Model-mediated Teleoperation for Movable Objects: Dynamics Modeling and Packet Rate Reduction

Xiao Xu[1]
Email: xiao.xu@tum.de

Sili Chen[1]
Email: sili.chen@tum.de

Eckehard Steinbach[1]
Email: eckehard.steinbach@tum.de

***Abstract*—We study Model-Mediated Teleoperation (MMT) for objects which can be moved and rotated on a planar surface. In MMT, a simple object model is employed on the master side to approximate the remote environment. The haptic feedback is rendered locally on the master side based on the model whose parameters are estimated on the slave side. Since the haptic control loop is running locally without noticeable delay, system stability can be guaranteed in the presence of arbitrary communication delays. The main challenges for MMT are environment modeling and online parameter identification. This work has two contributions. Firstly, we describe how to model movable objects with approximately known model geometry for MMT systems. The object dynamics including environment damping as well as the friction are modeled in real time during teleoperation. Secondly, we propose a hybrid deadband-based and time-triggered updating scheme to adaptively reduce the required packet rate while transmitting the estimated model parameters from the slave to the master. According to our experiments, the movable object can be accurately modeled in real time. In addition, compared to the conventional deadband-based updating scheme, our proposed hybrid updating scheme shows improvements in both model accuracy and packet rate reduction.**

I. Introduction

Bilateral teleoperation systems with haptic feedback allow human users to interact with and to immerse themselves in a remote environment by means of slave and master devices which exchange force and position/velocity information over a communication link [1]. As illustrated in Fig. 1(a), the slave robot follows the received position or velocity commands sent by the master. During the slave's interaction with the remote environment, the haptic, visual, and audio signals captured by the sensors on the slave side are transmitted back to the master, and displayed to the operator. The haptic feedback, in addition to the visual and audio information, increases the sense of being present in the remote environment thereby allowing a human operator to perform dangerous tasks at a safe distance such as handling nuclear/toxic/explosive materials, and improving the user's ability to perform complex tasks such as telesurgery, or tele-teaching/tele-training, etc. [2], [3].

The presence of communication delay in the haptic channel introduced by the network, however, degrades both the system stability and transparency [4]. The use of passivity-based control schemes, such as the time-domain passivity approach [7] and wave-variable transformation [5], [6], guarantees the system stability at the cost of degraded transparency.

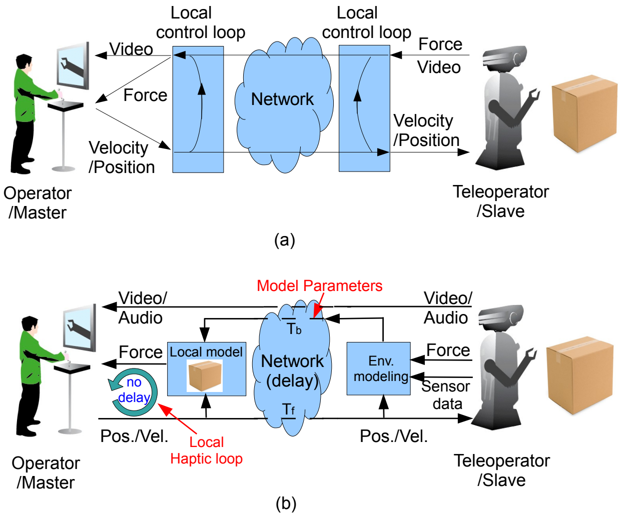

Fig. 1. (a) Classical teleoperation system with direct force feedback. (b) An overview of MMT architecture.

Different from the passivity-based teleoperation, the concept of Model-Mediated Teleoperation (MMT) has been proposed to address both stability and transparency issues in the presence of communication delays [8], [9], [10]. In the MMT approach, a virtual object model is employed to approximate the remote environment. Based on the slave's motion, force signals, and other sensor data, the parameters describing this virtual object model (model parameters) are continuously estimated in real time on the slave side, and transmitted back to the master whenever the slave obtains a new model. Larger update rates for the estimated parameters lead to higher model accuracy on the master side. On the master side, a local model is constructed based on the received model parameters, and the haptic feedback is generated based on the local model without noticeable delay (see Fig. 1(b)). If the estimated model parameters lead to an accurate approximation of the remote environment, both stable and transparent teleoperation can be achieved.

Online parameter estimation for environment modeling is the most important and challenging task in MMT, since a perfect match between the local model and the environment enables stable teleoperation in the presence of arbitrary communication delays. Model updating is required when the models of the master and the slave environment are mismatched, which

This work has been supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC Grant agreement no. 258941.

[1] The authors are with the Chair of Media Technology, Technische Universität München, Arcisstrasse. 21, 80333 München, Germany

happens frequently due to environment changes, operator commands and movements, improper model approximation, or inefficient parameter estimation methods. Considering objects motion constraints, the slave environment can be divided into two categories: 1) non-movable objects, including rigid objects and deformable objects. Fixed-axis rotation or constrained deformation of the objects is allowed, but the objects are not freely movable in the workspace. 2) Movable objects. As the name suggests, objects can be freely moved in the workspace.

For modeling non-movable objects, the key point is to estimate the object dynamics at the contact area. For modeling movable objects, all of the objects' dynamics including their interaction with surroundings like the ground friction, environment damping, etc., need to be estimated. When modeling movable objects, the system needs to deal with more parameters compared to modeling non-movable objects. This is why previous studies on MMT mainly focus on non-movable objects [11], [12], [13]. For dealing with movable objects, Passenberg has proposed a real-time algorithm for estimating object inertia for grasp tasks in [14]. To the best of our knowledge, a comprehensive study on MMT with freely movable objects is not available from the literature.

Motivated by this, we study the modeling of movable objects for MMT systems. However, it is difficult to find a simple model which can describe a real movable object with complicated and irregular exterior. As the first step, we investigate modeling regular objects whose geometry can be roughly described by a cubic object. The cubic object is considered as the bounding box of the real object. Cube-like objects, i.e. bricks, container boxes, part blanks, exist widely and play an important role in teleoperation tasks for industrial parts machining [15], machine-assisted obstacle avoiding [16], obstacle removing for environment exploration [3], [16], etc.

This work has two contributions. Firstly, we present our modeling approach for movable objects with constrained motion for MMT systems. The approximative geometry of the object is assumed to be known. The object is modeled as a three degrees-of-freedom (DoF) mass cube which has two DoFs in translation and one DoF in rotation. Secondly, we propose a hybrid deadband-based and time-triggered updating scheme to adaptively reduce the packet rate, while guaranteeing a sufficiently high model accuracy on the master side. According to our experiments, the hybrid updating scheme shows better performance compared to the state-of-the-art deadband-based updating approach [18].

The remainder of this paper is organized as follows. In Sec. II, we present the method for modeling movable objects in 3D space. In Sec. III, the hybrid deadband-based and time-triggered updating scheme is proposed. Experimental results are discussed in Sec. IV. Sec. V concludes this paper.

II. Dynamics Modeling for Movable Objects

We investigate a movable rigid cube in 3D space. The approximative object geometry (the size of the cube) is assumed to be known (e.g. measured by a visual sensor or pre-scanning procedure). This cube has three degrees of freedom (DoFs), including two DoFs in translation in the $x-y$ plane and one DoF in rotation along the z axis (see Fig. 2). The friction between the slave end-effector and the object is assumed to be negligible (smooth tool tip), while the friction between the object and the ground is not negligible. In this section, we discuss the proposed method for modeling a movable cubic object.

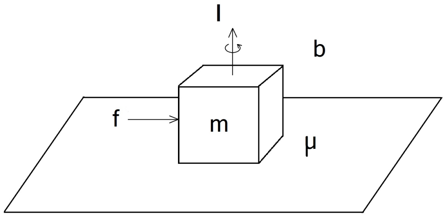

Fig. 2. The used dynamic model for approximating a movable object. f is the action force, m and I denote the object mass and rotation inertia, b represents the environment damping, and μ is the friction coefficient between the object and the ground.

A. Translation

1) Slave side: parameter estimation

Eq. (1) is the employed force model for approximating the cube translation.

$$\mathbf{f} = m \cdot \mathbf{a} + b \cdot \mathbf{v} + c \cdot sign(\mathbf{v}), \qquad c = \mu \cdot m \cdot g \quad (1)$$

where $\mathbf{a}$ and $\mathbf{v}$ are the object linear acceleration and velocity, respectively. $\mathbf{f}$ is the measured slave contact force, m is the object mass, b represents the environment damping, c denotes the Coulomb friction between the cube and the ground, μ is the dynamic friction coefficient, and g is the acceleration of gravity. We assume that the maximum static friction is identical to the dynamic friction.

Replacing m, b, c by their estimates $\hat{m}, \hat{b}, \hat{c}$, the computed force $\hat{\mathbf{f}}$ from this object model can be described as

$$\hat{\mathbf{f}} = \hat{m} \cdot \mathbf{a} + \hat{b} \cdot \mathbf{v} + \hat{c} \cdot sign(\mathbf{v}) \quad (2)$$

Thus, a linear optimization problem is established which minimizes the force error between the measured slave contact force and the computed model force:

$$\{\hat{m}, \hat{b}, \hat{c}\} = arg \min_{\{\hat{m}, \hat{b}, \hat{c}\}} ||\mathbf{f} - \hat{\mathbf{f}}||^2 \quad (3)$$

The block least squares algorithm is employed to solve this problem. Note that it is difficult to directly measure the exact object velocity and acceleration. However, by assuming no relative motion between the slave end-effector and the object surface after a stable contact, the object velocity and acceleration signals are identical to the slave velocity and acceleration signals. Thus, we can use the slave motion signals as the object motion signals for addressing this linear optimization problem. In addition, a Kalman filter is applied for smoothing the measured slave velocity signals.

2) Master side: model updating and haptic rendering

The estimated parameters $(\hat{m}, \hat{b}, \hat{c})$ are transmitted back to the master to update the local model. In order to avoid sudden changes in the model parameters on the master side, an exponential moving average filter is applied to smooth the updating process.

$$param_{applied} = param_{pre} \cdot \lambda + param_{recv} \cdot (1 - \lambda) \quad (4)$$

where $param_{applied}$ is the smoothed model parameter value which is applied to update the local model. $param_{pre}$ is the previously applied model parameters. $param_{recv}$ denotes the currently received parameters. The stronger the low-pass characteristic of this filter, defined by the parameter λ, the slower the master local model adapts to the new parameters. In our experiments (see Sec. IV), λ is set to be 0.5.

The motion of the local model on the master side is computed based on the model parameters and the master force:

$$\begin{aligned} \mathbf{a_m^n} &= (\mathbf{f_m^n} - \hat{b} \cdot \mathbf{v_m^{n-1}} - \hat{c} \cdot sign(\mathbf{v_m^{n-1}}))/\hat{m} \\ \mathbf{v_m^n} &= \mathbf{v_m^{n-1}} + \mathbf{a_m^n} \cdot T \\ \mathbf{p_m^n} &= \mathbf{p_m^{n-1}} + \mathbf{v_m^n} \cdot T \end{aligned} \quad (5)$$

where T denotes the sampling period of the discrete time simulation. $\mathbf{f_m^n}$ is the master force which is displayed to the human user at the n^{th} sampling time. $\mathbf{a_m^n}$, $\mathbf{v_m^n}$ and $\mathbf{p_m^n}$ represent the acceleration, velocity, and position of the local model at the n^{th} sampling time, respectively.

The master force is generated based on the proxy-HIP (haptic interacting point) algorithm. The proxy stays always on the object surface ($\mathbf{p_{proxy}^n} = \mathbf{p_m^n}$), while the HIP can penetrate into the object ($\mathbf{p_{HIP}^n}$). The master force can be computed as:

$$\mathbf{f_m^n} = k_p \cdot (\mathbf{p_{HIP}^n} - \mathbf{p_{proxy}^n}) \quad (6)$$

where k_p is the virtual stiffness between the proxy and the HIP. Since we assume the object is rigid, k_p is set to be the maximum stiffness value of the master device in our experiments.

Note that we assume a non-frictional contact between the slave end-effector and the object surface. Thus, the computed model force on the master side is always perpendicular to the surface of the local model. It should be emphasized that the friction between the object and the environment is not negligible, and included in our model.

B. Rotation

The object in the remote environment is rotatable as shown in Fig. 2. Although this rotation could be small, we still need to model relevant rotation parameters, in order to approximate the remote environment as accurately as possible. The employed dynamic model for object rotation is shown in (7).

$$M = I \cdot \dot{\omega} + e \cdot sign(\omega) \quad (7)$$

where M is the applied torque on the object, I is the object rotation inertia, e is the torque produced by the ground friction, ω is the angular velocity. The damping term is ignored, as we assume that the object rotation is not fast.

1) Slave side: parameter estimation

We need to estimate the rotation inertia I and the friction torque e. If there is a torque sensor, rotation parameters can be easily estimated similar to the estimation of the translation parameters. However, if the torque measurement is missing, the inertial and friction torque can be only analytically estimated with the help of known geometry and translation parameters.

With a known cube side length l and the estimated mass $\hat{m}$, the rotation inertia is

$$\hat{I} = \hat{m} \cdot l^2/6 \quad (8)$$

The friction torque e can be estimated by assuming a uniform friction action between the object bottom surface and the ground:

$$\hat{e} = \int_{-l/2}^{l/2} \int_{-l/2}^{l/2} \frac{\hat{c}}{l^2} \cdot \sqrt{x^2 + y^2} dxdy \quad (9)$$

2) Master side: dynamics rendering

For a given master force $\mathbf{f_m^n}$, the torque acting on the local model can be computed as

$$M^n = \mathbf{d_m^n} \times \mathbf{f_m^n} \quad (10)$$

where $\mathbf{d_m^n}$ is the displacement vector from the object geometry center (rotation axis) to the contact point between the master proxy and the surface of the local model.

Similar to the translational motion, the rotation of the local model can be computed as

$$\begin{aligned} \dot{\omega}^n &= (M^n - \hat{e} \cdot sign(\omega^{n-1}))/\hat{I} \\ \omega^n &= \omega^{n-1} + \dot{\omega}^n \cdot T \\ \theta^n &= \theta^{n-1} + \omega^n \cdot T \end{aligned} \quad (11)$$

where θ^n is the angular position of the local model at the n^{th} sampling time.

III. Model Revision

A. Model mismatch

In practice, we cannot always perfectly estimate a complex environment model in real time with limited sampling data. This imperfect estimation leads to model mismatch between the master local model and the real slave environment. Model mismatch occurs normally due to using an improper model to approximate the slave environment, large measurement noise/error during the parameter estimation, the change of the slave environment, improper assumptions/simplification during the parameter estimation, etc.

For our cubic object model, the potential model mismatch cases can be summarized as follows.

- Geometry mismatch case 1: The master HIP is in contact with the local object model, while the slave end-effector is still in free space. In this case, the local model needs to be moved away from the master HIP in order to release the contact on the master side.
- Geometry mismatch case 2: The master HIP is in free space, while the slave end-effector is in contact with the real object in the remote environment. In this case, the position as well as the rotation of the local model needs to be updated based on the object geometry currently detected. The local model is set to be at the position where the slave end-effect contacts with the real object. The object rotation can be computed based on the direction of the measured slave contact force. Since we assume a non-frictional contact between the slave end-effector and the object, the force direction is considered to be perpendicular to the object surface.
- Impedance and geometry mismatch: Both the master HIP and slave end-effector are in contact with the local model and the real object, respectively. However,

the measured slave contact force is different from the computed model force on the master side, either in magnitude or direction, or both. In this case, the recently estimated model parameters are transmitted back to the master to update the local model.

B. Data reduction

For the MMT approach, model mismatch cannot be avoided, but the model accuracy can be improved. Obviously, continuously updating the estimated model parameters at a high rate (i.e. $> 100Hz$) can improve the model accuracy. However, this leads to high packet rate over the network. Thus, we need to find a proper data reduction scheme which is able to significantly reduce the required packet rate while guaranteeing sufficiently high model accuracy.

In previous studies, the fixed deadband (DB) approach [17] and the perception-based deadband coding scheme [18], [19] show their excellent performance in packet rate reduction for haptic communication. The key idea of the DB approach is that the packet is transmitted only when the difference between the current signal and the most recently sent signal is larger than a predefined threshold. This threshold is called deadband parameter. The DB approach can also be used for reducing the packet rate in our MMT systems. For example, if the system detects a significant model mismatch between the master local model and the slave environment, a parameter update is triggered. Otherwise, no transmission is needed and the parameters of the local model remain the same as the previously received ones.

There are several metrics to define *model mismatch*, i.e. significant difference between the previously used model impedance parameters and the ones currently estimated, or a large position / rotation error between the local model and the real object, or a large difference between the measured slave contact force and the on-master-side rendered model force, etc. In our work, we use the difference between the slave contact force and the computed model force as the metrics for model mismatch. The mechanism to trigger updates is similar to the multi-DoF DB approach described in [18], and illustrated in Fig. 3. For example, if the relative force difference is larger than the deadband parameter, the local model is considered mismatched with the remote environment, and thus we need to update the model parameters (see (12)).

$$\text{update trigger} = \begin{cases} yes, & \text{if } ||\mathbf{f_s} - \mathbf{f_m}||/||\mathbf{f_s}|| > \Delta \\ no, & \text{else} \end{cases} \tag{12}$$

where $\mathbf{f_s}$ and $\mathbf{f_m}$ denote the measured slave contact force and the on-master-side computed model force. Δ is the deadband parameter.

In order to further improve the model accuracy, we modify the DB approach by fusing it with a time-triggered updating scheme. This means that the model parameters are not only updated when a significant model mismatch is detected, but also updated every T_{update}, if no update occurs during this time period. Here, T_{update} is a predefined time period for the time-triggered updating. The hybrid updating scheme proposed in this work can be summarized as follows:

1) If a significant model mismatch is detected, then update the model parameters and set $Timer$ to 0. Else, $Timer = Timer + 1$ and go to 2).

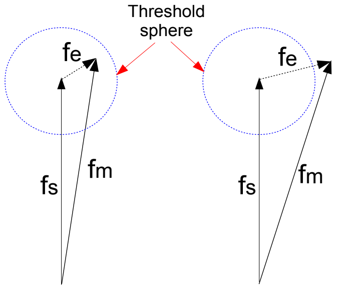

Fig. 3. An isotropic multi-DoF deadband (dead zone) approach. Left: Dead zone compliance. Right: Dead zone violation. $\mathbf{f_s}$ and $\mathbf{f_m}$ denote the measured slave contact force and the computed model force. $\mathbf{f_e}$ is the force error.

2) If $Timer > T_{update}$, then update the model parameters and set $Timer$ to 0. Else, $Timer = Timer + 1$.

Compared to the DB approach, we infer that the proposed hybrid updating scheme can improve both the packet rate reduction and the model accuracy. Since every additional update due to the time-triggered scheme decreases the magnitude of the model mismatch, the required number of packet transmission can be reduced. However, T_{update} needs to be carefully defined. If T_{update} is too small, it leads to significantly increased packet rate. If T_{update} is too large, the efficiency of using the hybrid updating scheme is quite low. In Sec. IV, we experimentally compare the performance of both the DB approach and the hybrid updating scheme.

IV. Experiments

A. Setup

A series of experiments are conducted using a virtual environment (VE). The software is developed based on the CHAI3D library (www.chai3d.org). A Novint Falcon haptic device is used as the master. The VE, including all the necessary computation for environment modeling, is simulated at a laptop with a Intel i7 processor and a 8 GB memory. Since the aim of the experiments is to verify the feasibility of the proposed environment modeling method and the hybrid data reduction scheme, communication delay over the network is set to be negligible.

The slave environment is shown in Fig. 4. A magenta cube is placed on a flat ground as the real object in the slave environment. This cube is freely movable in the xy-plane, and is rotatable along the z-axis. The slave proxy represents the virtual slave end-effector, and is controlled by the position of the master device. The transparent cube overlayed on the real object represents the estimated environment model, whose parameters (both geometry and impedance) are transmitted back to the master for local haptic rendering. During the experiments, the virtual slave is controlled to push the real object (magenta cube) along the blue trajectory. The local model (transparent cube), which is used for the haptic rendering on the master side, is not visually displayed to the human operator.

The side length of the real cube is 36 cm, weight 0.7 kg. Assuming the geometry parameters of the real cube can be measured using a 3D vision sensor with a small measurement error, we set the side length of the local model to be 40 cm. Note that there is a scaling factor of 13 for the workspace when

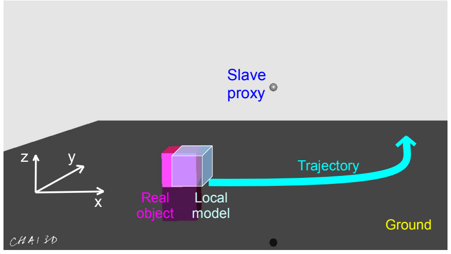

Fig. 4. Slave environment.

the slave environment is displayed to the user. The dynamic motion of the cube in the slave environment (VE) is computed by the CHAI3D library with a force sampling rate of 1 kHz. The dynamics of the local model are estimated based on the method proposed in Sec. II. The computational duration for model estimation is less than 1 ms, which allows a maximally 1 kHz update rate. During the experiments, we record the slave contact force, the computed model force, and the required average packet rate based on 10 times repetition.

B. Results

We compare the modeling accuracy and the required average packet rate for the two parameter updating approaches: the DB-based updating scheme and the hybrid updating scheme. For the DB-based updating scheme, we process seven different deadband parameters (thresholds for the relative force error): 2.5%, 5%, 10%, 15%, 20%, 25%, and 30%. For the hybrid updating scheme, besides the seven deadband parameters, we have tested two different time periods T_{update} for parameter updating: 50 ms and 150 ms.

In order to evaluate the modeling accuracy, we calculate the perceptual mean square error (PMSE) of the force signals as proposed in [20]. The measured slave contact force is considered as the ground truth. The closer to the ground truth the computed model force is, the more accurate the local model. The force PMSE described in [20] is:

$$PMSE_x = \frac{c^2}{N}\sum_{i=1}^{N}(ln|\frac{F_s^x(i)}{F_m^x(i)}|)^2 \tag{13}$$

$$PMSE_y = \frac{c^2}{N}\sum_{i=1}^{N}(ln|\frac{F_s^y(i)}{F_m^y(i)}|)^2 \tag{14}$$

$$PMSE = PMSE_x + PMSE_y \tag{15}$$

where F_s^x and F_s^y are the measured slave contact forces in x and y directions. F_m^x and F_m^y denote the computed model forces in x and y directions. N is the number of samples. c is a constant and is set to be 1 in our experiments. The higher the PMSE value, the lower the model accuracy.

According to [20], the computation of the PMSE is activated when both the slave contact force and the computed model force are larger than a threshold F_{min}. This threshold is the absolute force threshold below which no stimulus can be perceived at all. According to [21], the average minimum perceivable force for arm motion is 0.04 N. Thus, F_{min} is set to be 0.04 N in our experiments.

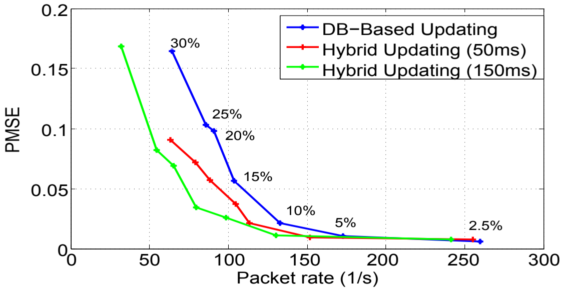

Fig. 5. Rate-PMSE performance of the three tested updating schemes.

The results of the experiments are shown in Fig. 5. The x-axis corresponds to the average packet rate (packet/s), while the y-axis shows the PMSE. In Fig. 5, the three curves represent the results of using the hybrid updating scheme with 50ms and 150ms updating period, and the DB-based updating scheme. The seven inflection points in each curve represent the different deadband parameters between 2.5% and 30%.

We observe that all three schemes show a similar rate-PMSE behavior. The PMSEs decrease steeply when the deadband parameter is reduced from 30% to 10%, and then tend to be flat for deadband parameters from 10% to 2.5%. This behavior reveals that a deadband parameter of about 10% is a good choice for balancing the packet rate and PMSE for the three updating schemes.

Comparing the three curves, the hybrid updating scheme with 150 ms updating period (150ms-scheme) shows the best performance, while the hybrid updating scheme with 50 ms updating period (50ms-scheme) takes the second place. Both of them outperform the DB-based updating scheme. Note that with the deadband parameter of 30%, the 50ms-scheme can only reach a minimum packet rate of about 60 packet/s, while the 150ms-scheme is able to reduce the packet rate to less than 40 packet/s. This is because when the 50ms-scheme is adopted, the system forces a transmission rate of at least 1000/50ms=20 packet/s. This is the lower bound of the packet rate while using the 50ms-scheme. However, if the 150ms-scheme is employed, the lower bound of the packet rate is $1000/150ms \approx 7$ packet/s. This means that for the same deadband parameter, decreasing the updating period T_{update} results in increased packet rate. On the other hand, the increased packet rate produces a relatively higher matching rate between the local model and the slave environment. Benefiting from this, the PMSE can be reduced.

Two individual experimental results of the force signals and the packet updating instants are shown in Fig. 6 and Fig. 7. Fig. 6 shows the records of using the DB-based updating scheme with the deadband parameter of 15%. Fig. 7 illustrates the records of using the hybrid updating scheme with the same deadband parameter and $T_{update} = 150ms$. The red solid lines, the green dotted lines and the blue solid lines represent the magnitude of the slave contact force, the magnitude of the computed model force, and the magnitude of the force error, respectively. The cyan lines represent the time instant when packets are transmitted.

For the commonalities, we observe that the packet rates in both figures are relatively high at the beginning, since the

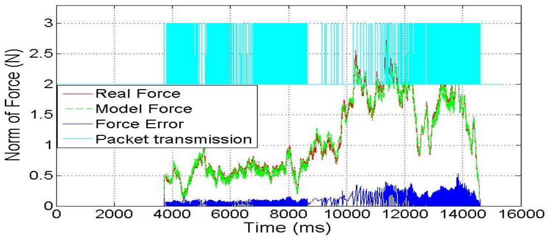

Fig. 6. The force signals with the packet udpating instants. Updating scheme: the DB-based updating. Deadband parameter: 15%

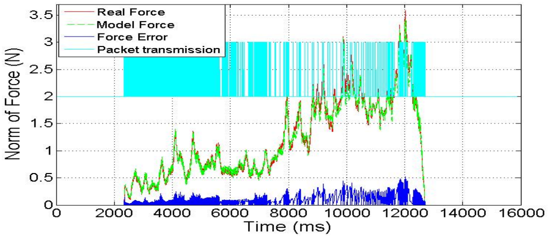

Fig. 7. The force signals with the packet udpating instants. Updating scheme: the hybrid updating. Deadband parameter: 15%, updating period: 150 ms.

estimated model parameters are not quite accurate. Thus, large model mismatch is detected and frequent model updating is required. About 4500 ms after the first contact, the packet rate is dramatically decreased, implying that the estimated model is comparable with the real object. In addition, a relatively higher packet rate can be found during the period of releasing. The rapid changes of the object position, velocity, and rotation lead to unreliable estimation and large model mismatch. This triggers frequent updating.

Comparing Fig. 6 with Fig. 7, the two methods show different packet transmission behaviors. In Fig. 6, there is no packet transmission from time 8600 ms to 9100 ms, and from 9900 ms to 10200 ms. During these time slots, the force error between the local model and the slave environment is smaller than 15%, requiring no updating. Although the packet rate is reduced, the potential model mismatch is accumulated during these time periods, which can be a reason of the continuously increasing packet rate from 10200 ms to 12500 ms. For Fig. 7, despite the model parameters are forced to be updated at least every 150 ms, the additional updates can improve the model accuracy and eliminate the accumulation of the modeling error. Therefore, the packet rate is more stable than that of the DB-based updating scheme. This behavior can be observed in Fig. 7 from about 7500 ms to 10500 ms.

V. Conclusion

In this work, we present model-mediated teleoperation for objects which can be moved and rotated on a planar surface. The object dynamics are modeled in real time during teleoperation, including object mass, environment damping as well as the friction between the object and the ground. In addition, we propose a hybrid updating scheme combining the deadband-based and the time-triggered updating schemes to adaptively reduce the required packet transmission for updating the estimated model parameters. Compared to the state-of-the-art deadband-based updating scheme, the proposed hybrid updating scheme shows improvements in both model accuracy and packet rate reduction. Future work will focus on looking for the optimal updating period T_{update} for the hybrid updating scheme. In addition, we will extend the proposed method for real teleoperation systems with communication delays.

References

[1] W. Ferrell and T. Sheridan. Supervisory control of remote manipulation. IEEE Spectrum, vol. 4, no. 10, pp. 81-88, 1967.

[2] E. Saddik. The potential of haptics technologies. IEEE Instrumentation Measurement Magazine, vol. 10, no. 1, pp. 10-17, April 2007.

[3] M. Buss and G. Schmidt, Control problems in multi-modal telepresence systems, Advances in Control: Highlights of the 5th European Control Conference, Karlsruhe, Germany, 1999.

[4] D. Lawrence. Stability and transparency in bilateral teleoperation. IEEE Trans. on Robotics and Automation, vol. 9, no. 5, pp. 624-637, 1993.

[5] R. Anderson, M.W. Spong. Bilateral Control of Teleoperators with Time Delay. IEEE Trans. on Automatic Control, vol. 34, no. 5, pp. 494-501, May 1989.

[6] G. Niemeyer and J.-J. Slotine. Stable Adaptive Teleoperation. IEEE Journal of Oceanic Engineering, vol. 16, no. 1, pp. 152-162, Jan. 1991.

[7] J. Ryu, J. Artigas and C. Preusche. A passive bilateral control scheme for a teleoperator with time-varying communication delay. Elsevier Journal of Mechatronics, vol. 20, no. 7, pp. 812-823, Oct. 2010.

[8] B. Hannaford. A design framework for teleoperators with kinesthetic feedback. IEEE Transactions on Robotics and Automation, vol. 5, no. 4, pp. 426-434, Aug. 1989.

[9] P. Mitra and G. Niemeyer. Model mediated telemanipulation. Int. J. of Robotics Research, vol. 27, no. 2, pp. 253-262, 2008.

[10] B. Willaert, J. Bohg, H. Brussel and G. Niemeyer. Towards multi-DOF model mediated teleoperation: using vision to augment feedback. IEEE International Workshop on HAVE, Munich, Germany, Oct. 2012.

[11] X. Xu, B. Cizmeci, A. Al-Nuaimi and E. Steinbach. Point Cloud-Based Model-Mediated Teleoperation With Dynamic and Perception-Based Model Updating. IEEE Transactions on Instrumentation and Measurement, May 2014.

[12] H. Li and A. Song. Virtual-Environment Modeling and Correction for Force-Reflecting Teleoperation With Time Delay. IEEE Transactions on Industrial Electronics, vol. 54, no. 2, pp. 1227-1233, 2007.

[13] A. Achhammer, C. Weber, A. Peer and M. Buss. Improvement of model-mediated teleoperation using a new hybrid environment estimation technique. In Proc. of ICRA2010, Anchorage, AK, May 2010.

[14] C. Passenberg. Transparency- and Performance-Oriented Control of Haptic Teleoperation Systems. Ph.D Thesis, Chair of Automatic Control Engineering, Technische Unversität München, 2012.

[15] J. Vertut, and P. Coiffet. Teleoperation and Robotics: Applications and Technology Hermes Publishing, 1st Edition, 1985.

[16] D. Aarno, S. Ekvall, and D. Kragic. Adaptive Virtual Fixtures for Machine-Assisted Teleoperation Tasks. Proceedings of ICRA 2005. April, 2005.

[17] P. Otanez, J. Moyne, and D. Tibury. Using deadbands to reduce communication in networked control systems. Proceedings of the American Control Conference, 2002.

[18] P. Hinterseer, S. Hirche, S. Chaudhuri, and E. Steinbach. Perception-Based Data Reduction and Transmission of Haptic Data in Telepresence and Teleaction Systems. IEEE Trans. on Signal Processing, vol. 56, no. 2, pp. 588-597, Feb. 2008.

[19] E. Steinbach, S. Hirche, J. Kammerl, I. Vittorias and R. Chaudhari. Haptic Data Compression and Communication. IEEE Signal Processing Magazine, vol. 28, no. 1, pp. 87-96, Jan. 2011.

[20] R. Chaudhari, E. Steinbach, and S. Hirche. Towards an objective quality evaluation framework for haptic data reduction. IEEE World Haptics Conference. Istanbul, June 2011.

[21] S. Feyzabadi, S. Straube, M. Folgheraiter, E. Kirchner, S. Kim, and J. Albiez. Human Force Discrimination during Active Arm Motion for Force Feedback Design. IEEE Trans. on Haptics, vol. 6, no. 3, pp. 309-319, July 2013.