Robotica (2017) volume 35, pp. 1121–1136. © Cambridge University Press 2015
doi:10.1017/S0263574715001010

Extending model-mediation method to multi-degree-of-freedom teleoperation systems experiencing time delays in communication

Emre Uzunoğlu and Mehmet İsmet Can Dede*

Department of Mechanical Engineering, İzmir Institute of Technology, İzmir, Turkey. E-mail: candede@iyte.edu.tr

(Accepted November 6, 2015. First published online: December 10, 2015)

SUMMARY

In this study, a bilateral teleoperation control algorithm is developed in which the model-mediation method is integrated with an impedance controller. The model-mediation method is also extended to three-degrees-of-freedom teleoperation. The aim of this controller is to compensate for instability issues and excessive forcing applied to the slave environment stemming from time delays in communication. The proposed control method is experimentally tested with two haptic desktop devices. Test results indicate that stability and passivity of the bilateral teleoperation system is preserved under variable time delays in communication. It is also observed that safer interactions of the slave system with its environment can be achieved by utilizing an extended version of the model-mediation method with an impedance controller.

KEYWORDS: Bilateral teleoperation; communication failure; model-mediation method; impedance control.

1. Introduction

Generally, teleoperation involves a human operator controlling a slave system through the use of a master system to perform tasks that are too complicated to be handled autonomously. These are often conducted remotely and over long distances. Examples of typical teleoperation systems include those used to conduct operations occurring in space, underwater, or where hazardous materials, such as nuclear materials, are handled.[1]

In force-reflecting bilateral teleoperation systems, the operator receives haptic feedback including kinaesthetic and/or tactile from the master device along with visual and/or auditory feedback. The goal is to facilitate the task execution in the remote environment by enhancing the feeling of being present in the remote environment, commonly referred to as tele-presence. As a result of this, the human operator can perform complex manipulations in a remote environment more conveniently and precisely. The transmitted signals between the two systems can either be motion, forcing or both as seen in the bilateral teleoperation. In the two-channel teleoperation systems, the human operator uses the master system to send the motion demand to the slave and receives the slave system's interaction forces via the master system. In the four-channel teleoperation, both the force and motion signals are sent and received between the two systems.[2]

The design of the teleoperation controllers involves a trade-off between the conflicting requirements of stability versus performance. The time delay in the communication line produces instability in a force-reflecting teleoperation system, which has been one of the main challenges for the past three decades. During this time, various methods including passivity-based approaches,[3,4] wave variable techniques,[5,6] and modifications of these methods[7] have been proposed to compensate for the instability issues due to the variable time delays in the communication line. However, these

* Corresponding author. E-mail: candede@iyte.edu.tr.

methods require a prediction of the amount of delay or the delay trend to be included in their algorithms.

A relatively new method known as the model-mediation method was proposed by Mitra and Niemeyer[8] to handle the instability issues caused by the constant time delays in communication line. The model-mediation method was first introduced for single DoF teleoperation systems.[8] In the model-mediation method, the master system receives the slave system environment's model updates in terms of the contact surface information instead of transmitting the motion and forcing data between the two systems. Thus, the slave system receives both motion and force data information from the master system through a proxy. In recent studies, the model-mediation method has been employed on a two-DoF teleoperation system where vision sensors were used to estimate geometric properties of the environment before actually touching the environment.[9] The conventional model mediation method without vision sensors is explained in more detail in the next sections.

Although the above mentioned controllers are working to improve the stability of the teleoperation system experiencing time delays in the communication line, there are still stability issues due to the rigid and unpredictable interactions of the slave system with its environment. When the slave system is directed by the master system and interacts with its environment, the master system only becomes aware of this interaction after a time delay. During these delays, the slave system should not damage itself or its environment by showing compliance to its environment. In order to cope with this problem, a teleoperation system experiencing time delays should include a compliance control algorithm for the slave system's controller.

The parallel position/force controllers provide a way to control the motion of the manipulator in parallel with the forcing that is exerted to its environment. The slave manipulator in the bilateral teleoperation is required to interact with its environment by exerting force to its environment. The necessary compliance can be achieved passively by the hardware (mechanically) or actively by the hardware and the control software. The control software in these types of applications is usually selected from the common parallel position/force controllers namely, the impedance control, the admittance control, the hybrid position/force control, and their modified versions.[10,11] In this study, the impedance controller initially developed by Hogan[12] was modified and used as the slave manipulator's controller. Therefore, the objective of our research was to achieve a safer and more stable interaction of the slave system with its environment by using this modified impedance controller.

The proposed control strategy comprises the model-mediation method in three-dimensional (3D) space and the impedance controller to be used as the slave system's controller. The core of the novelty of this work is the formulation of the safe and stable 3D teleoperation under unknown and possibly variable communication delays. The user is not subjected to excessive and instantaneous forces that may result from time-delayed teleoperation of the slave that is interacting with its environment since the user is not directly teleoperating the slave but the information flow is regulated in a mediated model. On the other side, the slave carries out the task with delayed demands from the mediated model and updates the model with a communication delay. While performing the task under communication delays, the slave device may cause damage to its environment and/or itself. The adaptation of the modified impedance controller as the slave system controller is another contribution for safer interactions of slave device independent of model mediation technique, especially during first contact with a new surface.

The validation of the proposed control system is investigated by evaluating the stability and performance of a bilateral teleoperation system under variable and constant time delays in the communication line. The experimental set-up is a bilateral teleoperation system consisting of the master system as the Novint's Falcon haptic device and slave device as the Geomagic's (previously Sensable Technologies) Phantom Desktop haptic device. A tip-point contact-force estimator is used to acquire interaction-force information from the slave system. In the experiments, the proposed controller is evaluated through various working scenarios. The experimental tests reveal that using the proposed controller results in a more stable and safer teleoperation system regardless of whether the time delay type in the communication is constant or variable.

In Section 2, an overview of proposed controller is described. In Section 3, the experimental set-up is presented, along with the examination of the test results. Last, observations and comments based on the experiment results are discussed in the conclusion.

https://doi.org/10.1017/S0263574715001010

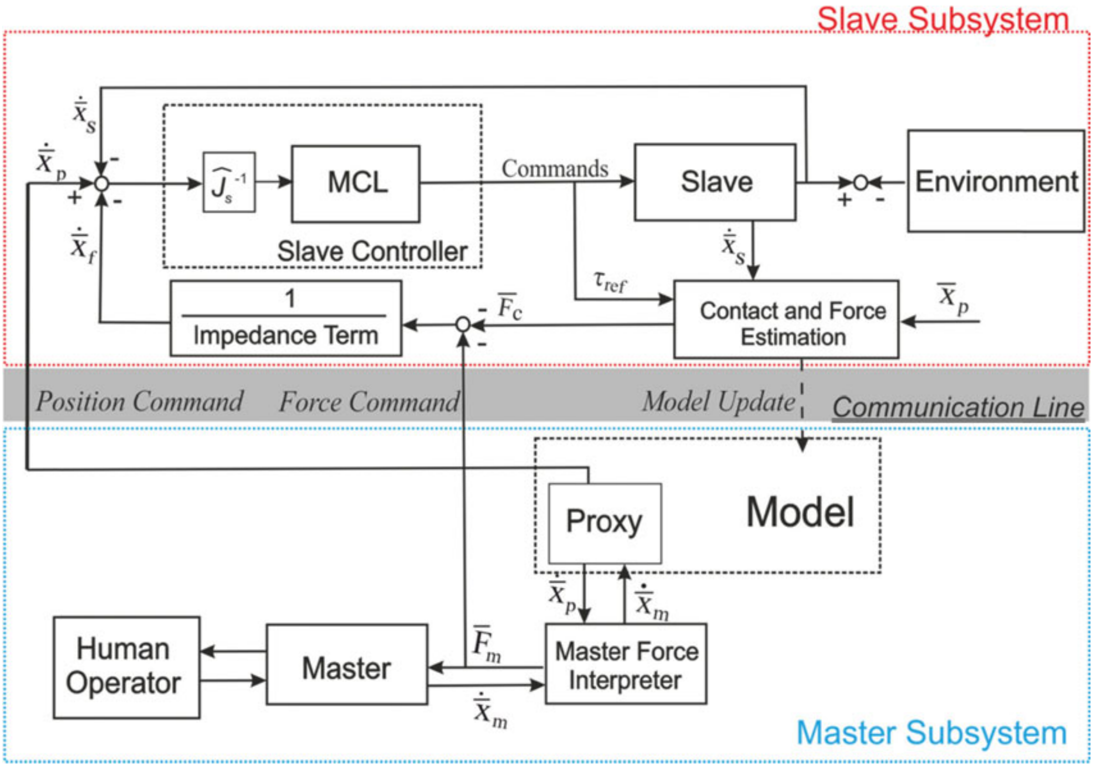

Fig. 1. The proposed teleoperation control scheme.

2. Overview of the Proposed Controller

The slave system of the teleoperation system developed in this study is comprised of the slave device (Phantom Desktop), slave controller, and contact estimation algorithms. Conversely, the master system includes the master haptic device, haptic device controller, and a mediated slave representation, which hereinafter is referred to as proxy. The proxy moves in the haptic environment model that represents the slave's environment.

In this teleoperation controller setting, the human operator sends the commands to the slave system and receives the interaction forces acquired in the slave system using the model and the proxy created in the master system. This information exchange between the operator and the model and proxy is achieved without a communication time delay. The operator using the master haptic device interacts with the virtual environment (the model) through the proxy, which has its own dynamics and assumptions for ensuring the stability of bilateral teleoperation systems as explained in the work of Mitra and Niemeyer.[8] The master and slave systems, their components, and the information flow between these components are illustrated in Fig. 1. The master system's components and slave system's components are grouped together among themselves and these groups are divided by a communication line highlighted in grey colour.

For an orthogonal task space, a coordinate system defined by the unit vectors with respect to the right-hand rule as $\vec{x}^1$, $\vec{x}^2$, and $\vec{x}^3$; the parameters appearing in Fig. 1 are described below:

$\dot{x}_p = \begin{bmatrix} \dot{x}_p^1 & \dot{x}_p^2 & \dot{x}_p^3 \end{bmatrix}^T$: The proxy velocity/motion;

$\dot{x}_s = \begin{bmatrix} \dot{x}_s^1 & \dot{x}_s^2 & \dot{x}_s^3 \end{bmatrix}^T$: The slave system velocity/motion;

$\dot{x}_m = \begin{bmatrix} \dot{x}_m^1 & \dot{x}_m^2 & \dot{x}_m^3 \end{bmatrix}^T$: The master system velocity/motion command;

$\dot{x}_f = \begin{bmatrix} \dot{x}_f^1 & \dot{x}_f^2 & \dot{x}_f^3 \end{bmatrix}^T$: The motion modification term resulting from the impedance term;

$\bar{F}_c = \begin{bmatrix} F_c^1 & F_c^2 & F_c^3 \end{bmatrix}^T$: The contact force along the normal of the contact surface;

https://doi.org/10.1017/S0263574715001010

$\bar{F}_m = \left[F_m^1 \quad F_m^2 \quad F_m^3 \right]^T$: The forces exerted to the operator through the master device;

$\bar{\tau}_{\text{ref}} = \left[\tau_{\text{ref1}} \quad \tau_{\text{ref2}} \quad \tau_{\text{ref2}} \right]^T$: The reference (command) torque for the joint actuators.

In the model-mediation method, the motion and forcing commands from the master system are delivered to the slave system through the communication line. However, only remote environment information necessary to create a model within the master system is delivered from slave system to the master system. An example of information sent from the slave system may be the location information of the experienced contact on the slave environment. Additional information about the slave environment can be delivered to create a more comprehensive slave model by integrating more sensors, such as vision sensors. However, this increase in information would require a surge in communication bandwidth.

The slave robot interacts with its environment based on the commands generated in the slave controller using the motion demand received from the master system, $\dot{\bar{x}}_p$, slave robot's motion feedback, $\dot{\bar{x}}_s$, and the motion modification, $\dot{\bar{x}}_f$, issued by the impedance term to regulate interaction forces during unexpected contacts.

The forces generated in the master force interpreter, $\bar{F}_m$, and motion commands from proxy, $\dot{\bar{x}}_p$, are both calculated in task space. As a result of this, the exchange of motion/force information between the two kinematically different robots becomes more convenient.

If no force sensor is used for the slave robot, a force and contact estimation algorithm should be devised using torque references/commands, $\bar{\tau}_{\text{ref}}$, motion demand, $\dot{\bar{x}}_p$, and motion feedback, $\dot{\bar{x}}_s$. Employing a parallel position/force controller allows a safe interaction of the slave with its environment during the unexpected contact situations. The operator's force demand is also transmitted to the slave system by modifying force measurements fed into the parallel position/force controller. In this study, the parallel position/force controller is composed of an impedance term to transform the forces measured to motion. Force demands received from the master system modify the measured forces, $\bar{F}_c - \bar{F}_m$, and through impedance term, these are transformed to the motion modification term, $\dot{\bar{x}}_f$, to be fed into the inner motion controller loop that is identified as slave controller in Fig. 1. Hence, the controller can also be considered a version of the impedance controller.

2.1. The master force interpreter

The proxy is a virtual representation of a slave device in the master system. The proxy follows the master system's motion with its own dynamics in the constraints of the modelled environment constructed in the master system.[13] The model of the remote environment in the master system is updated in accordance with the assumptions introduced below to ensure the stability of the teleoperation system while preventing application of the excessive forces to the human operator. The contact surface data is acquired along $\vec{x}^1$, $\vec{x}^2$, and $\vec{x}^3$ directions from the slave system and then sent through the communication line to the master system as an estimation of the surface location.

The forces exerted to the user, $\bar{F}_m$, is generated in the master system with the PD gains, as shown in Eq. (1), where the error is calculated by comparing the proxy and the master motions.

$$\bar{F}_m = \hat{K}_{pm}(\bar{x}_p - \bar{x}_m) + \hat{K}_{dm}(\dot{\bar{x}}_p - \dot{\bar{x}}_m). \tag{1}$$

In Eq. (1);

$\hat{K}_{pm} = diag(K_{pm1}, \quad K_{pm2}, \quad K_{pm3})$: The proportional gain constant matrix;

$\hat{K}_{dm} = diag(K_{dm1}, \quad K_{dm2}, \quad K_{dm3})$: The derivative gain constant matrix.

In order to exert forces to the user and ensure the stability of the teleoperation system during the communication failures, two assumptions are made for the interactions of the proxy with the model constructed within the master system.

Assumption 1: The proxy created as a representation of the slave system's tip point is massless and follows the master system's tip point through its own built-in dynamics.[8]

Assumption 2: It is designated that once a constraint is created in the model, the proxy motion is restricted by the surface constraint as proposed in the work of Mitra and Niemeyer.[13] Therefore, the surface is never penetrated by the proxy.

https://doi.org/10.1017/S0263574715001010

Dynamic behaviour of the proxy, mentioned in the Assumption 1, is achieved by calculating a dynamic velocity $\dot{\bar{x}}_p$ for the proxy as represented in Eq. (2).

$$\dot{\bar{x}}_p = \dot{\bar{x}}_m + \hat{K}_p(\bar{x}_m - \bar{x}_p). \tag{2}$$

In this equation, $\hat{K}_p$ is a 3×3 constant positive diagonal matrix as represented in Eq. (3). Its components are calculated with respect to the parameters used in Eq. (1).

$$\hat{K}_p = diag(\, K_{dm1}/K_{pm1}, \quad K_{dm2}/K_{pm2}, \quad K_{dm3}/K_{pm3}\,). \tag{3}$$

When no constraints are created on the master system, the proxy directly follows the master system demands and the condition described in Eq. (4) is satisfied.

$$\bar{x}_m - \bar{x}_p = \bar{0}. \tag{4}$$

When a contact surface information is sent to the master system, the position of the proxy position, $\bar{x}_p$, is restricted by Assumption 2. Therefore, the master system's tip-point position is no longer equal to the proxy position. If an update from the slave system as a result of a new surface constraint is detected, the existing surface constraint will be shifted or removed in the model. After the proxy reaches the master system's tip point position, assuming it is a no contact situation, it tracks the master system's motion perfectly and instantly responds to the master system's commands with the condition given in Eq. (4).

When the contact occurs, the proxy remains on the virtual surface obeying Assumption 2. As a result of these assumptions, no energy is stored in the system since the proxy is massless and the virtual wall is never penetrated by the proxy.

The dynamic behaviour of the proxy and the model construction in the master system ensures that no excessive forces are transmitted to the master system during time delays and during the changes in the slave system's environment. In this algorithm, the slave system's environment model updates are designated to be a shift, detection, or removal of the surface within the matched workspace.

2.2. *Slave controller*

An impedance controller that uses a tip-point contact-force estimator is implemented as the slave system controller. The purpose behind using a force estimator is that the slave system is a lightweight device that cannot carry a 3D force sensor as a payload. The main reason to employ an impedance controller on the slave system is to prevent damage on the slave system during the initial contact with the environment and to make the slave system comply with its environment. The impedance controller is modified so that the force demands issued from the master system are used to change the force estimator's output in such a way that the slave system can exert the demanded amount of forces to its environment.

The slave system is controlled by a motion control law (MCL) implemented within the impedance controller scheme presented in Fig. 1. The contact force information is calculated in the contact- and force-estimation block. The model updates information to be transmitted to the master system and from the contact- and force-estimation block. The motion errors fed into the MCL are calculated in the task space by comparing the transmitted proxy motion with the measured slave motion and by modifying the proxy motion with motion modification issued by impedance term. Following, the motion error calculated in task space is transferred to the joint space by the inverse of the Jacobian matrix. Due to the constructional restrictions in both devices, the kinematic singularities that may result from taking the inverse of the Jacobian matrix are avoided in hardware. However, as in any robotic application, kinematic singularities that are not accounted for will be a restriction for the application of the proposed method.

The slave system's controller, MCL, is composed of a PID-type control structure with a gravity compensation term. Other non-linear effects are neglected due to relatively low inertial characteristics of the manipulator links and relatively lower operational speeds. This type of a local controller would result in larger tracking errors in faster operations. However, the proposed method enables the teleoperation system designer to use any other motion controller algorithm including non-linear controllers.

The impedance term is selected to be a second-order system representation that enables to insert a virtual mass-damper-spring element between the tip point of the slave system and its environment. The impedance term is expressed in Laplace domain as represented in Eq. (5) with M_d^i, B_d^i, and K_d^i coefficients, which are mass, damping, and spring parameters of the desired impedance along the $\vec{x}^i$ direction.

$$Z^i(s) = \left(M_d^i s + B_d^i + K_d^i/s\right)/\sigma^i, \quad \text{for } i = 1, 2, 3. \tag{5}$$

The impedance controller implemented on the slave system provides compliance to the environment. Therefore, in the case of unexpected interactions with the environment, it renders a virtual mass-spring-damper system in between the tip point of the slave system and its environment to regulate the contact forces and provide contact stability.[14] This is required in teleoperation scenarios with time delays in the communication line and contact information is sent to the master system with a delay. It has been discussed in[15] that large feedback gains (smaller K_d and B_d) with time delays in the measurement system can result in unstable behaviour. Therefore, in selecting the feedback gains, the speed of force estimation is crucial in our controller.

The parameter K_d corresponds to the stiffness of the target impedance. High stiffness is desired for the case when the accuracy is very important and environment is compliant. Smaller value of K_d could be used when in contact with stiff environments or when smaller interaction forces should be maintained. The B_d parameter is the damping of the target impedance. A large damping coefficient means that the system should dissipate much of the energy.[15] M_d coefficient describes the mass of the target impedance and therefore it is a good tuning parameter to describe the transient behaviour during contact.[3,14] Obviously, the tuning process is highly related with the type of contact between the end-effector and the surface. A good estimation of the surface type is required for better performance during contact situations. This is required in teleoperation scenarios with time delays in the communication line and contact information is sent to the master system with a delay. In this work, the impedance controller parameters are selected by trial and error process for better contact performance in terms of contact stability and tracking the force demands sent from the master subsystem.

During the initial contact with the environment, only estimated contact forces, $\bar{F}_c$, are active in the impedance controller. However, as the model is updated with a delay and the master system's force and motion demands are calculated, these are sent to the slave system with another time delay. The received master force demand is the negative sign of the master force applied to the user, $\bar{F}_m$. For this reason, the estimated force is updated to exert the required $-\bar{F}_m$ amount of force to the slave system's environment. The necessary calculation to determine motion modification due to the impedance term is shown in Eq. (6).

$$-F_m^i - F_c^i = Z^i(s) \cdot \dot{x}_f^i(s), \quad \text{for } i = 1, 2, 3. \tag{6}$$

When there is no contact with the slave environment, only pure motion control must be active as the slave system's controller. This is accomplished by inserting a σ^i term as shown in Eq. (5). The σ^i term is 1 when the contact is detected and 0 when no contact is detected along the $\vec{x}^i$ direction. Hence, the motion modification term, $\vec{x}_f^i$, is cancelled out in the slave system's control system when there is no estimated contact with the slave environment.

Assumption 3: Only the objects that have surface normal along the $\vec{x}^1$, $\vec{x}^2$, and $\vec{x}^3$ axes are considered to be present in the slave system's environment.

Collision of the tip point of the slave system with its environment is detected when the tip point velocity, $\dot{x}_s^i$, along the contact direction, $\vec{n} = \vec{x}^i$, is approximately equal to zero and the calculated tip-point contact force is bigger than a predetermined threshold value. When the two conditions are satisfied, a contact state is declared and the force tracking is enabled. The contact detection conditions are formulated in Eqs. (7) and (8).

$$\dot{\bar{x}}_s^T \cdot \vec{n} \cong 0; \tag{7}$$

$$\left|\bar{F}_c^T \cdot \vec{n}\right| \geq \bar{F}_{\text{Threshold}}^T \cdot \vec{n}. \tag{8}$$

https://doi.org/10.1017/S0263574715001010

Prior to the experimentations, in order to calculate the threshold value, $\bar{F}_{\text{Threshold}}$, the slave manipulator is run by the human operator (through issuing commands from the master system) in an ideal teleoperation scenario with no time delays throughout its entire workspace. Applied torque values to the slave manipulator joint actuators, $\bar{\tau}_{sfm}$, are acquired throughout this process and translated to task space by Eq. (9).

$$\bar{F}_{\text{Threshold}} = \left(\hat{J}_s^T\right)^{-1} \cdot \bar{\tau}_{sfm} + \bar{\delta}. \tag{9}$$

A safety factor, $\bar{\delta}$, is added on top of the calculated task space threshold forces to provide a safety zone when estimating the contact with the slave environment as shown in Eq. (9). The joint actuator torques, $\bar{\tau}_{sfm}$, appearing in Eq. (9) is composed of the tangential inertial torques,$\bar{\tau}_{\text{inert}}$, the Coriolis and centripetal force effects, $\bar{\tau}_{cc}$, the gravitational force effects, $\bar{\tau}_g$, and the frictional force effects, $\bar{\tau}_f$. In the free motion case, there are no external forces acting on the system hence, the effect of external forces on the joints, $\bar{\tau}_{\text{ext}}$, is cancelled out from the formulation of $\bar{\tau}_{sfm}$, which is described in Eq. (10).

$$\bar{\tau}_{sfm} = \bar{\tau}_{\text{ext}} + \bar{\tau}_{\text{inert}} + \bar{\tau}_{cc} + \bar{\tau}_g + \bar{\tau}_f. \tag{10}$$

In order to estimate the contact condition and the interaction forces, a force estimator is used. The estimation algorithm that appears in ref. [8] is improved for increased precision. The new estimator is based on the work of Ohishi *et al.*[16] in order to take into account the dynamic and gravitational effects. Within the force estimation algorithm, when the velocity along one of the $\vec{x}^1$, $\vec{x}^2$, and $\vec{x}^3$ axes approaches to zero, applied joint torques are translated to the forces in task space and compared with the threshold force values. It is assumed that if it is a contact condition, the motion in terms of accelerations and velocities is equal to zero and thus, the inertial, Coriolis, centripetal, and frictional force effects are cancelled out in Eq. (10). In this condition, applied torques to the joints of the slave system, $\bar{\tau}_{\text{ref}}$, only consists of gravitational forces and external force effects. Therefore, cancelling out the gravitational forces, the external forces acting on the system is calculated as shown in Eq. (11).

$$\bar{F}_c = \left(\hat{J}_s^T\right)^{-1} \cdot (\bar{\tau}_{\text{ref}} - \bar{\tau}_g). \tag{11}$$

In order to estimate the slave environment constraints, such as contact surfaces, a state observer and contact force estimation interpret the position feedback from the slave motion sensors when the reference torque values exceed the required amount of torques for the task space motions along $\vec{x}^1$, $\vec{x}^2$, and $\vec{x}^3$ axes.

Only the contact position information from the contact and force estimation block is transmitted as model updates to construct a model in the master system. The model update and the tracking methodology are applied consistent with the proposed method of Mitra and Niemeyer in.[8] Additionally, this method is extended to three-DoF by introducing the volumetric modelling of flat surfaces in the directions of $\vec{x}^1$, $\vec{x}^2$, and $\vec{x}^3$ axes. The outcome of contact detection is used in the construction of volumetric model of the slave environment.

2.3. *The model construction*

The model update flow is represented in Fig. 2. In the model, the proxy interacts within the constraints of the virtual surfaces as commanded through the master system while the slave system interacts with the object surfaces in its own environment.

In this work, the slave workspace is divided into cubes with a side length 10 mm. The length of the cube sides is selected with respect to the relatively larger contact surface of the slave manipulator's tip point used in this study.

Therefore, a rigid body with variable surface conditions is discretized to a rigid body composed of 10 mm side length cubes. When a contact is detected in a divided region with respect to the Assumption 3, the surface constraint in that region is sent to the master system as a model update. If no contact is detected in a region, the surface constraints are constructed outside of the slave system's workspace where the slave system cannot have interactions.

All the updated surface constraints for the divided regions are stored within the master system and as a result, an updated volumetric model of the slave system's environment is constructed within the

https://doi.org/10.1017/S0263574715001010

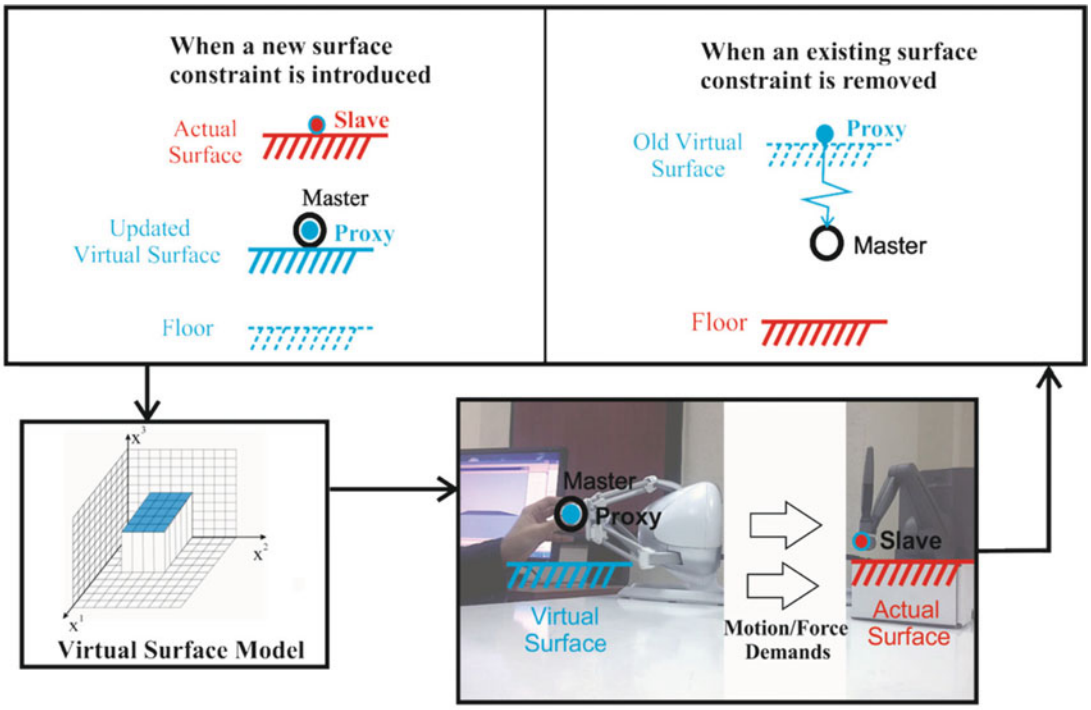

Fig. 2. The model update flow.

master system. For example, in the slave system's environment model presented in Fig. 2, the location of a surface, which is highlighted in blue with a surface normal along $\vec{x}^3$ direction, is updated. The updated information is obtained by contact estimation which was proposed in Assumption 3. For the implementation of model updates, Assumption 4 is applied to avoid the unexpected reaction forces that may be generated during the model updates.

Assumption 4: As the surface contact information and the direction of the contact forces is sent from the slave system, the virtual surface in the model is updated in such a way that the virtual surface is never above the proxy along the direction of contact force.

If a contact is detected above the current position of the proxy (with respect to the contact force direction), the virtual surface is re-constructed just below the proxy's position and kept below the proxy until it reaches the slave system's contact position as described in Fig. 2. When a previously updated virtual surface constraint is removed, the proxy gradually approaches the master system's position with the proxy dynamics.

2.4. Further discussion on the stability of the proposed teleoperation system

Bilateral teleoperation systems are closed-loop systems and system architecture is important for the overall stability of the system. There have been numerous studies about the stability of bilateral teleoperation system using absolute stability conditions.[17,18] However, model-mediated systems are non-linear and don't allow straightforward stability analysis. It has been discussed in ref. [19] that the model-mediated systems can be investigated based on classic stability by gain stabilization. In this case, gain is the stiffness of the slave system's environment and when the stiffness of the environment gets a larger value, gain used in this analysis becomes larger respectively. On the other hand, phase lag is considered as a result of communication delay and slave controller lag. Therefore, if the contact is a relatively stiffer one and there are considerable amount of delays either in communication or slave controller response, then the system can become unstable. In this case, the force feedback gain can be lowered to result in a gain margin higher than one which will make the system stable.

It has been discussed that rather than the gain stabilization approach, stability of model-mediated systems can be considered as phase stabilization.[19] Phase lead required by this approach is achieved through phase lead that is exploited by the predictive behaviour of the model-mediation method. In

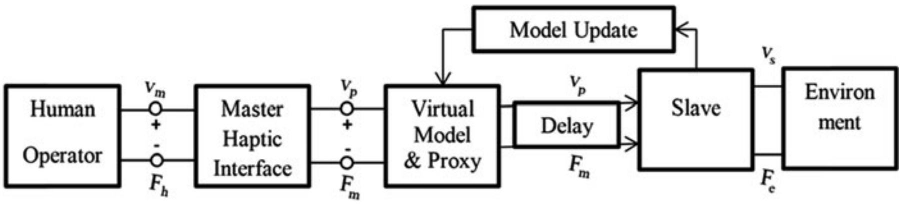

Fig. 3. Information flow in model-mediated teleoperation system.

model-mediation method, the estimated surface information is used to predict and display environment force. How accurate and how fast this prediction happens directly affects the system stability. When the predicted motion and/or forces differ from the observed ones, they correlate to the higher loop gains which can also make the system unstable. However, if the model estimation works faster, then the system phase lag lowers and the system stability improves.

Another approach to model-mediated teleoperation system stability can be done by investigating the system as a combination of two teleoperation systems. In Fig. 3, the signal flow is presented and it indicates that there is a bilateral information flow between the human operator and the virtual model and proxy.

This part of the teleoperation system can be seen as a two-port system as human operator being one port and the virtual model and proxy being the other port. The two port system can be described in Eq. (12) in terms of a hybrid matrix.[17,18]

$$\begin{bmatrix} F_h \\ -v_p \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} v_m \\ F_m \end{bmatrix}. \tag{12}$$

Hybrid Matrix's h parameters are defined as

$$h_{11} := \left. \frac{F_h(s)}{\dot{x}_m(s)} \right|_{F_m=0} = M\ddot{x}_m(t) + B\dot{x}_m(t) + Kx_m(t) \begin{bmatrix} v_m \\ F_m \end{bmatrix} \tag{13}$$

$$h_{12} := \left. \frac{F_h(s)}{F_m(s)} \right|_{v_m:=0} = 1 \tag{14}$$

$$h_{21} := \left. \frac{-\dot{x}_p(s)}{\dot{x}_m(s)} \right|_{F_m:=0} = -1 \tag{15}$$

$$h_{22} := \left. \frac{-\dot{x}_p(s)}{F_m(s)} \right|_{v_m:=0} = 0. \tag{16}$$

F_h and F_m are forces applied (efforts) by human and master device and $\dot{x}_m$ and $\dot{x}_p$ represents the motions (flows) of master and proxy, respectively. Master device is governed by the equations of motion of a rigid manipulator with M, B, and K which are mass, damping, and stiffness respectively. As introduced in model mediation method, the virtual surfaces created in the model are infinitely stiff and the proxy itself is massless. Proxy follows the master motion when no constraints are introduced in the model when Eq. (4) is satisfied.

For a two-port system stability, absolute stability criterion can be applied as presented in ref. [18] Sufficient conditions are provided with Llewellyn's stability criteria[20] for unconditional stability of master haptic interface. According to Llewellyn's stability criteria, linear time invariant (LTI) two-port network system is absolutely stable if,

(1) The hybrid parameters h_{11} and h_{22} have no poles in the right half plane.
(2) Any poles of h_{11} and h_{22} on the imaginary axis are simple with real and positive residues.
(3) For all real values of w,

$$\begin{gathered} Re\,[h_{11}] \geq 0, \qquad Re\,[h_{22}] \geq 0; \\ 2Re\,[h_{11}]\,Re\,[h_{22}] - Re\,[h_{12}]\,Re\,[h_{21}] - |h_{12}h_{21}| \geq 0. \end{gathered} \tag{17}$$

https://doi.org/10.1017/S0263574715001010

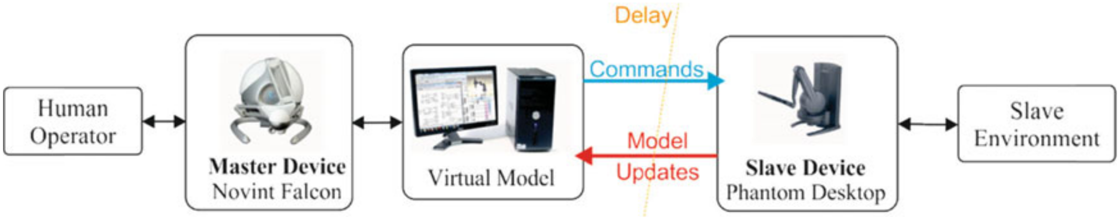

Fig. 4. Experimental set-up.

When necessary conditions for hybrid matrix parameters are provided the master system will be stable with passive human operator and virtual model.

$$\begin{bmatrix} F_h \\ -v_p \end{bmatrix} = \begin{bmatrix} Ms + B + K/s & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} v_m \\ F_m \end{bmatrix}. \tag{18}$$

By adding h parameters into hybrid matrix with Eq. (12), the resultant two-port system is expressed in Eq. (19). The haptic interface presented with the two-port continuous-time is reciprocal and satisfies absolute stability.[18] One can also discuss the effect of discretization in the virtual model on the system stability as in ref. [17] Higher sampling frequency range obviously helps to maintain stability in this scenario.

The second part of the teleoperation system represented in Fig. 3 is the unilateral teleoperation between the proxy and the slave subsystem. The force and the motion commands are sent to the slave subsystem in an open-loop architecture. The only feedback signal from the slave system is the location of the surface which is not in terms of motion or forces. Therefore, this teleoperation system cannot be considered as a closed-loop bilateral teleoperation but an open-loop unilateral teleoperation. Hence, stability issues for the second teleoperation subsystem vanish and instability issues can only arise from the slave subsystem's local controller. The parameter selection for the slave subsystem's local controller is discussed in the previous section.

3. The Experimental Set-Up

Two haptic desktop devices are used as the master and the slave manipulators in the experiments. Novint Falcon haptic device is used as the master manipulator, which is a three-DoF parallel-mechanism haptic device. It can track the hand motion and reflect forces along $\vec{x}^1$, $\vec{x}^2$, and $\vec{x}^3$ axes (Novint Technologies, Inc.; www.home.novint.com). Phantom Desktop from Geomagic, http://www.geomagic.com, (formerly Sensable Technologies) is used as the slave device, which is a serial mechanism haptic desktop device. Restricting its wrist motion mechanically, it is able to track its own motion and reflect forces along $\vec{x}^1$, $\vec{x}^2$, and $\vec{x}^3$ axes. Real-Time Windows Target$^{\text{TM}}$ is used to compile the control algorithms developed in MATLAB© Simulink environment for both the master and the slave systems. The experimental set-up is illustrated in Fig. 4 in which the human operator, the master system, the slave system, and the remote environment relations are indicated.

The interface for the communication with the haptic devices under MATLAB© Simulink is provided by the QuaRC© software. The interfaces include blocks that enable the hardware to receive the encoder data and send the actuator commands in the joint space or in the task space. The hand motion tracking data for Novint Falcon device is received and the force demands are sent in the task space by using embedded coded kinematics and quasi-static force calculations within the QuaRC© software blocks. However, the slave system, Phantom Desktop device, is controlled in the joint space to accommodate the execution of the slave control algorithms described in this paper.

Both systems are placed in the same laboratory. However, in order to simulate the communication failures, the variable and constant time delays in communication line between the master and the slave systems are modelled and implemented in experiments. In this study, only a rectangular object with previously known properties is considered as the object to be placed in the slave system's environment.

https://doi.org/10.1017/S0263574715001010

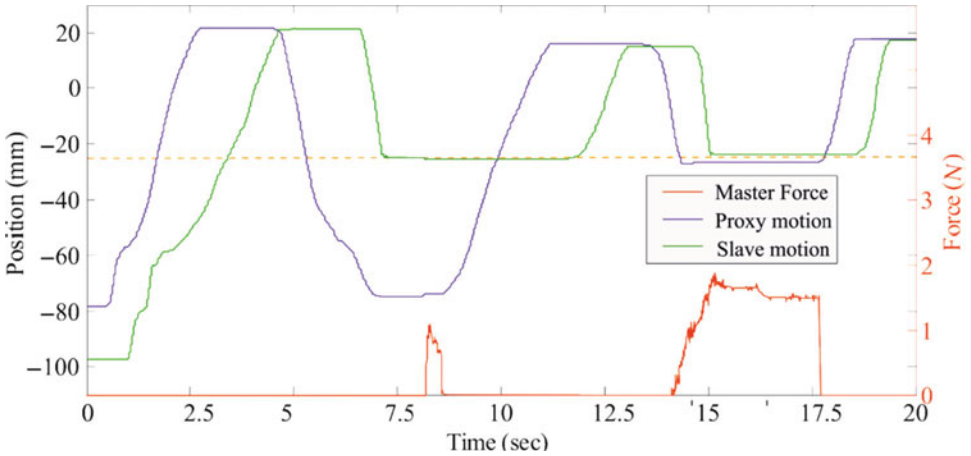

Fig. 5. The motion tracking performance and the forces applied to the user in surface detection case.

4. The Experimental Test Results

Two sets of experiments are conducted to test the model-mediation method against the variable time delays and test the effect of implementing an impedance control algorithm as the slave controller. In all experiments, the motion of the proxy is transmitted to the slave system in the position and the velocity levels to compensate for the possible offsets in the position tracking performance. In addition, the virtual model created within the master system, which is composed of the proxy and the detected surface representations, is projected as a visual feedback for the user. In the visual interface, the user interacts with the updated virtual surfaces through the proxy without experiencing any time delays. Hence, the user's perception of the slave environment, tele-presence level, was enhanced.

4.1. The experimentation under variable time delays in the communication line

Two conditions are tested in this set of experimentations. In the first condition, an object is placed in the workspace of the slave system at a previously unknown location. In the experiment, the user controls the slave to interact with this object. In the second condition, the slave system interacts with the previously placed object first and then the object is removed from the workspace. The tests are carried out with variable time delays that vary between 0.7 to 2 s.

The test result for the first condition is shown in Fig. 5. The position tracking along the $\vec{x}^3$-axis is provided only since the contact surface's normal is along the $\vec{x}^3$-axis. However, the user moves the slave through the master in 3D space during the experimental test. The initial contact of the slave system with the surface occurs just below −20 mm at 7 s. The surface location is also represented in Fig. 5 with the orange-coloured dashed line. As the slave system contacts the object for the first time, the surface location is acquired within the slave system and sent to the model within the master system as a model update with a time delay. As soon as the model is updated with a delay, the surface constraints are introduced to the master in accordance with the assumptions listed in Section 2.3 Model Construction. Hence, the master device starts to exert forces to the user along the direction of the contact surface's normal if the user moves the master in the opposite direction of the surface's normal. After the surface information is updated, in Fig. 5, the forces are exerted to the user after 8 s as a result of the motion of the user in the opposite direction of the contact surface's normal. Meanwhile, the surface constraint's position is preserved just below the proxy's position. The surface constraint is kept under the proxy until the proxy is moved by the user to a position above the measured surface position in the slave system's environment. The second collision takes place at 14 s in which the surface constraint is introduced to the master in the virtual model and the forces are reflected to the user without any delay. This is based on the previous knowledge of the surface constraint location since this information was received during the first collision. It can be observed just after the 17.5 s that as soon as the operator starts to move in the direction of the contact surface's normal, the force feedback is decreased to zero magnitude.

The surface removal test is initiated with an actual surface that had been introduced to the model within the master system. The results of the test are presented in Fig. 6. In between 1, 2, and 3 s, the

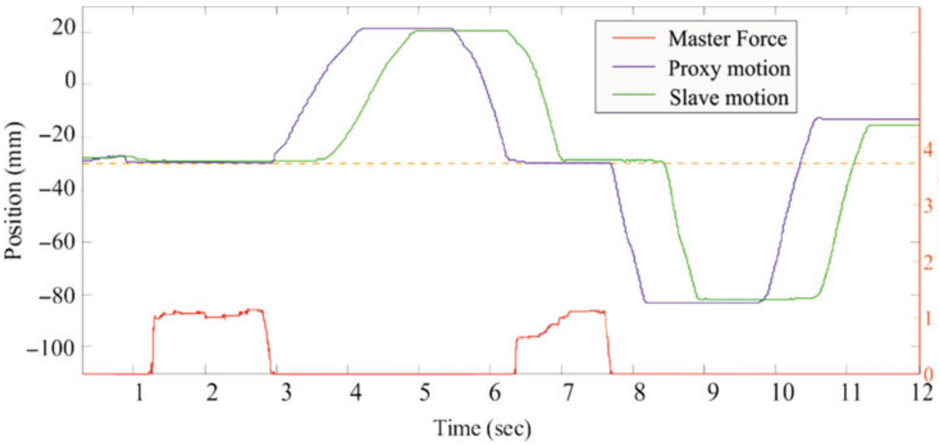

Fig. 6. The motion tracking performance and the forces applied to the user in surface removal case.

master system directs the slave system onto the previously known surface and the slave interacts with the actual surface. The force feedback signal during this period is a result of the motion of the user towards the opposite direction of the surface's normal. As it happened in the previous test, as soon as the user moves in the direction of the surface normal, the forced feedback magnitude is decreased to zero. Then, the user moves the slave off the surface and after 4 s, the actual surface (physical object in the slave system's workspace) is removed from the remote environment when the slave is no longer in contact with the surface. The position of the actual surface introduced in the beginning of the test is shown with the horizontal dashed line drawn in orange colour in Fig. 6.

After a couple of seconds of free motion, a collision happens after 6 s in the master system with the previously constructed virtual surface. Although the surface constraint is no longer present in the slave system's environment at that time, it is not yet removed from the model within the master system. Therefore, the user experiences forces as if the object in the slave system's environment still exists. A model update is sent to the model within the master system as the slave system reaches the virtual contact surface location at second 7 s and no object is detected by the contact estimation algorithm in that region. Until the update is transmitted at the 7.7 s, the user still perceives the surface constraint. After the update for no contact is received in the master system, the virtual surface in the model is removed at the 7.7 s and the free-motion tracking is continued within the limits of the workspace. The workspace limit is reached just after 8 s at just above −80 mm. The plateaus observed in the position of the master system in between 8 and 10 s and the position of the slave system between approximately 9 and 11 s are because of this workspace limitation. Since there was no physical interaction with an object during these periods, no force feedback is issued to the user.

Another motion tracking performance of the model-mediated teleoperation system in 3D space is presented in Fig. 7. Slave motion represents the motion realized by the slave device's end-effector, while the proxy motion represents the motion commands transmitted through the master subsystem. In this test, the slave follows the demand sent as the proxy motion with a variable time delay. It should be noted that the figure is drawn as a partial representation of the total task and also, the position tracking errors are in the range of less than 1 mm throughout the task.

The proxy penetrates the actual surface condition during the first collision since the surface contact information has not been received in the master subsystem yet due to time delays. The delayed contact information received in the master subsystem results in a model update. In the second collision with the surface, the proxy complies with the modelled virtual surface and does not penetrate the virtual surface. Consistent with previous examples, as the user drives the master device inside the virtual surface once again, collision forces are created on the master side to be reflected to the user while the proxy is still on the surface.

4.2. *The experimentation for the slave system's contact behaviour*

The effect of implementing an impedance controller is evaluated through conducting experimental tests with a collision case study when the impedance control is active and inactive. In both tests,

https://doi.org/10.1017/S0263574715001010

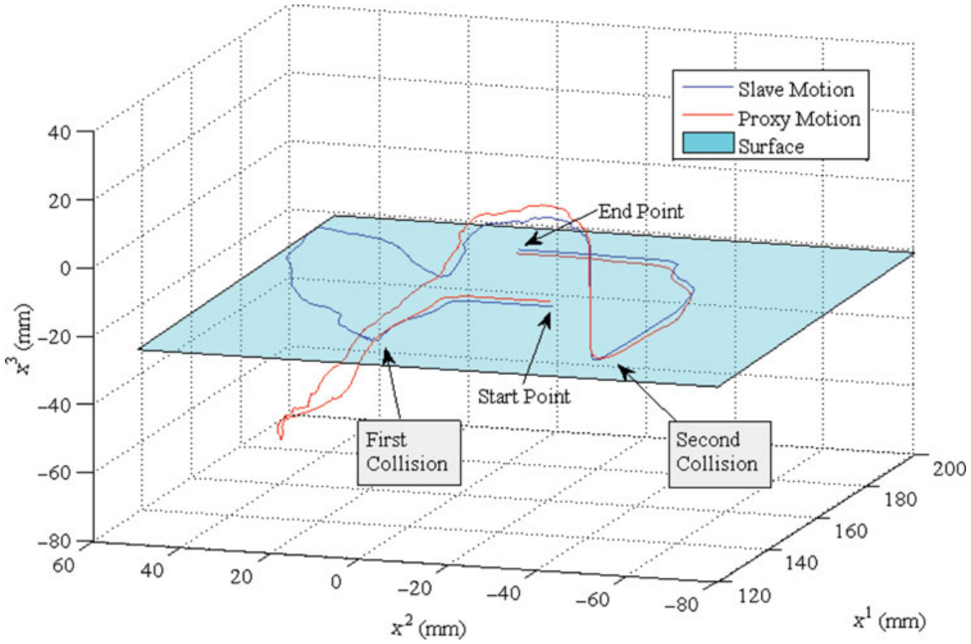

Fig. 7. The 3D motion tracking performance of the slave manipulator with surface detection.

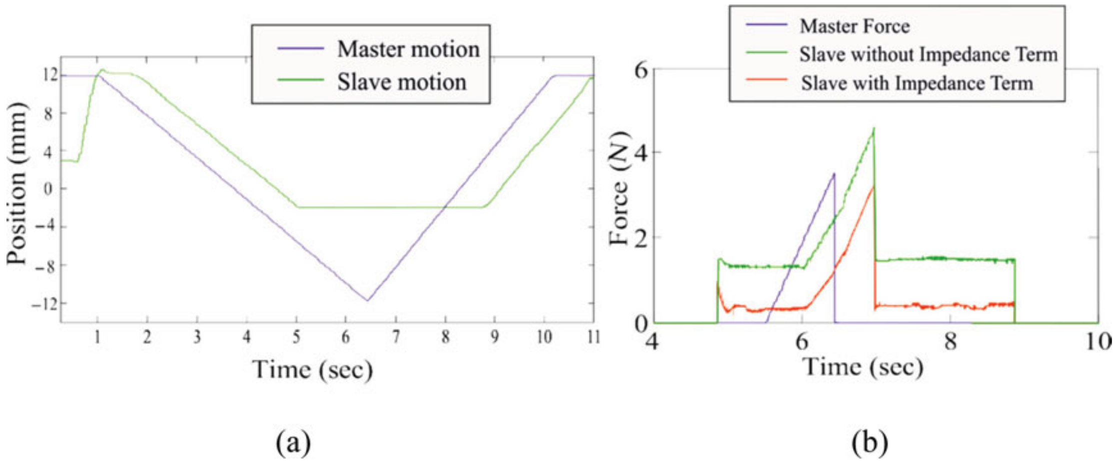

Fig. 8. (a) The position tracking of the master and the slave system for impedance controller test. (b) The force tracking performance of the slave with and without impedance controller.

there are constant communication line time delays. In order to test two cases while receiving the same inputs from the master system, a position profile is defined as the master system's motion and is forwarded to the slave system through the proxy. The test results are presented in Figs. 8(a) and (b) for the case when a pure motion controller is used and for the case when a second-order impedance term is used in the slave controller. The position tracking performance of the slave system along the $\vec{x}^2$-axis is shown in Fig. 8(a). The slave system tracked the motion of the master system with a constant 0.7 s time delay. The collision with the object present in the slave system's environment happens at 5 s and the slave preserved its position on the surface as demanded by the master until 9 s.

Since the master system's motion is defined to move in the opposite direction of the surface's normal until the 6.5 s, at 5.7 s when the contact information is received in the master system, the master system's force to be applied to the user is generated from the master system and sent to the slave system until 6.5 s. This information is the force trajectory to be tracked by the slave manipulator. Results for the force tracking performance along the $\vec{x}^2$-axis, which is the contact surface's normal, can be observed in Fig. 8(b). The demand sent from the master system is shown with the purple

https://doi.org/10.1017/S0263574715001010

colour. Results for force tracking performance of the slave system without the impedance controller is shown with the green colour and the tracking performance of the slave system with the impedance controller is shown with the colour red.

The slave collided with the object, which appears to be located below 0 mm with respect to the base frame, at 5 s. The slave force tracking results without impedance controller indicates that a higher amount of forces are exerted to the environment during the initial contact and remain on the object's surface until 9 s. After 7 s, there is a sudden drop in the demanded force due to the motion of the user in the opposite direction of the surface's normal. However, motion command to move away from the surface is not issued yet by the proxy because the proxy waits for the master to move above the virtual surface. Meantime, the slave robot does not lose contact with the surface in the slave environment. Only after the master moves above the virtual surface and the proxy starts to follow the master, the motion demands are issued to move the slave robot away from the surface. Although the master moves way from the surface at about 8 s, the slave starts to move away from the actual surface at 9 s due to the delay in communications.

Without the impedance controller, the slave system tracks the forces directed by the master system in an open-loop control fashion, in which the force demand is directly translated into the joint space by using the transpose of the Jacobian matrix. It is observed from Fig. 8, that there is an offset in the force tracking performance of the controller without the impedance term and higher forces than the demanded forces are exerted onto the slave environment. The test result with the impedance controller indicates that the force tracking improves, respectively. During the initial contact in the impedance controller test, a peak in the force is observed at the beginning, after which this overshoot in response is dampened out by the impedance controller.

5. Conclusions

In this work, the model-mediation method is re-formulated to be extended for multi-DoF teleoperation, in which an impedance controller is implemented for more stable interaction of the slave system with its environment. The proposed controller is implemented and tested on a three-DoF teleoperation system. The communication failures, which are the variable and the constant time delays in the communication line, are modelled and implemented in the experiments to evaluate the performance of the proposed controller. In order to comply with the remote environment, especially during initial contact, an impedance control algorithm is employed as the slave system controller and its effectiveness is evaluated under the constant time delays while the model-mediation algorithm is used for the teleoperation control system.

Since the Phantom Desktop device or slave system in this study, does not have a force transducer to measure forces exerted to the environment by the slave system, a contact and force estimation algorithm is integrated and used in all of the experiments. In addition, a virtual representation of the estimated model is constructed and used as a visualization aid for the operator during experiments to increase the level of tele-presence.

During variable time delays in the communication line, the proposed bilateral teleoperation control system preserved its stability, which allowed the operator to safely deliver commands to the slave and continue the operation stably. Compared to the conventional teleoperation system, the stability performance proved to be not affected from communication delays. A rectangular object in the slave system's environment is considered as the object to be interacted with and the control system is developed with respect to this constraint. In addition, it is deduced that making use of an impedance controller in the slave system prevents exertion of excessive forces to the environment especially when the slave system collides with the unknown objects in its environment. The test results revealed that the impedance controller enabled the slave system to follow the master system's force demands with better tracking performance with respect to the case where the master system's force demands are directly fed as external forces to be applied by the slave system.

The main advantage of the proposed teleoperation controller is that the controller's performance in terms of tracking and stability is independent of the type of the communication time delay being variable or constant. The proposed method has practical application potential especially when the slave environment is gradually changing and there are unknown and possibly large (in the range of seconds to minutes) communication delays between the master and the slave subsystems. Unlike the other passivity-based teleoperation controllers, in the proposed method, the forces from the slave

https://doi.org/10.1017/S0263574715001010

environment are not directly sent to the user with a time delay. As a result of this, interaction forces reflected to the user can be regulated independent of the actual interaction forces happening in the slave side. Therefore, the user has an increased comfort during the operation not being subjected to instantaneous excessive force feedback. Virtual representation of the slave environment also helps the user to get a better sense of the interaction with the slave environment through visual feedback. Overall, the proposed method provides a better alternative for move-wait types of teleoperation scenarios with large communication delays. Possible fields for application may be space or underwater missions.

However, further studies may be carried out for increased performance. As an extension of this work and to create more precise feedback to the master system as model updates, different sensors with better precisions can be integrated to the slave system. These sensors could be force sensors and advanced visual sensors. The visual sensors such as RGBD cameras can be used for pre-construction of the slave system's environment model and force sensors can be used for more accurate interactions with environment. In this way, the restriction given in Assumption 3 that the normal of the surfaces should be along the Cartesian axes can be removed. Another future study can be conducted for user tests to evaluate the usability of the model-meditated teleoperation against the teleoperation systems whereby other controllers, such as wave variable technique, are used.

Acknowledgements

The authors would like to thank to The Scientific and Technological Research Council of Turkey for funding the research presented in this work (grant number 113E147).

Supplementary material

To view supplementary material for this article, please visit http://dx.doi.org/10.1017/S0263574715001010.

References

1. J. Cui, S. Tosunoglu, R. Roberts, C. Moore and D. W. Repperger, "A Review of Teleoperation System Control," *Proceedings Florida Conf. Recent Adv. Robot.*, Boca Raton, Florida (2003).
2. M. Dede and S. Tosunoglu, "Fault-tolerant teleoperation systems design," *Ind. Robot J.* **33**(5), 365–372 (2006).
3. D. A. Lawrence, "Stability and transparency in bilateral teleoperation," *IEEE Trans. Robot. Automat.* **9**(5), 624–637 (1993).
4. R. J. Anderson and W. Spong, "Bilateral control of teleoperation with time delay," *IEEE Trans. Autom. Control* **34**(5), 494–501 (1989).
5. G. Niemeyer and J. Slotine, "Using Wave Variables for System Analysis & Robot Control," *Proceedings IEEE Int. Conf. Robot*, Albuquerque, New Mexico, vol. 2, (1997) pp. 1619–1625.
6. S. Munir and W. Book, "Control techniques and programming issues for time delayed internet based teleoperation," *ASME J. Dyn. Sys. Meas. Control* **125**(2), 157–277 (2003).
7. N. Chopra, M. W. Spong, S. Hirche and M. Buss, "Bilateral Teleoperation over the Internet: The Time Varying Delay Problem," *Proceedings Amer. Control Conf.*, Denver, Colorado, vol. 1, (2003) pp. 155–160.
8. P. Mitra and G. Niemeyer, "Model-mediated telemanipulation," *Int. J. Robot. Res.* **27**, 253 (2008).
9. B. Willaert, J. Bohg, H. Van Brussel and G. Niemeyer, "Towards Multi-DOF Model Mediated Teleoperation: Using Vision to Augment Feedback," *IEEE Int. Workshop on HAVE*, Munich, Germany (2012).
10. G. Zeng and A. Hemami, "An overview of robot force control," *Robotica* **15**, 473–488 (1997).
11. H. Kazerooni, T. B. Sheridan and P. K. Houpt, "Robust compliant motion for manipulators-part I: The fundamental concepts of compliant motion; part II: Design method," *IEEE J. Robot. Automat.* **2**(2), 83–105 (1986).
12. N. Hogan, "Impedance control: An approach to manipulation-Part I: Theory; Part II: Implementation; Part III: Applications," *ASME J. Dyn. Sys. Meas. Control* **107**(1), 124 (1985).
13. P. Mitra and G. Niemeyer, "Dynamic Proxy Objects in Haptic Simulations," *IEEE Conf. Robot Autom. Mechatronics*, Singapore (2004) pp. 1054–1059.
14. S. Dragoljub, "Contact Stability Issues in Position Based Impedance Control: Theory and Experiments," *Proceedings IEEE Conf. on Robot. and Automat.*, Minneapolis, Minnesota, vol. 2, (1996) pp. 1675–1680.
15. D. A. Lawrence, "Impedance Control Stability Properties in Common Implementations," *Proceedings. IEEE Conf. on Robot Autom.*, Philadelphia, Pennsylvania, vol. 2, (1988) pp. 1185–1190.
16. O. Kiyoshi, M. Miyazaki and M. Fujita, "Hybrid Control of Force and Position without Force Sensor," *Proc. IEEE Conf. on Indus. Elec. Control, Instrumen. and Automat.*, San Diego, California, vol. 2, (1992) pp. 670–675.

17. R. J. Adams and B. Hannaford, “Stable haptic interaction with virtual environments,” *IEEE Trans. Robot. Automat.* **15,** 3465–473 (1999).
18. H. C. Cho and J. H. Park, “Stable bilateral teleoperation under a time delay using a robust impedance control,” *Mechatronics* **15**(5), 611–625 (2005).
19. B. Willaert, H. V. Brussel and G. Niemeyer, “Stability of model-mediated teleoperation: Discussion and experiments,” *Haptics: Perception, Devices, Mobility, Commun.* **7282**, 625–636 (2012)
20. F. B. Llewellyn, “Some Fundamental Properties of Transmission Systems,” *Proceedings IRE*, **40**, 271–283 (1952).

https://doi.org/10.1017/S0263574715001010