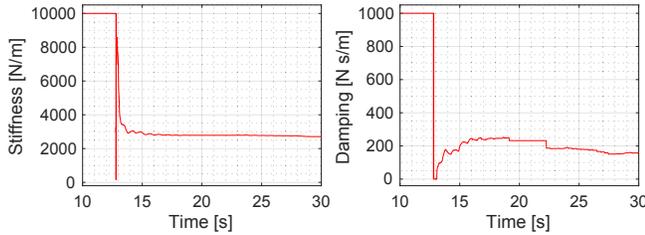


Fig. 10: The identification results during the experiment shown in Figure 9.

IV. CONCLUSIONS

In this paper, we proposed a model-reference model-mediated control architecture to deal with the issue of time delays in bilateral teleoperation systems. In this control architecture, virtual slave and environments are implemented at the master (local) and slave (remote) sides. The local virtual models are updated with delays; however, they keep the master responsive. The remote virtual models are updated locally on the fly with the environment identified dynamic parameters. The slave follows the position of the remote virtual model; therefore, the remote slave model acts as a model reference for the slave, improving the synchronization between the slave and local virtual slave. The proposed control architecture demonstrated smoother response with significantly less oscillations compare to a benchmark impedance reflective control architecture. Future work will include extensive experimental assessment of the

proposed architecture with various amount of time delays and environment impedances.

ACKNOWLEDGMENT

The first author wishes to thank Amanda Pecly for the support. The authors also thank INPE for supporting the ETE Course/CMC Option and the Emerging Leaders in the Americas Program (ELAP) funded by the Department of Foreign Affairs, Trade and Development (DFATD) of Canada.

REFERENCES

- [1] D. A. Lawrence, "Stability and transparency in bilateral teleoperation," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 5, pp. 624–637, 1993.
- [2] Y. Yokokohji and T. Yoshikawa, "Bilateral control of master-slave manipulators for ideal kinesthetic coupling-formulation and experiment," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 605–620, 1994.
- [3] K. Hashtrudi-Zaad and S. E. Salcudean, "Analysis of control architectures for teleoperation systems with impedance/admittance master and slave manipulators," *The International Journal of Robotics Research*, vol. 20, no. 6, pp. 419–445, 2001.
- [4] C. Tzafestas, S. Velanas, and G. Fakiridis, "Adaptive impedance control in haptic teleoperation to improve transparency under time-delay," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2008, pp. 212–219.
- [5] R. J. Anderson and M. W. Spong, "Asymptotic stability for force reflecting teleoperators with time delay," *The International Journal of Robotics Research*, vol. 11, no. 2, pp. 135–149, 1992.
- [6] D. Lee and M. W. Spong, "Passive bilateral teleoperation with constant time delay," *IEEE Transactions on robotics*, vol. 22, no. 2, pp. 269–281, 2006.
- [7] B. Hannaford and J.-H. Ryu, "Time-domain passivity control of haptic interfaces," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 1–10, 2002.
- [8] Y. Ye, Y.-J. Pan, Y. Gupta, and J. Ware, "A power-based time domain passivity control for haptic interfaces," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 4, pp. 874–883, 2011.
- [9] B. Hannaford, "A design framework for teleoperators with kinesthetic feedback," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 4, pp. 426–434, 1989.
- [10] W.-K. Yoon, T. Goshozono, H. Kawabe, M. Kinami, Y. Tsumaki, M. Uchiyama, M. Oda, and T. Doi, "Model-based space robot teleoperation of ets-vii manipulator," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 602–612, 2004.
- [11] F. Mobasser and K. Hashtrudi-Zaad, "Predictive teleoperation using laser rangefinder," in *Proceedings of IEEE Canadian Conference on Electrical and Computer Engineering*, 2006, pp. 1279–1282.
- [12] A. C. Smith and K. Hashtrudi-Zaad, "Smith predictor type control architectures for time delayed teleoperation," *The International Journal of Robotics Research*, vol. 25, no. 8, pp. 797–818, 2006.
- [13] P. Mitra and G. Niemeyer, "Model-mediated telemanipulation," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 253–262, 2008.
- [14] F. Mobasser and K. Hashtrudi-Zaad, "Stable impedance reflecting teleoperation with online collision prediction," in *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, 2007, pp. 476–482.
- [15] G. C. Goodwin and K. S. Sin, *Adaptive filtering prediction and control*. Prentice Hall, 1984.
- [16] D.-J. Park and B.-E. Jun, "Selfperturbing recursive least squares algorithm with fast tracking capability," *Electronics Letters*, vol. 28, no. 6, pp. 558–559, 1992.
- [17] A. Haddadi and K. Hashtrudi-Zaad, "Online contact impedance identification for robotic systems," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 974–980.
- [18] H. Kazerooni, T.-I. Tsay, and K. Hollerbach, "A controller design framework for telerobotic systems," *IEEE Transactions on Control Systems Technology*, vol. 1, no. 1, pp. 50–62, 1993.