Title:

Increasing stability in model-mediated teleoperation approach by reducing model jump effect

Behnam Yazdankhoo, Borhan Beigzadeh*

School of Mechanical Engineering, Iran University of Science and Technology, Tehran, Iran, P.O.B. 16765163

*Corresponding author. Tel.: +98 21 77240094; Mobile NO.: +98 9183412475; Fax: +98 21 73021587

Email addresses: b_yazdankhoo@mecheng.iust.ac.ir (Behnam Yazdankhoo), b_beigzadeh@iust.ac.ir (Borhan Beigzadeh)
Abstract

Model-mediated teleoperation is a predictive control approach for controlling haptic teleoperation systems whereby the environment force is virtually located on master side in order to increase the stability and transparency of the system. This promising approach, however, results in new challenges. One pivotal challenge is the model jump effect, which stems from the delay in correct creation of the virtual environment. Previous works have endeavored to reduce this effect; however, they either led to transparency decrease or assumed simplified environment models. In this paper, we propose a control approach for this aim based on the idea of decoupling. This means that when a new environment has been identified, the operation is interrupted and no signal is transmitted between master and slave sides. During this time, both sides are controlled by their own sliding mode controllers until the system reaches stability. The main advantage of this method is its independence from environment type, which makes it usable for different kinds of applications. To verify the effectiveness of the proposed approach simulation tests are conducted. The results show the system is stable in interaction with hard and soft environments in presence of large time delays in communication channels.

Keywords: predictive control, model-mediated teleoperation, transparency, model jump, decoupling method, sliding mode control

1. Introduction

Design and control of haptic teleoperation systems has gained many researchers’ attention during the past decades. Conventional control methods had long been utilized in teleoperation systems as control strategies by adopting classical control architectures, such as position-position, force-position, or 4C (four-channel) [1]. All these architectures have the method of sending the intended signals (position and/or force) through communication channels to the other side, where the delayed signals are employed to design the desired controller, in common. In position-position architecture position signals are sent from each side to the other side. This architecture is used in [2], for instance, in order to design a teleoperation system. Another prevalent type of architecture which resembles 4C is the one which sends position and force signals from master to slave side, but only force signal from slave to master side. Many teleoperation systems have been designed based on this architecture, such as [3, 4]. These control architectures, however, fail to provide high transparency. This happens because a compromise should always be considered between the two conflicting objectives transparency and stability [5]. It should be noted that by transparency, we mean the quality of position and force tracking of the system. In other words, the extent to which the slave and master robots are able to replicate the position of the master and the environment force, respectively, is called transparency. Although quantitative definitions have also been presented [1], we stick to the qualitative definition of transparency herein since it is sufficient to describe the results of the proposed method.
For conquering the deficiency mentioned in the previous paragraph, predictive control approaches were introduced in teleoperation systems, primarily by researches such as [6] which attempted to exploit online estimated parameters instead of actual transmitted ones. The main advantage of these approaches is that they are able to circumvent the time delay which exists in the communication channels, thus making the system more transparent and stable. Various predictive control methods have been proposed in the literature up to now [7], in which the main focus of the researches has been on predicting the environment force on master side, which is typically called model-mediated teleoperation. In other words, according to this approach, the environment impedance is estimated on slave side and then transmitted to master side, where the force is locally recreated based on the position and velocity of the master robot. A general scheme of this approach is depicted in Figure 1.

In spite of providing high transparency, the model-mediated approach creates some new challenges. One of the main challenges is that when the slave robot encounters a new environment, obtaining the true environment impedance may take some time and also rendering them to the master side is delayed due to the nature of the communication channels. During this period, the master side is not aware of the true environment parameters and, thus, the local force cannot be correctly created. This time interval which is called the period of model mismatch, puts the system in an unstable situation and needs to be suitably controlled [8].

Stability of model-mediated teleoperation has been addressed by some researches in recent years. Some of the proposed approaches provide stability at the expense of transparency, such as [9] which proposes the idea of displacement of the local virtual environment in the direction of the movement of the master robot to stabilize the system when the first contact on the slave side is occurred. Passivity-based model updating is another technique which is adopted in [10] by inducing an additional damping factor in the virtual environment when a change in model happens. This may also lead to a decrease in transparency depending on the type of model update. The other proposed approaches, however, have reached the ultimate goal of stabilizing the system through the assumption of a simple environment model [11, 12]. As a consequence, presenting a control method which can conserve system stability and transparency together regardless of the environment type seems necessary.

In this paper, a control strategy for the period of model mismatch (or equivalently transition state) is incorporated so that the system stability can be preserved even for large amounts of time delay. The strategy is based on the idea of full decoupling whereby master and slave sides are decoupled when a new environment appears. In that period, no signal is transmitted between master and slave sides and both sides are separately controlled by their own controllers. These controllers are also independent of environment type, so the system is able to interact with both hard and soft environments.

The rest of this paper is organized as follows. Section 2 describes the dynamic modeling of the system. Section 3 is dedicated to explaining the estimation algorithms used to estimate the
impedance for both hard and soft environments. Section 4 describes the controllers designed for the different states of the system and provides a discussion on overall stability of the system. Finally, in section 5 the simulation results are provided, while conclusions and suggestions for future researches are included in section 6.

2. System modeling

2.1. Master side modeling

The dynamics of the master robot is considered as a one-degree-of-freedom, linear second-order system in time domain, as indicated in (1).

\[ m_m \ddot{x}_m + b_m \dot{x}_m + k_m x_m = F_h^* + U_m \]  

(1)

where \( m \), \( b \) and \( k \) denote the mass, damping coefficient and stiffness, respectively. \( F_h^* \) is the interaction force between the operator and the master robot and \( U \) is the control input. \( x \) represents the position of the robot and subscript \( m \) denotes the master robot.

For simulation purposes, it is necessary to obtain a model for human or operator hand. Several models have been proposed for representing the dynamic behavior of human hand during interaction with teleoperation systems. The first and possibly most well-known model is the linear second-order one proposed in [13]. However, the authors of [14] proved that a suitable model should be of relative order one. Hence, they derived a two-parameter model based on experimental data which can be written as

\[ b_a \dot{x}_m + k_a x_m = F_h - F_h^* \]  

(2)

where \( b_a \) and \( k_a \) represent the damping and stiffness coefficients of hand, respectively, and \( F_h \) is the human exogenous force. It should be pointed out that in (2) it is assumed that the operator continuously and completely grasps the end effector of the master robot, in such a way that the position of his/her hand is always equal to the position of the robot.

Combining (1) and (2), we can obtain the dynamics of the master side, which is given by

\[ m_m \ddot{x}_m + b_{r,m} \dot{x}_m + k_{r,m} x_m = F_h + U_m \]  

(3)

where \( b_{r,m} \triangleq b_m + b_a \) and \( k_{r,m} \triangleq k_m + k_a \) are assumed.

2.2. Slave side modeling

The dynamics of the slave robot is considered as a one-degree-of-freedom, linear second-order system in time domain, as indicated in (4).
\[ m_s \ddot{x}_s + b_s \dot{x}_s + k_s x_s = U_s - F_e \]  

(4)

where \( m, b \) and \( k \) denote the mass, damping coefficient and stiffness, respectively. \( F_e \) is the environment force and \( U \) is the control input. \( x \) specifies the position of the robot and subscript \( s \) denotes the slave robot.

The environment force can be represented by the augmented Kelvin-Voigt (KV) model for hard environments and the augmented Hunt-Crossley (HC) model for soft environments, which are given by (5) and (6), respectively.

\[
F_e = \begin{cases} 
  K_{KV} \delta x_s + B_{KV} \dot{x}_s, & \delta x_s > 0 \land \dot{x}_s \geq 0 \\
  K_{KV} \delta x_s, & \delta x_s \geq 0 \land \dot{x}_s < 0 \\
  0, & \text{else}
\end{cases}
\]

(5)

\[
F_e = \begin{cases} 
  K_{HC} \delta x_s'' + B_{HC} \dot{x}_s'' \dot{x}_s, & \delta x_s \geq 0 \land \dot{x}_s \geq 0 \\
  K_{HC} \delta x_s'', & \delta x_s \geq 0 \land \dot{x}_s < 0 \\
  0, & \text{else}
\end{cases}
\]

(6)

In (5) and (6) \( K \) and \( B \) denote stiffness and damping coefficients, \( n \) is a constant and \( \delta x_s \) represents the environment compression.

The selection of the proper model between KV and HC for the environment is carried out by the hybrid object modeling approach which is proposed in [15] and augmented in [16]. This approach is represented by the \( S \) operator

\[
S(K_{KV}, t) = \begin{cases} 
  KV, & K_{KV} \geq K_{th} \land \delta t \leq t_{th} \\
  HC, & \text{else}
\end{cases}
\]

(7)

where \( K_{th} \) and \( t_{th} \) are stiffness and time thresholds, respectively, and \( \delta t \) is defined as

\[
\delta t \triangleq t - t_w
\]

(8)

where \( t_w \) specifies the time of the first interaction between the slave robot and the environment.

3. Estimation of environment parameters

As mentioned in section 1, in model-mediated teleoperation the environment is virtually created on master side. Therefore, a particular approach should be adopted for this purpose from different methods proposed in the literature. Relevant approaches can be divided into two main categories of model-based and model-free. An example of a model-free approach is [17] in
which the environment force is predicted by a neural network on slave side and is reproduced on master side. However, our approach in this paper is model-based. Thus, an algorithm first estimates the environment parameters ($K_{KV}$ and $B_{KV}$ for $KV$ and $K_{HC}$, $B_{HC}$ and $n$ for $HC$) on slave side, and then the estimated parameters are sent to the master side in order to create the virtual environment.

Several methods exist for determining the parameters of a system. There are numerical approaches, such as [18], which are mainly proposed for identification of parameters of relatively complicated mechanical systems. Nonetheless, based on the dynamics of the environments defined by (5) and (6), a simpler algorithm can be employed for the sake of parameter estimation in this paper. Hence, the SPRLS [19] which is a simple, yet powerful, recursive algorithm seems a suitable choice. According to this algorithm, the actual and estimated systems should be linearized in the following form

$$y = \theta^T \Phi$$  \hspace{1cm} (9)

$$\hat{y} = \hat{\theta}^T \Phi$$  \hspace{1cm} (10)

where $y$ and $\hat{y}$ are the actual and estimated outputs and $\theta$ and $\hat{\theta}$ are actual and estimated parameters, respectively, and $\Phi$ is the vector of inputs.

If we define the estimation error as $\hat{e} \triangleq y - \hat{y}$, then the vector $\hat{\theta}$ can be obtained from the following recursive relations

$$L_i = P_{i-1} \Phi_i (1 + \Phi_i^T P_{i-1} \Phi_i)^{-1}$$ \hspace{1cm} (11-a)

$$P_i = (I - L_i \Phi_i^T) P_{i-1} + \beta \text{NINT}(\gamma \hat{e}_{i-1}^2) I$$ \hspace{1cm} (11-b)

$$\hat{\theta}_i = \hat{\theta}_{i-1} + L_i (y_i - \Phi_i^T \hat{\theta}_{i-1})$$ \hspace{1cm} (11-c)

where $\beta$ and $\gamma$ are constants, $I$ is the identity matrix and $i = 1, 2, \ldots$. NINT(.) operator is also defined as

$$\text{NINT}(x) \triangleq \begin{cases} x, & x \geq 0.5 \\ 0, & 0 \leq x < 0.5 \end{cases}$$  \hspace{1cm} (12)

The next step is to adapt the relations (9) and (10) to the $KV$ and $HC$ models. For $KV$ this can conveniently be done due to its inherent linearity, so we have

$$y_{KV} = F_e$$  \hspace{1cm} (13-a)

$$\Phi_{KV} = [\delta x \dot{x}]^T$$  \hspace{1cm} (13-b)
\[ \theta_{KV} = \begin{bmatrix} K_{KV} & B_{KV} \end{bmatrix}^T \] (13-c)

However, adaptation to HC model is not attained as easily as KV. In [20] and [21], double-stage and single-stage methods are proposed, respectively, in order to identify linearized HC parameters to be used in the form of (9) and (10). But Schindeler and Hashtrudi-Zaad utilized a novel and robust procedure for linearizing the HC model [22]. In this method, which is called polynomial linearization, the term \( \delta x_s^n \) in the HC model is expanded based on Taylor Series around its center point of working space and yields

\[ (\delta x)^n \approx \nu_0 + \nu_1 \delta x + \nu_2 (\delta x)^2 \] (14)

where \( \nu_0, \nu_1 \) and \( \nu_2 \) are coefficients that are obtained by arranging the Taylor expansion according to (14). Now the HC model can be rewritten as

\[
F_e = \begin{cases} 
K_{HC}(\nu_1 \delta x_s + \nu_2 (\delta x_s)^2) + B_{HC}(\nu_1 \delta x_s + \nu_2 (\delta x_s)^2) \frac{\delta x_s}{\delta x_s} + \frac{\delta x}{\delta x_s}, & \delta x_s \geq 0 \wedge \dot{x}_s \geq 0 \\
K_{HC}(\nu_1 \delta x_s + \nu_2 (\delta x_s)^2), & \delta x_s \geq 0 \wedge \dot{x}_s < 0 \\
0, & \text{else}
\end{cases}
\] (15)

where \( \nu_0 = 0 \) is assumed. Consequently, the linearized relations for the HC model are obtained as (16).

\[ y_{HC} = F_e \] (16-a)

\[ \Phi_{HC} = \begin{bmatrix} (\delta x)^2 & \delta x & (\delta x)^2 \dot{x} & (\delta x) \ddot{x} \end{bmatrix}^T \] (16-b)

\[ \theta_{HC} = \begin{bmatrix} K_{HC} \nu_2 & K_{HC} \nu_1 & B_{HC} \nu_2 & B_{HC} \nu_1 \end{bmatrix}^T \] (16-c)

### 4. Control approach and stability analysis

#### 4.1. Controller design for steady state

The term steady state is adopted from [8] and by that, we mean the time when the slave robot is not experiencing new contact with an environment. In other words, the slave robot either is in free motion or has previously encountered a new environment and is now just maintaining the contact. In this situation, regular control approaches suffice. Although any suitably designed controller can be used, in this paper impedance control and sliding mode control are implemented for master and slave sides, respectively. Impedance control has the property of rendering the desired impedance on master side, while with sliding mode controller the slave is able to asymptotically track the position of the master robot.
Before continuing with design of controllers, it should be noticed that in this paper, the delayed signals between the master and slave sides are designated by the superscripts d and D, as illustrated in Figure 2 and by the following relations

\[ X(t - \tau_m) \triangleq X^d(t) \]  

(17)

\[ Y(t - \tau_s) \triangleq Y^D(t) \]  

(18)

where \( \tau_m \) and \( \tau_s \) are communication time delays from master to slave and slave to master, respectively, and \( X \) and \( Y \) are two arbitrary parameters.

For impedance control of master side, a reference impedance behavior should be first defined [23, 24]. So we have

\[ M\dot{x}_m + B\dot{x}_m + Kx_m = F_h - \hat{F}_{e,m} \]  

(19)

where \( F_h \) is the human or operator force and

\[
\hat{F}_{e,m} = \begin{cases} 
\hat{\Phi}_{s,KV}^D T \Phi_{m,KV}, & S(K_{KV}, t) = KV \\
\hat{\Phi}_{s,HC}^D T \Phi_{m,HC}, & S(K_{KV}, t) = HC 
\end{cases}
\]  

(20)

where the subscripts \( m \) and \( s \) mean that the parameter should be inserted from or calculated on master and slave sides, respectively. \( M, B \) and \( K \) are the reference mass, damping coefficient and stiffness, respectively. Also \( \hat{F}_{e,m} \) represents the locally recreated environment force on master side.

The term \( \Phi_{m,HC} \) in (20) includes the term \( \delta x_m \), as can be regarded in (16-b), which should be obtained from (21)

\[ \delta x_m \triangleq x_m - x_w \]  

(21)

where \( x_w \) is the virtual environment position which is calculated on slave side (refer to section 4.2).

Finally, combining (3) and (19), the control input for master side is attained.

\[ U_{m,st} = (b_{,m} - \frac{m_{,m}}{M} B)\dot{x}_m + (\frac{m_{,m}}{M} - 1)F_h - \frac{m_{,m}}{M} \hat{F}_{e,m} + (k_{t,m} - \frac{m_{,m}}{M} K)x_m \]  

(22)

where subscript \( st \) refers to the steady state.
For sliding mode control of slave side, the error should be defined first. Here, the definition 
\[ e_{st} \triangleq x_s - x^d_m \] is used [25]. Then, the sliding surface is obtained in the following form
\[ s_{st}(x,t) \triangleq \dot{e}_{st} + \lambda_{st} e_{st} \] (23)
where \( \lambda_{st} \) is strictly positive. Ultimately, the slave control signal can be derived by implementing the approach explained in the appendix
\[ U_{st,sl} = \hat{b}_x \ddot{x}_s + \hat{k}_x x_s + \hat{F}_e + \hat{m}_s (\ddot{x}^d_m - k_{gain,s} \text{sat}(\frac{s_{st}}{\varphi_{st}}) - \lambda_{st} \dot{e}_{st}) \] (24)
where \( \varphi \) is boundary layer width for reducing the chattering phenomenon, \( \text{sat}(.) \) is the saturation function and the accent \( ^\wedge \) refers to the uncertain value of the corresponding parameter. Explanation of the parameter \( k_{gain,s} \) can be found in the appendix. Furthermore, if one cannot or do not desire to use \( \ddot{x}^d_m \) in (24) directly, he/she can replace it with the following relation which is the delayed rearranged form of (19)
\[ \ddot{x}^d_m = \frac{1}{M} (F^d_{r} - \hat{F}_{e,m} - B\ddot{x}_m^d - K\dot{x}_m^d) \] (25)

4.2. Locating virtual environment on master side

In teleoperation systems which are designed based on model-mediated approach, locating the virtual environment on the correct position on master side can greatly influence the transparency of the system. In fact, if the virtual environment is wrongly placed, \( \delta x_m \) in (21) and subsequently the vector \( \Phi_m \) in (20) will deviate from their real value and, thus, the calculated force on master side will not follow the actual force on slave side.

Until now, different researches have been conducted in line with this aim. One of the first approaches is based on [26], which assumes the position of the virtual environment as one of the environment parameters (as in (13-c) and (16-c)) and estimates it by an estimation algorithm. This approach requires the environment model to be as simple as possible, which is not always the case. More recent researches have also proposed the use of additional sensors, like vision or proximity sensors, on slave side for prior detection of the environment, such as the work done in [9].

However, the authors have previously proposed a method which is able to define the location of the virtual environment and the time of collision with environment on slave side. This method, which is expressed in the following form, leads to the maximum possible transparency [16].
with $X_{w,j} = \begin{cases} x_{s,j-1} & F_{e,j} > F_{th} \wedge |F_{e,j}| \leq F_{th} \\ X_{w,j-1} & F_{e,j} > F_{th} \wedge |F_{e,j}| > F_{th} \\ \Lambda & t_j \end{cases}$, (26)

Even though this method works well for relatively soft environments, it leads to model jump in interaction with hard environments[16]. Model jump is a phenomenon which produces a relatively high amount of force in the system in a limited time, thereby making the initial moments of the interaction hazardous for a teleoperation system.

Researches like [9] have previously considered reducing the effect of the model jump in the system. But the main challenge still exists; generally model jump and system transparency adversely affect each other. Therefore, if we aim at decreasing the model jump phenomenon to the most possible extent, we should provide a separate control approach for the initial moment of contact with a new environment, which is discussed in the following section.

4.3. Controller design for transition state

The term transition state is adopted from [8] and by that, we mean the initial time of interaction with a new environment. In this situation as mentioned before, the system may experience large contact forces which may be extremely detrimental to it. This is generally called model jump effect. To prevent system from being influenced by this effect, we propose a novel control approach for transition state in this section, which is based mainly upon decoupling master and slave sides.

Mitra and Niemeyer proposed a technique in [27] whereby the virtual environment is gradually moved toward the correct position using a sliding surface (such as (23)) when the first contact with environment occurs. In this approach, however, the master main controller is still active during transition state, which results in degradation of transparency. To compensate for this, master and slave sides need to be fully decoupled during transition state. This means that the controller which is active during steady state should be deactivated during transition state.

To the authors’ knowledge, the idea of decoupling was first implemented on a teleoperation system by Smisek et al. [11]. However, the control strategy adopted there was different and also only relatively stiff environments with elastic model were considered, i.e. $F_e = K\delta x_s$. In this paper, we extend this approach to a general case which is independent of the environment model.
and, moreover, guarantees system stability during transition state while maintaining high system transparency. A general scheme of the decoupling method is depicted in Figure 3.

Before designing controllers, a precise definition for transition state is presented. Transition state starts from the initial moment of interaction with a new environment and finishes at the time when both sides have become stable or, in other words, have returned to their initial contact positions.

For controlling the system during transition state, we utilize sliding mode control for both master and slave sides. Firstly, the error and sliding surface should be defined. In (23) the error was considered such that slave robot tracked the delayed position of the master robot, whereas here the error must be defined with the aim of reaching the location of the initial contact. Hence, the error definitions for master and slave sides are

\[ e_{s_{tr}} = x_s - x_w \]  
\[ e_{m_{tr}} = x_m - x_w^P \]  

where the subscript tr denotes the transition state and subscripts m and s denote master and slave sides, respectively.

Secondly, the sliding surfaces are defined as follows.

\[ s_{s_{tr}}(x, t) = \dot{e}_{s_{tr}} + \lambda_{s_{tr}} e_{s_{tr}} \]  
\[ s_{m_{tr}}(x, t) = \dot{e}_{m_{tr}} + \lambda_{m_{tr}} e_{m_{tr}} \]  

Finally, similar to the procedure for (24), the control signals for both sides in transition state can be achieved as mentioned in the appendix.

\[ U_{s_{tr}} = \hat{b}_s \dot{x}_s + \hat{k}_s x_s + \hat{F}_e + \hat{m}_s (-k_{gain_{s_{tr}}} \text{sat}(\frac{s_{s_{tr}}}{\varphi_{s_{tr}}}) - \lambda_{s_{tr}} \dot{e}_{s_{tr}}) \]  
\[ U_{m_{tr}} = \hat{b}_m \dot{x}_m + \hat{k}_m x_m - \hat{F}_e m + \hat{m}_m (-k_{gain_{m_{tr}}} \text{sat}(\frac{s_{m_{tr}}}{\varphi_{m_{tr}}}) - \lambda_{m_{tr}} \dot{e}_{m_{tr}}) \]
Note that \( x_w \) and \( x_w^D \) are constants by definition (as defined in (26)). So taking the derivatives of the defined errors in (27) and (28), we can easily conclude that \( \dot{e}_{s,ir} = \dot{x}_s \) and \( \dot{e}_{m,ir} = \dot{x}_m \).

It can be obviously seen that (31) and (32) do not contain any of the environment parameters. Therefore, this control strategy can be incorporated in a teleoperation system that deals with not only the environments defined by (5) and (6) (which can be indeed a good approximation for most environments), but also any other environment with different dynamics.

The important point which is considered in transition state is that the operator exerts no force during this period [11]. In other words, when the operator notices the first signal of collision between the slave robot and the environment, he/she ceases pushing the master robot forward while still completely grasping it. When the transition state has finished, the operator reinitializes the operation.

### 4.4. Discussion on system stability

Overall stability of a teleoperation system should be discussed from two aspects. One aspect is regarding the stability of the controllers designed on master and slave sides, and the other is the stability of the communication channels. In this paper, the former aspect is readily fulfilled due to utilizing impedance and sliding mode controllers, which are well known to be stable if tuned suitably. The latter, however, requires a deeper analysis, since it deals with the delayed signals which are transmitted through communication channels.

Considering classical control architectures described in section 1, generally two well-known approaches are employed to analyze the stability of the communication channels in a teleoperation system, though other approaches exist, too. The first one is absolute stability, the necessary and sufficient conditions for which is provided by Llewellyn's stability criteria [28]. An example of a design based on absolute stability can be found in [3]. The second one is passivity, which is adopted to analyze the stability of the system from energy generation point of view. An example of a passivity-based design can be found in [29]. These methods, however, basically make effort to stabilize a teleoperation system when merely delayed signals are available at each side of it, which is not the case when it comes to model-mediated teleoperation.

As explained before, in model-mediated teleoperation the master side is in interaction with a local virtual environment instead of the delayed force signal from the slave side. Accordingly, the master control loop is closed locally and, considering passive human operator and environment, the system is always stable (the master local loop could be inferred from Figure 1, too). The only remaining stability issues are thus as follows. First, the estimation algorithm must be convergent, which is attained in this paper by using the SPRLS algorithm. Second, the transition state must be controlled so that it does not destabilize the system, which is addressed in this paper.
Finally, it should be noted that the aforementioned analysis pertains to the slave-to-master direction of the teleoperation system applied in this work. The stability analysis regarding the master-to-slave direction is analogous to the previous works in this field, for instance [30]. Also note that some researches such as [10] have recently extended the concept of passivity to the context of model-mediated teleoperation. Despite having the same name and overall concept, these works should be separated from the conventional passivity analysis mentioned in the beginning of this section due to the difference in their control architectures.

5. Simulation results

To study the validity of the proposed approach, simulation results are presented for the designed teleoperation system in this section.

For simulating human operator force, the following function is considered

\[
F_h = \begin{cases} 
2t, & 0 \leq t < 2 \\
4, & 2 \leq t < T \wedge t \in \text{steady state} \\
0, & 2 \leq t < T \wedge t \in \text{transition state} \\
\frac{1}{7}(t-6)+4, & T \leq t < 13 \\
\frac{-5}{2}(t-15), & 13 \leq t \leq 15 
\end{cases}
\] (33)

where \( T \) is the time when the transition state finishes, and can be tuned based on the operation circumstances. Here we determined \( T = 6 \) s by trial and error. Also, when \( F_h \) is zero, it indicates that the human operator has stopped pushing forward the master robot and is waiting for the system to reach a stable point. Note that in (33) \( t \) is in seconds.

A time delay of 0.5 s for both communication channels is considered (equivalent to a round-trip time delay of 1 s), which means \( \tau_m = \tau_s = 0.5 \) s. Also an uncertainty interval is assumed for the mass of the slave robot as follows

\[1.058 \text{ kg} \leq m_s \leq 1.267 \text{ kg}\] (34)

Furthermore, in order to validate the generality of the proposed approach, two sample hard and soft environments are considered for interaction with the slave robot. Although hard and soft are intrinsically qualitative terms, it is common in the literature to assume an environment with stiffness bigger than 2500 N/m (i.e. \( K > 2500 \text{ N/m} \)) as hard and otherwise as soft [15, 20]. For damping coefficient, however, such a general convention does not exist; but it seems a good estimation to consider the ratio \( \frac{K}{B} \approx 30 \text{ s}^{-1} \) [15]. It is also a suitable approximation to consider
$1 < n < 2$ for the exponent $n$ [20, 22]. The properties of the simulated environments in addition to other assumptions for the system are mentioned in Tables 1 to 6. The tuning of the sliding mode controllers (Tables 3 and 4) and the parameters of the SPRLS algorithm (Tables 5 and 6) was conducted manually.

The results of the simulation are also illustrated in Figures 4 to 11. From Figures 4 and 8, it can be seen that when a new environment is faced, whether hard or soft, the transition state starts and continues until both master and slave sides become stable. After that, the system goes to the steady state again while normal operation is being carried out. Two important points can be inferred from these two figures: firstly, the master robot is able to track the environment force ahead accurately, due to the predictive architecture used, during the steady state (which shows high transparency) and secondly, the model jump effect has been drastically decreased, especially if compared with the results presented in [16] (which shows increased stability).

Figures 5 and 9 show that for both hard and soft environments, the slave robot tracks the position of the master robot behind (due to the inherent time delay in the communication channels). In Figures 6 and 10, the position error between the master and slave robots are depicted which further proves the quality of the position tacking of the system for both hard and soft contacts and during free motion (no contact with environment). Finally, the control input of the slave robot is illustrated in Figures 7 and 11 during contact with hard and soft environments, respectively. It should be pointed out that the master control input is equal to the master reaction force (in magnitude) as shown in Figures 4 and 8. Also note that transparency cannot be considered for transition state due to its nature. In simple words, transparency can be defined when signals are being transmitted between two sides of a teleoperation system, while no signal is transmitted during transition state by definition.

Despite high transparency, it can be observed that after the end of the transition state during interaction with the soft environment (Figure 8), transparency is not as high as when it is in interaction with the hard environment for some time. The reason is that the system has not yet decided which model, i.e. KV or HC, to use for the virtual environment. Referring to hybrid object modeling approach defined by (7), we can conclude that when encountering a soft environment, the system needs a particular amount of time to change the default model of virtual environment (from KV to HC) on master side. Considering the time delays in the communication channels, we can say that the system carries out this task in approximately $\tau_m + \tau_s + t_{th}$ seconds. This effect is inevitable during the first interaction with a completely unknown environment. However, in subsequent contacts, this phenomenon can be avoided by reproducing the first contact’s data if the environment is known to remain unchanged during the operation.

Another issue which may arise when utilizing the proposed full decoupling method is the time when the operator is required to wait until the system becomes stable during the transition state. This waiting duration might be conceived of as a distracting factor by the user. This interval,
however, could be adjusted based on the desired application by suitably tuning the sliding mode controller parameters, although it cannot be completely omitted.

6. Conclusions and future work

In this paper, the idea of decoupling master and slave sides in a teleoperation system during a new contact with an environment for decreasing the model jump effect was addressed. A novel position control approach based on sliding mode control was proposed for transition state, while the master and slave sides were fully decoupled. The principal advantage of the proposed approach in comparison with previous works is its independence from environment model. This means that regardless of environment type and system dynamics, the proposed control approach leads the system towards stability when it first encounters a new unknown environment, while high transparency is preserved.

In addition, this approach is capable of dealing with any finite time delay in communication channels. Thus, with suitable tuning of different parameters of controllers, the stability of teleoperation system is guaranteed in the presence of large amounts of time delay.

Despite guaranteed stability and high transparency, the waiting time interval during transition state, in which the operator must stop the operation until the system reaches stability, can be an inconvenience for the user. Nevertheless, suitably tuning the relevant sliding mode controller parameters can partly compensate for this issue. Interaction with an unknown soft environment for the first time may also provide an inconvenient condition for the user, as thoroughly discussed in section 5. These two sources of inconvenience are to be further investigated through experimental tests in future work. How and how much these two sources can affect the haptic feel of the user and how to mitigate them, if necessary, can be discussed.

Moreover, this paper used predictive control approach merely for slave-to-master direction. In order to further augment the transparency and stability of the system, master-to-slave state prediction can also be taken into account in the presence of environment force in future works.

Appendix

The procedure of determining the control input of the sliding mode control for slave side in steady state is explained here. First, the relation (4) is rewritten in the following form

\[ \ddot{x}_s = f(x_s, t) + g(x_s, t)U_{s,st} \quad (A.1) \]

where \( x_s = [x_s \quad \dot{x}_s]^T \) is the state vector. Also \( f \triangleq f(x_s, t) \) and \( g \triangleq g(x_s, t) \) can be obtained as

\[ f = \frac{1}{m_s} (-b_s \dot{x}_s - k_s x_s - F_e) \quad (A.2) \]
\[ g = \frac{1}{m_s} \]  \hspace{1cm} (A.3)

Here \( f \) and \( g \) can have uncertainties, but in known ranges which are defined as follows

\[ |\hat{f} - f| \leq F \]  \hspace{1cm} (A.4)

\[ \frac{1}{\alpha} \leq \frac{g}{\hat{g}} \leq \alpha \]  \hspace{1cm} (A.5)

with \( \hat{f} \) and \( \hat{g} \) representing the estimated values of \( f \) and \( g \), respectively. In (A.5) \( \hat{g} \) and \( \alpha \) are defined according to the following relations

\[ \hat{g} \triangleq \sqrt{g_{\text{max}} g_{\text{min}}} \]  \hspace{1cm} (A.6)

\[ \alpha \triangleq \sqrt{\frac{g_{\text{max}}}{g_{\text{min}}}} \]  \hspace{1cm} (A.7)

where subscripts max and min denote the maximum and minimum values of the corresponding parameter, respectively. Next, by having the sliding surface as defined in (23) and applying the condition \( s_{st} = \dot{s}_{st} = 0 \), we can find the estimated control input as follows

\[ \hat{U}_{s,st} = \frac{1}{\hat{g}} \hat{u} \]  \hspace{1cm} (A.8)

where

\[ \hat{u} \triangleq -\hat{f} + \ddot{x}_m - \lambda_s \dot{s}_s \]  \hspace{1cm} (A.9)

Finally, the sliding condition must be applied according to (A.10)

\[ s_{st} \dot{s}_{st} \leq \eta_{st} |s_{st}| \]  \hspace{1cm} (A.10)

where \( \eta_{st} \) is strictly positive. It can be shown that a control input in the following form satisfies the condition (A.10)
\[ U_{s,st} = \hat{U}_{s,st} - \frac{1}{\bar{g}} k_{\text{gain}_{st}} \text{sat} \left( \frac{s_{st}}{\varphi_{st}} \right) \]  

(A.11)

In (A.11) \( k_{\text{gain}_{st}} \) is defined such that

\[ k_{\text{gain}_{st}} \geq k_{\text{gain}_{min}} \]

(A.12)

\[ k_{\text{gain}_{min}} \triangleq \alpha(F + \eta_{st}) + (\alpha - 1) |\dot{\hat{\theta}}| \]

(A.13)

Other parameters in (A.11) have been defined in section 4.1. Ultimately, by rewriting (A.11) according to (A.2), (A.3), (A.8) and (A.9) we can obtain the control input which is represented by (24) in section 4.1.

Notice that the control inputs for transition state (represented by (31) and (32) in section 4.3) can also be achieved through the aforementioned process. The only alterations are that the subscript \( st \) should be replaced by \( tr \) (denoting the transition state) and also the subscript \( s \) should be replaced by \( m \) for obtaining \( U_{m,tr} \).

References


Borhan Beigzadeh, received his PhD degree in mechanical engineering from Sharif University of Technology in 2011, working on dynamic walking systems and studying its correlation to dynamic passive/active manipulation systems. He then joined Iran University of Science and Technology (IUST) in Tehran where he established Biomechatronics and Cognitive Engineering Research Laboratory, and since then is an assistant professor in School of Mechanical Engineering. His research interests cover non-linear dynamics and control, robotics, biomechatronics, and cognitive engineering.

Behnam Yazdankhoo received the BSc degree in Mechanical Engineering from Amirkabir University of Technology, Tehran, Iran, in 2015. He is currently MSc student in the School of Mechanical Engineering at Iran University of Science and Technology, Tehran, Iran. His main research interests are design and control of haptic teleoperation systems and humanoid robots.

List of captions

Figure 1. General scheme of model-mediated teleoperation approach. x, $F_e$ and $\hat{F}_{e,m}$ represent position, environment force and estimated environment force, respectively. $\hat{\theta}_s$ is also the estimated environment parameters. Subscripts m and s denote master and slave, respectively.

Figure 2. A scheme of the communication channel in the proposed teleoperation system. X and Y represent two arbitrary parameters.

Figure 3. General scheme of the decoupling method. The Coupling/Decoupling blocks choose the input signals to be presented to the controller. Signals passing through the communication channels are only used when the coupling mode is active. Thus, if the decoupling mode is activated, the two control loops are closed separately and locally. x, $U$, $F_e$, $F_h$ and $\hat{\theta}_s$ represent
position, control signal, environment force, human force and estimated environment parameters, respectively. Subscripts m and s denote master and slave, respectively.

**Figure 4.** Environment and master reaction forces in interaction with the hard environment. The black dashed lines represent the start and end of the transition state.

**Figure 5.** Master and slave positions in interaction with the hard environment. The black dashed line (vertical) represents the end of the transition state (the transient state starts when the slave reaches the environment for the first time). The green dashed line (horizontal) shows the position of the environment on slave side.

**Figure 6.** Position error in interaction with the hard environment. The black dashed lines represent the start and end of the transition state. Although otherwise defined in section 4.3, the error depicted for the transition state is based on the definition in section 4.1.

**Figure 7.** Slave control input in interaction with the hard environment.

**Figure 8.** Environment and master reaction forces in interaction with the soft environment. The black dashed lines represent the start and end of the transition state.

**Figure 9.** Master and slave positions in interaction with the soft environment. The black dashed line (vertical) represents the end of the transition state (the transient state starts when the slave reaches the environment for the first time). The green dashed line (horizontal) shows the position of the environment on slave side.

**Figure 10.** Position error in interaction with the soft environment. The black dashed lines represent the start and end of the transition state. Although otherwise defined in section 4.3, the error depicted for the transition state is based on the definition in section 4.1.

**Figure 11.** Slave control input in interaction with the soft environment.

**Table 1.** Dynamical properties of the master and slave robots (Phantom Omni for master and Novint Falcon for slave) [24]

**Table 2.** Dynamical properties of the human hand [14]

**Table 3.** Parameters of sliding mode controller for steady state

**Table 4.** Parameters of sliding mode controllers for transition state

**Table 5.** Parameters of the hard environment, the corresponding SPRLS algorithm and the hybrid object modeling approach

**Table 6.** Parameters of the soft environment, the corresponding SPRLS algorithm and the hybrid object modeling approach
Figures and Tables

Figure 1

![Diagram of communication channels between Operator & Master, Virtual Environment, Environment & Slave, and Estimation]

Figure 2

![Diagram of communication channel between Master Side, Communication Channel, and Slave Side, with variables X and X^d]

Figure 3

![Diagram of communication channels between Master Robot, Controller, Coupling / Decoupling, Controller, Slave Robot, and Estimation, with variables F_h, U_m, F_e, x_m, x_m, x_s, x_s, and theta]

21
**Figure 4** (*Color reproduction on the Web*)

![Figure 4 Diagram](image)

**Figure 5** (*Color reproduction on the Web*)

![Figure 5 Diagram](image)

**Figure 6** (*Color reproduction on the Web*)

![Figure 6 Diagram](image)
Figure 7

Figure 8 (Color reproduction on the Web)

Figure 9 (Color reproduction on the Web)
Figure 10 *(Color reproduction on the Web)*

![Graph of Position Error vs Time](image)

Figure 11

![Graph of Slave Control Input vs Time](image)

**Table 1**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_m$</td>
<td>0.223</td>
<td>kg</td>
</tr>
<tr>
<td>$b_m$</td>
<td>17.227</td>
<td>Ns/m</td>
</tr>
<tr>
<td>$k_m$</td>
<td>6.286</td>
<td>N/m</td>
</tr>
<tr>
<td>$M$</td>
<td>0.223</td>
<td>kg</td>
</tr>
<tr>
<td>$B$</td>
<td>20.827</td>
<td>Ns/m</td>
</tr>
<tr>
<td>Parameter</td>
<td>Value</td>
<td>Unit</td>
</tr>
<tr>
<td>-----------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>$m_s$</td>
<td>1.158</td>
<td>kg</td>
</tr>
<tr>
<td>$b_s$</td>
<td>115.40</td>
<td>Ns/m</td>
</tr>
<tr>
<td>$k_s$</td>
<td>31.46</td>
<td>N/m</td>
</tr>
</tbody>
</table>

Table 2

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b_a$</td>
<td>3.6</td>
<td>Ns/m</td>
</tr>
<tr>
<td>$k_a$</td>
<td>40</td>
<td>N/m</td>
</tr>
</tbody>
</table>

Table 3

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\eta_{st}$</td>
<td>15</td>
</tr>
<tr>
<td>$\lambda_{st}$</td>
<td>1</td>
</tr>
<tr>
<td>$\phi_{st}$</td>
<td>0.1</td>
</tr>
<tr>
<td>$K_{gain_{st}}$</td>
<td>$1.1K_{gain_{min_{st}}}$</td>
</tr>
</tbody>
</table>

Table 4

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\eta_{s,tr}$</td>
<td>15</td>
</tr>
<tr>
<td>$\lambda_{s,tr}$</td>
<td>1</td>
</tr>
<tr>
<td>$\phi_{s,tr}$</td>
<td>0.3</td>
</tr>
<tr>
<td>$K_{gain_{s,tr}}$</td>
<td>$1.1K_{gain_{min_{s,tr}}}$</td>
</tr>
<tr>
<td>$\eta_{m,tr}$</td>
<td>15</td>
</tr>
<tr>
<td>$\lambda_{m,tr}$</td>
<td>1</td>
</tr>
<tr>
<td>$\phi_{m,tr}$</td>
<td>0.3</td>
</tr>
<tr>
<td>$K_{gain_{m,tr}}$</td>
<td>$1.1K_{gain_{min_{m,tr}}}$</td>
</tr>
</tbody>
</table>
Table 5

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_{KV}$</td>
<td>6000</td>
<td>N/m</td>
</tr>
<tr>
<td>$B_{KV}$</td>
<td>230</td>
<td>Ns/m</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>0.1</td>
<td>-</td>
</tr>
<tr>
<td>$\beta$</td>
<td>30000</td>
<td>-</td>
</tr>
<tr>
<td>$\hat{\theta}_0$</td>
<td>$[3000 ; 100]^T$</td>
<td>-</td>
</tr>
<tr>
<td>$P_0$</td>
<td>$I_2^a$</td>
<td>-</td>
</tr>
<tr>
<td>$e_0^{\square}$</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>SPRLS sample time</td>
<td>0.0005</td>
<td>s</td>
</tr>
<tr>
<td>$K_{th}$</td>
<td>2500</td>
<td>N/m</td>
</tr>
<tr>
<td>$t_{th}$</td>
<td>0.1</td>
<td>s</td>
</tr>
</tbody>
</table>

$^a$ I is the identity matrix

Table 6

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_{HC}$</td>
<td>1000</td>
<td>N/m</td>
</tr>
<tr>
<td>$B_{HC}$</td>
<td>30</td>
<td>Ns/m</td>
</tr>
<tr>
<td>$n$</td>
<td>1.5</td>
<td>-</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>19000</td>
<td>-</td>
</tr>
<tr>
<td>$\beta$</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>$\hat{\theta}_0$</td>
<td>$[400 ; 400 ; 13 ; 13]^T$</td>
<td>-</td>
</tr>
<tr>
<td>$P_0$</td>
<td>$I_4^a$</td>
<td>-</td>
</tr>
<tr>
<td>$e_0^{\square}$</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>SPRLS sample time</td>
<td>0.0005</td>
<td>s</td>
</tr>
<tr>
<td>$K_{th}$</td>
<td>2500</td>
<td>N/m</td>
</tr>
<tr>
<td>$t_{th}$</td>
<td>0.1</td>
<td>s</td>
</tr>
</tbody>
</table>

$^a$ I is the identity matrix