Short Papers

Description of Instantaneous Restriction Space for Multi-DOFs Bilateral Teleoperation Systems Using Position Sensors in Unstructured Environments

Keehoon Kim, Wan Kyun Chung, and M. Cenk Çavuşoğlu

Abstract—**This paper investigates a novel position-sensor-based force reflection framework for multi-degree-of-freedom (DOF) bilateral teleoperation systems in unstructured environments. The conventional position-sensor-based force reflection method, which is known as position error feedback, may generate grossly inaccurate force reflection directions during collisions involving the slave manipulator links. The proposed restriction space projection framework calculates the instantaneous restriction space to provide the accurate force reflection, regardless of kinematic dissimilarity (KDS) conditions of bilateral teleoperation systems. Simulation results confirmed the validity of the proposed framework in a KDS bilateral teleoperation system under various constraint conditions.**

Index Terms—**Bilateral teleoperation system, haptic interface, instantaneous restriction space (IRS), obstacle avoidance, restriction space projection (RSP).**

I. Introduction

In a bilateral teleoperation system, force reflection plays an important role in the understanding of physical interaction status at a remote side. Since interaction forces are commonly detected by position control errors or force sensor signals, force reflection methods can be classified into position–position (p-p) [1]–[4], position–force (p-f) [1], [3], [5]–[12], force–position (f-p) [10], force–force (f-f) [13], and their combinations, such as general four-channel architectures, from a sensory configuration standpoint.

Although force sensors dramatically enhance the accuracy of the force reflection compared with the position-sensor-based force reflection, it is practically burdensome to distribute enough force sensors to detect every possible physical interaction. In addition, a bilateral teleoperation system using the force-sensor-based force reflection is confronted by a serious problem when the force sensor cannot detect the interaction force. For example, if a slave robot collides with unexpected obstacles that are not detected by force sensors, no force is reflected to a master device. As a human operator is able to move the master device without constraint, the system becomes unstable due to the significant position difference between the master device and the slave robot. This situation can happen frequently when the operation environment is unstructured, cluttered, or moving environments such as underwater manipulations, prosthetic arm manipulations, or minimally invasive surgical robot operations.

On the other hand, since position sensors are typically placed on every joint of a bilateral teleoperation system, the position-sensor-based force reflection method is able to enhance the robustness against unexpected obstacle collisions, especially against collisions that do not occur at the end-effector, but at the links of the slave manipulator. In the present study, the conventional position-sensor-based force reflection method, which is known as position error feedback method, is shown, which is not able to adequately describe the direction of the force reflection, especially when the bilateral teleoperation system is multi-degrees-of-freedom (DOFs), as will be explained in detail in Section II. Therefore, an enhanced position-sensor-based force reflection framework has been proposed for multi-DOFs bilateral teleoperation systems that satisfy the following conditions.

1) The force reflection framework should be applicable to multi-DOFs bilateral teleoperation systems for all types of kinematic dissimilarity (KDS) conditions shown in Table I.[1]
2) The force reflection framework should deal with the joint constraints, as well as the endpoint constraint.
3) The force reflection framework should be applicable in unstructured environment without *a priori* knowledge of the obstacles or the environment.
4) The force reflection framework should be able to use force sensor signals when obstacles are detectable by force sensor.

Table I classifies bilateral teleoperation systems according to the KDS conditions. The conventional position error feedback method is applicable to kinematically similar (KS) type, in which one-to-one joint angle control is available. For kinematically dissimilar (KDS) teleoperation systems, the conventional position error feedback is applicable only if the two manipulators have equal DOFs, and a tasks space controller with the same gain is used in all directions. This limitation will be explained in depth in Section II.

Manuscript received August 30, 2008; revised March 18, 2009. First published July 24, 2009; current version published October 9, 2009. This paper was recommended for publication by Associate Editor P. Rocco and Editors K. Lynch and F. Park upon evaluation of the reviewers' comments. This work was supported in part by the Acceleration Research Program of the Ministry of Education, Science, and Technology of the Republic of Korea and the Korea Science and Engineering Foundation under Grant R17-2008-021-01000-0, in part by the Korea Health 21 R&D Project, Ministry of Health and Welfare, Republic of Korea, under Grant A020603, in part by the Agency for Defence Development and by the Unmanned Technology Research Center, Korea Advanced Institute of Science and Technology, in part by the IT R&D Program of the Ministry of Knowledge Economy/the Institute of Information Technology Assessment (2008-F-038-1, Development of Context Adaptive Cognition Technology), in part by the U.S. National Science Foundation under Grant CISE CNS-0423253 and Grant CISE IIS-0805495, and in part by the U.S. Department of Commerce under Grant TOP-39-60-04003.

K. Kim is with Korea Institute of Science and Technology, Seoul 130-650, Korea (e-mail: khk@postech.ac.kr).

W. K. Chung is with Pohang University of Science and Technology, Pohang 790-784, Korea (e-mail: wkchung@postech.ac.kr).

M. C. Çavuşoğlu is with Case Western Reserve University, Cleveland, OH 44106 USA (e-mail: cavusoglu@case.edu).

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org, provided by the author. This material serves as a supplement to MATLAB codes appearing in the paper. These codes require MATLAB Simulink, SimMechanics Toolbox, and Virtual Reality Toolbox (optional). This material includes 14 files, including simulation.wmv (a movie clip). In order to execute the simulation described in Section V, in the paper, open "simulation.mdl," run "run.m," and run "plot_data.m." Then, one can see the same simulation results as described in the paper. Other features are described in "Readme.txt." Contact khk@postech.ac.kr for further questions about this work.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2009.2024789

[1] Jacobian of the master device $\boldsymbol{J}_m : \dot{\boldsymbol{q}}_m \in \mathbb{R}^m \to \dot{\boldsymbol{x}}_m \in \mathbb{R}^r$ and Jacobian of the slave manipulator $\boldsymbol{J}_s : \dot{\boldsymbol{q}}_s \in \mathbb{R}^n \to \dot{\boldsymbol{x}}_s \in \mathbb{R}^r$, where $\boldsymbol{q}_m$ and $\boldsymbol{q}_s$ are the joint angles of the master and slave robot in the joint spaces $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively. The variables $\boldsymbol{x}_m$ and $\boldsymbol{x}_s$ are the pose of the master and slave robot in the task space $\mathbb{R}^r$. α is an arbitrary nonzero scalar. Note that, as defined, KDS is different from geometric dissimilarity. For example, if the length scale ratio of a master device is the same as that of a slave robot, it is geometrically similar but is KDS.

TABLE I
APPLICATION RANGE

Type	Description	Conventional position error feedback	Restriction Space Projection (RSP) method in [14]	Restriction Space Projection (RSP) method proposed here
KS	$\boldsymbol{J}_m = \alpha \boldsymbol{J}_s$ Kinematically similar haptic interface	Applicable	Applicable	Applicable
KDS-EQDOF	$\boldsymbol{J}_m \neq \alpha \boldsymbol{J}_s,\ m = n$ Kinematically dissimilar haptic interface with the same degrees-of-freedom	Partially applicable with single gain task space controller at the slave	Partially applicable to single link collision only	Applicable
KDS-RMASTER	$\boldsymbol{J}_m \neq \alpha \boldsymbol{J}_s,\ m > n$ Kinematically dissimilar haptic interface with insufficient degrees-of-freedom at the slave side	Not applicable	Partially applicable to single link collision only	Applicable
KDS-RSLAVE	$\boldsymbol{J}_m \neq \alpha \boldsymbol{J}_s,\ m < n$ Kinematically dissimilar haptic interface with redundancy at the slave side	Not applicable	Not applicable	Applicable

Previously, Kim *et al.* proposed a position-sensor-based force reflection method, called the restriction space projection (RSP) method [14], which is based on the projection of the restriction space encountered by the slave manipulator in contact with a rigid obstacle in the environment to the master. However, the RSP method proposed in [14] is limited to *specific* joint constraints and KDS conditions, was not able to calculate the restriction space accurately in cases with multiple obstacle collisions, and was also not able to handle redundancy at the slave side at all. This paper significantly extends the RSP method to the *general* case under any type of constraint for all of the KDS conditions listed in Table I.

The remainder of this paper is organized as follows. Section II discusses the limitation of the conventional position-sensor-based force reflection method, namely the position error feedback method, through a case study. In Section III, the instantaneous restriction space (IRS) is introduced and classified by the sources of constraints. Section IV discusses the implementation of the generalized RSP framework. A new obstacle avoidance technique is introduced to deal with redundancies at the slave side. This section also explains how the proposed method can be combined with force-sensor-based force reflection framework existing in the previous literature. Section V confirms the validity of the proposed framework through simulations of a KDS teleoperation system with a redundant slave manipulator under diverse constraint conditions. The conventional position-sensor-based method is compared with the proposed framework, followed by the discussion and concluding remarks in Section VI.

II. Limitation of the Conventional Position Error Feedback Method

This section discusses the limitation of the conventional position-sensor-based force reflection method, namely the position error feedback method, when an unexpected obstacle constrains the motion of the slave robot in a KDS with equal number of DOFs (KDS-EQDOF) bilateral teleoperation system.

Consider a 2-DOF bilateral teleoperation system. When $\boldsymbol{x}_m(t) \in \mathbb{R}^2$ and $\boldsymbol{x}_s(t) \in \mathbb{R}^2$ are the positions of the master device and the slave robot, respectively, the force reflection in the position error feedback framework is calculated as

$$\boldsymbol{F}_R = K_m(\boldsymbol{x}_s - \boldsymbol{x}_m) \tag{1}$$

where $\boldsymbol{F}_R \in \mathbb{R}^2$ is force reflection and $K_m \in \mathbb{R}$ is a scalar force reflection gain. However, in (1), $\boldsymbol{x}_s$ depends on the local controller of the slave robot, as well as the robot dynamics, even in the steady state. Therefore, the direction of $\boldsymbol{F}_R$ would depend on the local controller, even with the same desired command and the constraint conditions. It was demonstrated in [14] that the joint space controller would not be able to correctly reflect the constraints resulting from collision involving slow motion links (as opposed to end-effector) when the master and slave are KDS.

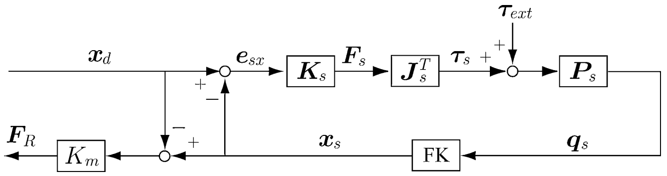


Fig. 1. Conventional force reflection framework when the slave robot uses a task-space position controller.

In the remainder of this section, the behavior of KDS teleoperation system with the task space (also referred to as Cartesian space) controller shown in Fig. 1 will be explained with a case study, demonstrating that they also suffer from the same issue.

In Fig. 1, $\boldsymbol{x}_d$ is desired position command from the master side, $\boldsymbol{K}_s$ is the task-space controller, $\boldsymbol{e}_{sx}(= \boldsymbol{x}_d - \boldsymbol{x}_s)$ is the control error, $\boldsymbol{F}_s$ is the control force, $\boldsymbol{J}_s$ is the Jacobian of the slave robot, $\boldsymbol{\tau}_s$ is the control torque, $\boldsymbol{\tau}_{ext}$ is the external torque distributed to the slave joints from obstacle collisions, $\boldsymbol{P}_s$ is the slave robot dynamics, and FK is the forward kinematics map used to calculate slave robot position $\boldsymbol{x}_s$, from joint angles, and $\boldsymbol{q}_s$. The variable $\boldsymbol{x}_d$ is assumed to be $\boldsymbol{x}_m$.

For the case study, we tested two task-space controllers with various control gains for $\boldsymbol{K}_s$. The first is a conventional position-derivative (PD) task-space controller described as

$$\boldsymbol{F}_s = \boldsymbol{K}_s \boldsymbol{e}_{sx} = \left(\boldsymbol{K}_p + \boldsymbol{K}_v \frac{d}{dt}\right) \boldsymbol{e}_{sx} \tag{2}$$

where

$$\boldsymbol{K}_p = \begin{bmatrix} K_{px} & 0 \\ 0 & K_{py} \end{bmatrix} \qquad \boldsymbol{K}_v = \begin{bmatrix} K_{vx} & 0 \\ 0 & K_{vy} \end{bmatrix}.$$

The second is a model-based task-space controller that uses torque control law (e.g., [15]), which is described as

$$\boldsymbol{F}_s = \boldsymbol{K}_s \boldsymbol{e}_{sx} = \tilde{\boldsymbol{M}}(\boldsymbol{q}_s)\left\{\ddot{\boldsymbol{x}}_d + \left(\boldsymbol{K}_p + \boldsymbol{K}_v \frac{d}{dt}\right)\boldsymbol{e}_{sx}\right\} + \tilde{\boldsymbol{C}}(\boldsymbol{q}_s, \dot{\boldsymbol{q}}_s)\dot{\boldsymbol{x}}_s \tag{3}$$

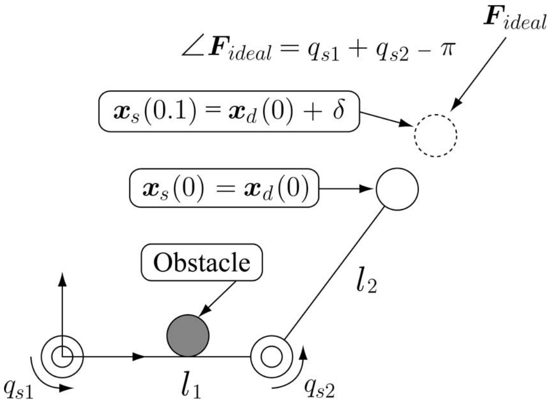

$\boldsymbol{F}_{ideal}$
$\angle \boldsymbol{F}_{ideal} = q_{s1} + q_{s2} - \pi$
$\boldsymbol{x}_s(0.1) = \boldsymbol{x}_d(0) + \delta$
$\boldsymbol{x}_s(0) = \boldsymbol{x}_d(0)$
Obstacle
l_2
q_{s1}
l_1
q_{s2}

Fig. 2. Ideal force reflection $\boldsymbol{F}_{ideal}$ during an obstacle collision at the first link of 2-DOF slave robot.

where $\tilde{\boldsymbol{M}} = \boldsymbol{J}^{-T}\boldsymbol{M}\boldsymbol{J}^{-1}$, and $\tilde{\boldsymbol{C}} = \boldsymbol{J}^{-T}(\boldsymbol{C}\boldsymbol{J}^{-1} + \boldsymbol{M}(d/dt)(\boldsymbol{J}^{-1}))$ when the dynamics of the slave robot in joint space has the form

$$\boldsymbol{\tau}_s + \boldsymbol{\tau}_{ext} = \boldsymbol{M}(\boldsymbol{q}_s)\ddot{\boldsymbol{q}}_s + \boldsymbol{C}(\boldsymbol{q}_s, \dot{\boldsymbol{q}}_s)\dot{\boldsymbol{q}}_s \tag{4}$$

where $\boldsymbol{M}$ and $\boldsymbol{C}$ represent the inertial and the centrifugal properties.

The kinematics of the slave robot is given by

$$\boldsymbol{x}_s(t) = \begin{bmatrix} x_s(t) \\ y_s(t) \end{bmatrix} = \begin{bmatrix} l_1 C(q_{s1}(t)) + l_2 C(q_{s1}(t) + q_{s2}(t)) \\ l_1 S(q_{s1}(t)) + l_2 S(q_{s1}(t) + q_{s2}(t)) \end{bmatrix} \tag{5}$$

where $C(\cdot)$ and $S(\cdot)$ represent $\cos(\cdot)$ and $\sin(\cdot)$, respectively, $q_{s1}(t)$ and $q_{s2}(t)$ are the first and the second revolute joint angles of the slave robot, l_1 and l_2 are the lengths of the first and the second link of the slave robot, and m_i and I_i $(i = 1,\ 2)$ are the mass and the inertia of ith link.

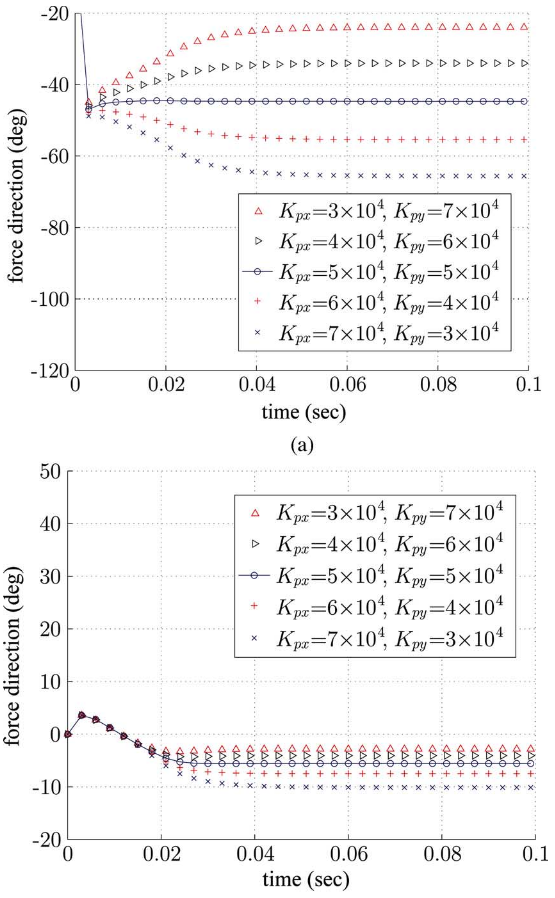
We have built a custom simulator using MATLAB Simulink, Virtual Reality Toolbox, and SimMechanics Tool box. $l_1 = 0.2$ m, $l_2 = 0.3$ m, $m_1 = 0.2$ kg, $m_2 = 0.3$ kg, $I_1 = 727.2 \times 10^{-6}$ kg·m^2, and $I_2 = 2440.80 \times 10^{-6}$ kg·m^2. $K_{vx} = K_{vy} = 500$ N·s/m. K_{px} and K_{py} vary from 3×10^4 to 7×10^4 N/m, and $K_m = 1.0$. The simulation is operated for 0.1 s. The initial angle is $\boldsymbol{q}_s(0) = [0, 45]^T$ (in degrees). Initially, the master and the slave robot have the same position, i.e., $\boldsymbol{x}_m(0) = \boldsymbol{x}_s(0)$. The desired position is given as

$$\boldsymbol{x}_d(t) = \left\{ \begin{array}{ll} \boldsymbol{x}_s(0) + 0.5\boldsymbol{\delta}(1 - C(\pi t/0.02)), & \text{if } 0 < t < 0.02 \\ \boldsymbol{x}_s(0) + \boldsymbol{\delta}, & \text{if } 0.02 \leq t < 0.1 \end{array} \right\}$$

where $\boldsymbol{\delta} = [0.02, 0.02]^T$.

However, since there is an obstacle that constrains the motion of the first joint when $q_{s1} > 0$, the slave robot cannot follow the desired position. The stiffness and the damping coefficient of the obstacle are 10^5 and 10^2 N·s/m, respectively.

As illustrated in Fig. 2, the direction of the ideal force reflection is the direction of the second link $(q_{s1} + q_{s2} - \pi)$ since it is the direction the slave robot cannot achieve due to the obstacle collision at the first link. However, the calculated force reflection of the simulation results shown in Fig. 3 is deviated from the ideal force reflection direction (-45°) depending on the controller used and the control gains. Under the position error feedback framework, the force reflection gives inaccurate reflection force directions since the resulting second joint angle depends on the control gains. As a result, the human operator would perceive an incorrect configuration of the constraints at the slave side.



(a)

(b)

Fig. 3. Calculated force reflection under conventional position error feedback framework when the first joint motion is constrained in a 2-DOF bilateral teleoperation system. (a) Force reflection direction under the conventional PD task-space controller. The amplitudes are 0.018–0.020 N. (b) Force direction under the computed torque method task-space controller. The amplitudes are 0.023–0.027 N.

Note that the direction of the force reflection from the position error feedback method corresponds to that of $\boldsymbol{F}_{ideal}$ if the PD controller in (2) is used and if $K_{px} = K_{py}$ [see circle and line plot in Fig. 3(a)], i.e., a single-gain PD task-space controller. However, the major drawback is to use the same controller gain for all of the axes for a multi-DOF slave robot. Especially, when the task space includes both position and orientation, the use of a single gain is problematic as the relative scaling between the position and orientation gains would arbitrarily change the force reflection direction.

Furthermore, when an advanced-model-based controller, such as a computed torque, $\mathcal{H}_\infty$, or μ-synthesis-based controller, is used, even the use of isotropic control gains does not guarantee accurate direction of force reflection. In Fig. 3(b), the force reflection results with a task-space computed torque controller are shown. In the task-space computed torque method, the position error is multiplied with the nominal mass matrix $\tilde{\boldsymbol{M}}$, as shown in (3). Therefore, even if isotropic PD gains $(K_{px} = K_{py})$ are used, an isotropic control output for each axes is not guaranteed as the mass matrix is typically not isotropic.

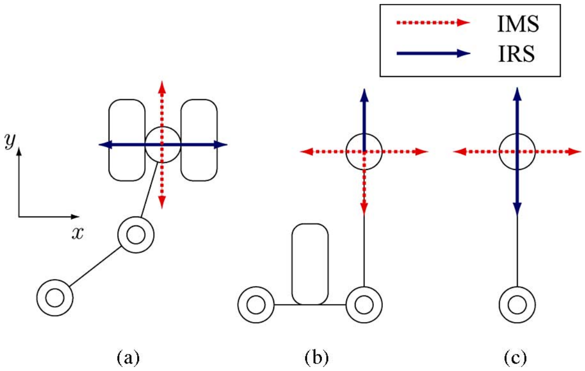


Fig. 4. Examples of IRS (solid arrows) and IMS (dotted arrows) in 2-D x–y task space. (a) IRS due to an exogenous endpoint constraint. (b) IRS and IMS due to the exogenous joint constraint. (c) IRS and IMS due to the insufficient DOFs to the 2-D task space.

Consequently, the conventional position error feedback framework has a fundamental limitation in representing the accurate force direction, since the reflection force is calculated based on the slave endpoint position.

Furthermore, the conventional position-error-based force feedback algorithm does not provide a convenient method to incorporate signals from a force sensor, if one is available.

III. Introduction of IRS

Before dealing with the issue demonstrated in Section II, we need to understand the motion space and the restriction space. During motion in the free space, there are no constraints on the motion of the slave robot, so the operator will feel no reflection forces at all. However, when the motion of the slave robot is constrained, the operator needs to feel the reflection force at the master to prevent the operator from moving the teleoperator into the constraint space. Therefore, the objective of a bilateral teleoperation system should be the creation of a restriction space at the master side that is identical to the restriction space at the slave against the operator's infeasible desired command, rather than matching the end position of the master device to that of the slave robot, such as the technique used in position error feedback method. We can consider the position error feedback method as one of the simplest method to generate restriction space for KS type (see Table I) without considering the force reflection direction.

Then, we begin with a discussion of the concept of IRS. Fig. 4 shows the examples of IRS of manipulators in 2-D x–y Cartesian space when $\boldsymbol{x} \in \mathbb{R}^2$. In Fig. 4(a), an obstacle constrains the end-effector of the slave robot in x-direction. Therefore, the slave robot can only make a motion in the y-direction instantaneously, and we call the direction as the instantaneous motion space (IMS). The algebraic complement of IMS is defined as IRS. Therefore, x-direction in Fig. 4(a) is in IRS. Fig. 4(b) shows that the positive direction of the first joint is constrained by an obstacle so that the motion space is positively spanned by the second joint and the negative direction of the first joint, and therefore, the created IRS is instantaneously in the positive y-direction. Fig. 4(c) shows that IRS is created by the insufficient DOFs of a manipulator compared with the dimension of the given task space. In this case, the manipulator has only one DOF, even though the task space is 2-D x–y space, so that the manipulator's motion space cannot span the whole task space. IRS in Fig. 4(c) is the tangential direction of the link, i.e., y-direction. In Fig. 4(a) and (b), IRS is created by the exogenous constraints like obstacle and link collisions. In Fig. 4(c), IRS is generated by insufficient DOFs compared with the given task space. Therefore, IRS can be defined as follows.

IMS convex cone of all possible velocity vector that the manipulator can kinematically achieve[2];
IRS algebraic complement of IMS;
$\mathrm{IRS_G}$ IRS caused by insufficient DOFs;
$\mathrm{IRS_E}$ IRS caused by exogenous constraints.

IMS and IRS can be defined mathematically as follows when joints are bidirectionally constrained:

$$\mathrm{IMS} = \mathcal{R}(\boldsymbol{J}_{s\ominus}) \tag{6}$$

$$\mathrm{IRS_G} = \mathcal{R}(\boldsymbol{J}_s)^{\perp} \tag{7}$$

$$\mathrm{IRS_E} = \mathcal{R}(\boldsymbol{J}_s) \cap \mathcal{R}(\boldsymbol{J}_{s\ominus})^{\perp} \tag{8}$$

where $\mathcal{R}(\cdot)$ stands for the range space of $(\cdot)$, $(\cdot)^{\perp}$ is the orthogonal complementary space of $(\cdot)$, $\boldsymbol{J}_s = \partial \boldsymbol{x}_s / \partial \boldsymbol{q}_s \in \mathbb{R}^{r \times n}$, where $\boldsymbol{x}_s \in \mathbb{R}^r$ and $\boldsymbol{q}_s \in \mathbb{R}^n$ are the pose and the joint angles of the slave robot, $\boldsymbol{J}_{s\ominus} \in \mathbb{R}^{r \times n}$ is defined as an inherited Jacobian from $\boldsymbol{J}_s$, and the columns of $\boldsymbol{J}_{s\ominus}$ corresponding to the constrained joints are replaced by zero column vectors. In order to detect the constrained joints, we introduce a certain threshold ε to determine whether the source of the joint angle errors come from the motion constraints by obstacles or from the free motion control errors. In other words, when the joint angle error is larger than the ε, the corresponding column of $\boldsymbol{J}_{s\ominus}$ is replaced by a zero vector. The threshold should be larger than the expected free motion control errors. It determines the sensitivity to trigger the proposed method. $\mathcal{R}(\boldsymbol{J}_s)^{\perp}$ is the restriction space that the slave robot cannot reach due to the insufficient DOFs, i.e., $\mathrm{IRS_G}$, even if there is no exogenous motion constraint. If the motion of the slave robot is constrained, IMS becomes $\mathcal{R}(\boldsymbol{J}_{s\ominus})$. $\mathrm{IRS_E}$ becomes $\mathcal{R}(\boldsymbol{J}_s) \cap \mathcal{R}(\boldsymbol{J}_{s\ominus})^{\perp}$.

IV. Restriction Space Projection Method

We have developed a new position-sensor-based force reflection method, which is called the RSP method, that describes the accurate IRS satisfying the requirements described in Section I. Section IV-A introduces the RSP matrices to calculate IRS in the forms of the force reflection. The implementation issues of the proposed RSP method are discussed in Section IV-B.

A. Calculation of IRS

As defined in (7) and (8), $\mathrm{IRS_G}$ can be calculated from Jacobian of the slave robot, i.e., $\boldsymbol{J}_s$. $\mathrm{IRS_E}$ can be calculated from $\boldsymbol{J}_{s\ominus}$ using the measured joint error configurations and $\boldsymbol{J}_s$. When the desired pose is in IRS, the motion constraint of the slave robot forms the control errors. In the conventional position-sensor-based force reflection method, a scalar (K_m) is multiplied to the task error (1). In the proposed RSP method, however, the task error is projected onto IRS to generate the reflection force. In order to calculate $\mathrm{IRS_G}$ and $\mathrm{IRS_E}$, two RSP matrices are defined: $\boldsymbol{R}_G$ and $\boldsymbol{R}_E$.

From the definition of $\mathrm{IRS_G}$ in (7), $\boldsymbol{R}_G$, which maps the task-space pose error into the reflection force in $\mathrm{IRS_G}$, can be defined as

$$\boldsymbol{R}_G : \boldsymbol{e}_{sx} \in \mathbb{R}^r \rightarrow \boldsymbol{F}_{RG} \in \mathrm{IRS_G} \tag{9}$$

$$\boldsymbol{R}_G = -K_G(\boldsymbol{I} - \boldsymbol{J}_s \boldsymbol{J}_s^{\#}) \tag{10}$$

[2] When there are no half-space-type constraints (resulting from unidirectional constraints), IMS becomes a subspace, and IRS becomes the orthogonal complement subspace of IMS.

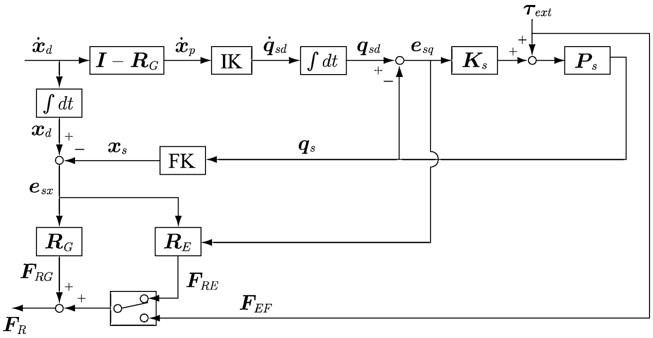


Fig. 5. Implementation of RSP method in a haptic interface. The variable $\dot{\boldsymbol{x}}_d$ is the desired command transferred from the master device, and $\dot{\boldsymbol{x}}_p$ is the projected desired command onto the range space of the slave robot. IK and FK stand for inverse and forward kinematic maps. The variables $\boldsymbol{q}_{sd}$ and $\boldsymbol{q}_s$ are the desired angle and the actual angle of the slave robot. Also, $\boldsymbol{e}_{sq} = \boldsymbol{q}_{sd} - \boldsymbol{q}_s$. The local controller and the dynamics of the slave robot. $\boldsymbol{F}_{\text{ext}}$ is the external forces caused by obstacles. $\boldsymbol{e}_{sx} = \boldsymbol{x}_d - \boldsymbol{x}_s$, where $\boldsymbol{x}_d$ and $\boldsymbol{x}_s$ are the desire pose and the actual pose of the slave robot. $\boldsymbol{R}_G$ and $\boldsymbol{R}_E$ are the RSP matrices. $\boldsymbol{F}_R$, $\boldsymbol{F}_{RG}$, and $\boldsymbol{F}_{RE}$ are the total reflection force, the reflection force by IRS_G, and the reflection force by IRS_E, respectively.

where $(\cdot)^{\#}$ is the pseudoinverse of $(\cdot)$, $\boldsymbol{e}_{sx} (= \boldsymbol{x}_d - \boldsymbol{x}_s)$ is the task-space pose error, r is the workspace dimension, and K_G is a scalar force gain to adjust the magnitude of the reflection force $\boldsymbol{F}_{RG}$.

Similarly, from the definition of IRS_E in (8), $\boldsymbol{R}_E$, which maps the task-space pose error into the reflection force in IRS_E, can be defined as

$$\boldsymbol{R}_E : \boldsymbol{e}_{sx} \in \mathbb{R}^r \rightarrow \boldsymbol{F}_{RE} \in \text{IRS}_\text{E} \tag{11}$$

$$\boldsymbol{R}_E = -K_E(\boldsymbol{I} - \boldsymbol{J}_{s\ominus}\boldsymbol{J}_{s\ominus}^{\#})\boldsymbol{J}_s\boldsymbol{J}_s^{\#} \tag{12}$$

where K_E is a scalar force gain.

Therefore, the total reflection force $\boldsymbol{F}_R$ can be calculated as

$$\begin{aligned}\boldsymbol{F}_R &= \boldsymbol{F}_{RG} + \boldsymbol{F}_{RE} = \boldsymbol{R}_G\boldsymbol{e}_{sx} + \boldsymbol{R}_E\boldsymbol{e}_{sx}\\ &= -\{K_G(\boldsymbol{I} - \boldsymbol{J}_s\boldsymbol{J}_s^{\#}) + K_E(\boldsymbol{I} - \boldsymbol{J}_{s\ominus}\boldsymbol{J}_{s\ominus}^{\#})\boldsymbol{J}_s\boldsymbol{J}_s^{\#}\}\boldsymbol{e}_{sx}. \end{aligned}\tag{13}$$

Although the proposed RSP method calculates the direction of the restriction space, it cannot estimate the magnitude. Therefore, users should scale K_E and K_G to fit to their applications considering weighting between $\boldsymbol{F}_{RG}$ and $\boldsymbol{F}_{RE}$. Note that we can adjust the force gains K_G and K_E independently since $\boldsymbol{F}_{RG} \in \text{IRS}_\text{G}$ and $\boldsymbol{F}_{RE} \in \text{IRS}_\text{E}$ are in orthogonal complementary space.

B. Implementation of IRS Using RSP Method

A general form of the proposed RSP method can be implemented, as shown in Fig. 5. When the desired command $\dot{\boldsymbol{x}}_d$ is transferred from the master side, it is projected to the range space of the slave robot, i.e., $\mathcal{R}(\boldsymbol{I} - \boldsymbol{R}_G)$. Then, the desired joint velocities are calculated from the projected desired command, $\dot{\boldsymbol{x}}_p$ and the desired joint angle of the slave robot $\boldsymbol{q}_{sd}$ through inverse kinematics (IK). The local controller of the slave robot,[3] i.e., $\boldsymbol{K}_s$, moves the slave robot to follow the desired pose. At the same time, the external torques are distributed to the joints due to the obstacle collisions $\boldsymbol{\tau}_{ext}$. $\boldsymbol{P}_s$ stands for the slave robot dynamics. The pose error $\boldsymbol{e}_{sx}$ is calculated from $\boldsymbol{x}_d$ and $\boldsymbol{x}_s$. The variable $\boldsymbol{q}_s$ is the joint angle. FK is a forward kinematic mapping from $\boldsymbol{q}_s$ to $\boldsymbol{x}_s$. From (10), $\boldsymbol{R}_G$ can be calculated from the Jacobian $\boldsymbol{J}_s$. Then, $\boldsymbol{F}_{RG} = \boldsymbol{R}_G\boldsymbol{e}_{sx}$. From $\boldsymbol{J}_s$ and $\boldsymbol{e}_{sq}$, $\boldsymbol{J}_{s\ominus}$ can be calculated, as explained in Section III, and $\boldsymbol{F}_{RE} = \boldsymbol{R}_E\boldsymbol{e}_{sx}$ from (12). $\boldsymbol{F}_{EF}$ is the force signal detected by force sensors.

However, when there is redundancy at the slave robot, we need more consideration for IK solution, i.e., IK in Fig. 5, since the joint error configuration $\boldsymbol{e}_{sq}$ can be changed by the desired joint angle $\boldsymbol{q}_d$ even for the same $\dot{\boldsymbol{x}}_p$.

When $\dot{\boldsymbol{x}}_p$ is determined, IK solution $\dot{\boldsymbol{q}}_{sd}$ can generally be calculated as

$$\dot{\boldsymbol{q}}_{sd} = \boldsymbol{J}_s^{\#}\dot{\boldsymbol{x}}_p + (\boldsymbol{I}_n - \boldsymbol{J}_s^{\#}\boldsymbol{J}_s)\boldsymbol{z} \tag{14}$$

where $\boldsymbol{I}_n$ is an identity matrix in $\mathbb{R}^{n\times n}$, and $\boldsymbol{z}$ is an arbitrary vector in $\mathbb{R}^n$. The solution depends on $\boldsymbol{z}$ when $\dot{\boldsymbol{x}}_p$ and $\boldsymbol{J}_s$ are determined. If there is null space $\mathcal{N}(\boldsymbol{J}_s) = \mathcal{R}(\boldsymbol{I}_n - \boldsymbol{J}_s^{\#}\boldsymbol{J}_s)$, then there are infinite number of IK solutions for the given desired command $\dot{\boldsymbol{x}}_p$.

Since IRS is calculated from the joint angle error configuration in the RSP framework, an appropriate IK solution should be sought to decrease the joint angle error avoiding the motion of the constrained joint. Thus, it is possible to allow the human operator to move the master device in larger IMS. We propose the following IK solution inspired by the IK using a potential function as a second manipulation variable [16], [17]:

$$\begin{aligned}\dot{\boldsymbol{q}}_{sd} &= \boldsymbol{J}_s^{\#}\dot{\boldsymbol{x}}_d\\ &\quad + (\boldsymbol{I}_n - \boldsymbol{J}_s^{\#}\boldsymbol{J})\left(-k_1\frac{\partial p}{\partial \boldsymbol{q}_{sd}}^T - k_2\frac{\partial p}{\partial \boldsymbol{q}_{sd}}^T\frac{\partial p}{\partial \boldsymbol{q}_{sd}}\boldsymbol{J}_s^{\#}\dot{\boldsymbol{x}}_d\right)\end{aligned}\tag{15}$$

where $p = (1/2)\boldsymbol{e}_{sq}^T\boldsymbol{e}_{sq} = (1/2)(\boldsymbol{q}_{sd} - \boldsymbol{q}_s)^T(\boldsymbol{q}_{sd} - \boldsymbol{q}_s)$, $k_1 > 0$, and $k_2 = \|(\partial p/\partial \boldsymbol{q}_{sd})(\boldsymbol{I}_n - \boldsymbol{J}_s^{\#}\boldsymbol{J}_s)\|^{-2}$. Unless $(\partial p/\partial \boldsymbol{q}_{sd})(\boldsymbol{I}_n - \boldsymbol{J}_s^{\#}\boldsymbol{J}_s) = \boldsymbol{0}$, $\dot{\boldsymbol{q}}_{sd}$ from (15) contributes to decrease the value of p using its null motion since

$$\begin{aligned}\frac{\partial p}{\partial \boldsymbol{q}_{sd}}\dot{\boldsymbol{q}}_{sd} &= \frac{\partial p}{\partial \boldsymbol{q}_{sd}}\Bigg\{\boldsymbol{J}_s^{\#}\dot{\boldsymbol{x}}_d + (\boldsymbol{I}_n - \boldsymbol{J}_s^{\#}\boldsymbol{J})\\ &\qquad \left(-k_1\frac{\partial p}{\partial \boldsymbol{q}_{sd}}^T - k_2\frac{\partial p}{\partial \boldsymbol{q}_{sd}}^T\frac{\partial p}{\partial \boldsymbol{q}_{sd}}\boldsymbol{J}_s^{\#}\dot{\boldsymbol{x}}_d\right)\Bigg\}\\ &= -k_1\frac{\partial p}{\partial \boldsymbol{q}_{sd}}(\boldsymbol{I}_n - \boldsymbol{J}_s^{\#}\boldsymbol{J}_s)\frac{\partial p}{\partial \boldsymbol{q}_{sd}}^T\\ &= -k_1\frac{\partial p}{\partial \boldsymbol{q}_{sd}}(\boldsymbol{I}_n - \boldsymbol{J}_s^{\#}\boldsymbol{J}_s)(\boldsymbol{I}_n - \boldsymbol{J}_s^{\#}\boldsymbol{J}_s)^T\frac{\partial p}{\partial \boldsymbol{q}_{sd}}^T\\ &\le 0.\end{aligned}\tag{16}$$

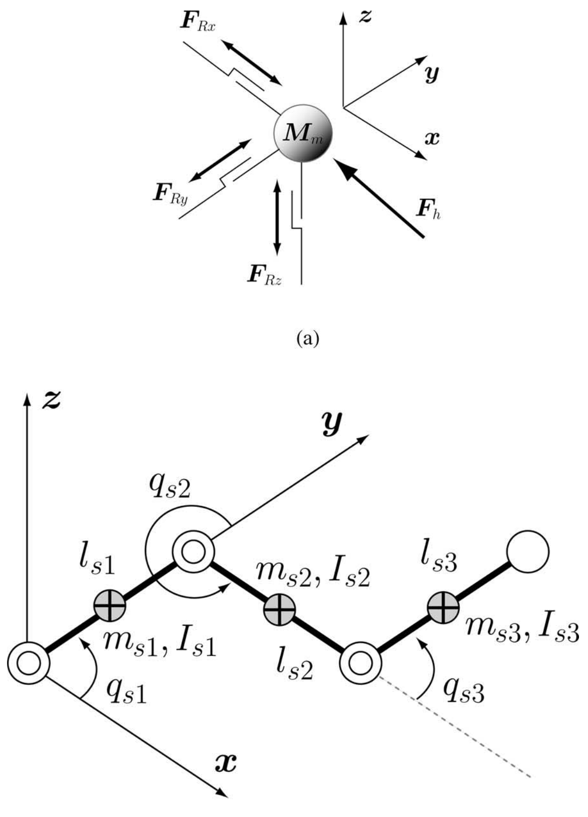
Note that $\boldsymbol{I}_n - \boldsymbol{J}_s^{\#}\boldsymbol{J}_s$ is an idempotent matrix.

Therefore, we can conclude that the IK solution in (15) changes the desired joint angle configuration to decrease the joint angle error avoiding the joint motion constraints.

Since IRS is transferred to the master side in forms of force reflection, it can use force sensor signals in cases when the force sensor can detect collisions. If there are nonzero force signals from the force sensor ($\boldsymbol{F}_{EF} \neq \boldsymbol{0}$), then $\boldsymbol{F}_R = \boldsymbol{F}_{RG} + \boldsymbol{F}_{EF}$. In cases when the force sensor misses a collision ($\boldsymbol{F}_{EF} = \boldsymbol{0}$) but there is calculated restriction space ($\boldsymbol{F}_{RE} \neq \boldsymbol{0}$), $\boldsymbol{F}_R = \boldsymbol{F}_{RG} + \boldsymbol{F}_{RE}$. In Section V, simulation has been performed to demonstrate the use of force sensor signal in the proposed framework.

Since the input is the velocity $\dot{\boldsymbol{x}}_d$ and the output is the force $\boldsymbol{F}_R$, the architecture of the proposed RSP method is similar to p-f architecture, as shown in Fig. 5. $\boldsymbol{I} - \boldsymbol{R}_G$, $\boldsymbol{R}_G$, and $\boldsymbol{R}_E$ are just projection matrices that change the direction of the vector signals. Thus, the stability analysis techniques for p-f architecture in previous literature [1], [3], [5]–[12] can be applied in the same manner to the bilateral teleoperation systems using the proposed RSP framework.

[3] The proposed RSP works regardless of the local control algorithm of the slave robot.



(a)

(b)

Fig. 6. Kinematics of the master device and the slave robot in a KDS bilateral teleoperation system. (a) Master device. (b) Slave robot.

In the case when force command is delivered from the master device instead of position command, we can introduce a virtual admittance model to calculate desired position $\dot{\boldsymbol{x}}_d$ from the force command, and the proposed framework can be used in the same manner.

V. Simulation

This section performs a simulation to validate that the proposed RSP framework in a multi-DOFs KDS bilateral teleoperation system under various constraint conditions. The reflection forces given by the conventional position error feedback method are also compared with those given by the proposed RSP method.

In this simulation, the bilateral teleoperation system has a 3-DOF Cartesian master device in x–y–z space and a 3-DOF planar slave robot in x–y space. The slave robot has 1-DOF redundancy in x–y planar space and no DOFs in the z-direction. Therefore, the bilateral teleoperation system can be classified into mixed KDS condition of KDS-RMASTER and KDS-RSLAVE (see Fig. 6). The slave robot has a force sensor at the end-effector. In the master device, the end-effector is connected to three translation actuators that can display reaction forces in the x-, y-, and z-directions, respectively. We assume that human position is same as the master device's position, i.e., a rigid master. The dynamics of the master device coupled with the human operator are assumed to be represented, as shown in Fig. 7, as

$$\boldsymbol{F}_h - \boldsymbol{F}_R = \boldsymbol{M}_m \ddot{\boldsymbol{x}}_d \tag{17}$$

$$\boldsymbol{F}_h = K_{hp}(\boldsymbol{x}_{target} - \boldsymbol{x}_d) + K_{hv}(\dot{\boldsymbol{x}}_{target} - \dot{\boldsymbol{x}}_d). \tag{18}$$

Equation (17) gives the master device dynamics, where $\boldsymbol{F}_h$ $(=[F_{hx}, F_{hy}, F_{hz}]^T)$ and $\boldsymbol{F}_R$ $(=[F_{Rx}, F_{Ry}, F_{Rz}]^T)$ are the human force and the reflection force transferred from the slave, respectively. Equation (18) gives the simulated human operator dynamics. Therefore, the velocity command $\dot{\boldsymbol{x}}_d$ is transferred to the slave robot side. For the slave robot, l_{si} $(i = 1, 2, \text{and } 3)$ is the length of each link. The variables m_{si} and I_{si} $(i = 1, 2, \text{and } 3)$ are the mass and the inertia of each link.

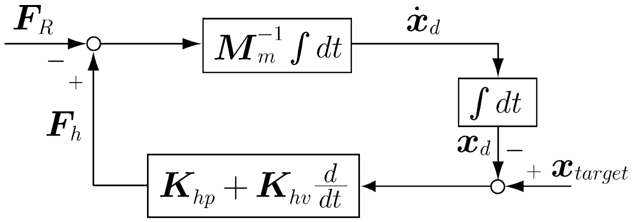

$\boldsymbol{F}_R$
$-$
$+$
$\boldsymbol{M}_m^{-1} \int dt$
$\dot{\boldsymbol{x}}_d$
$\int dt$
$\boldsymbol{F}_h$
$\boldsymbol{x}_d$
$-$
$+$
$\boldsymbol{x}_{target}$
$\boldsymbol{K}_{hp} + \boldsymbol{K}_{hv}\frac{d}{dt}$

Fig. 7. Block diagram at the master side.

TABLE II
Kinematic and Dynamic Properties of the Bilateral Teleoperation System

i	1	2	3
l_{si}(m)	0.2	0.3	0.2
m_{si}(kg)	0.2	0.3	0.2
$I_{s3}(\times 10^{-6}$kg m$^2)$	727.2	2440.1	727.2
q_{si}(deg)	90.0	-90.0	90.0
$\boldsymbol{M}$ (kg)	$\boldsymbol{I}_{3\times3}$		
K_{hp} (N/m)	10		
K_{hv} (N/m/s)	10		
$\boldsymbol{x}_{target}$ (m)	$[0.0, 0.5, 0.1]^T$		
K_G	100		
K_E	100		
ε (deg)	0.5		
$\boldsymbol{K}_s$(slave local controller, (See Fig.5)	$(5\int dt + 10 + d/dt)\boldsymbol{I}_{3\times3}$		

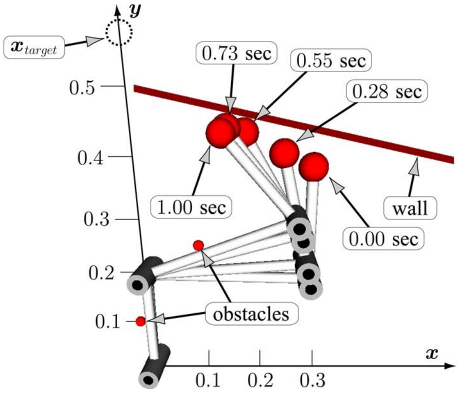
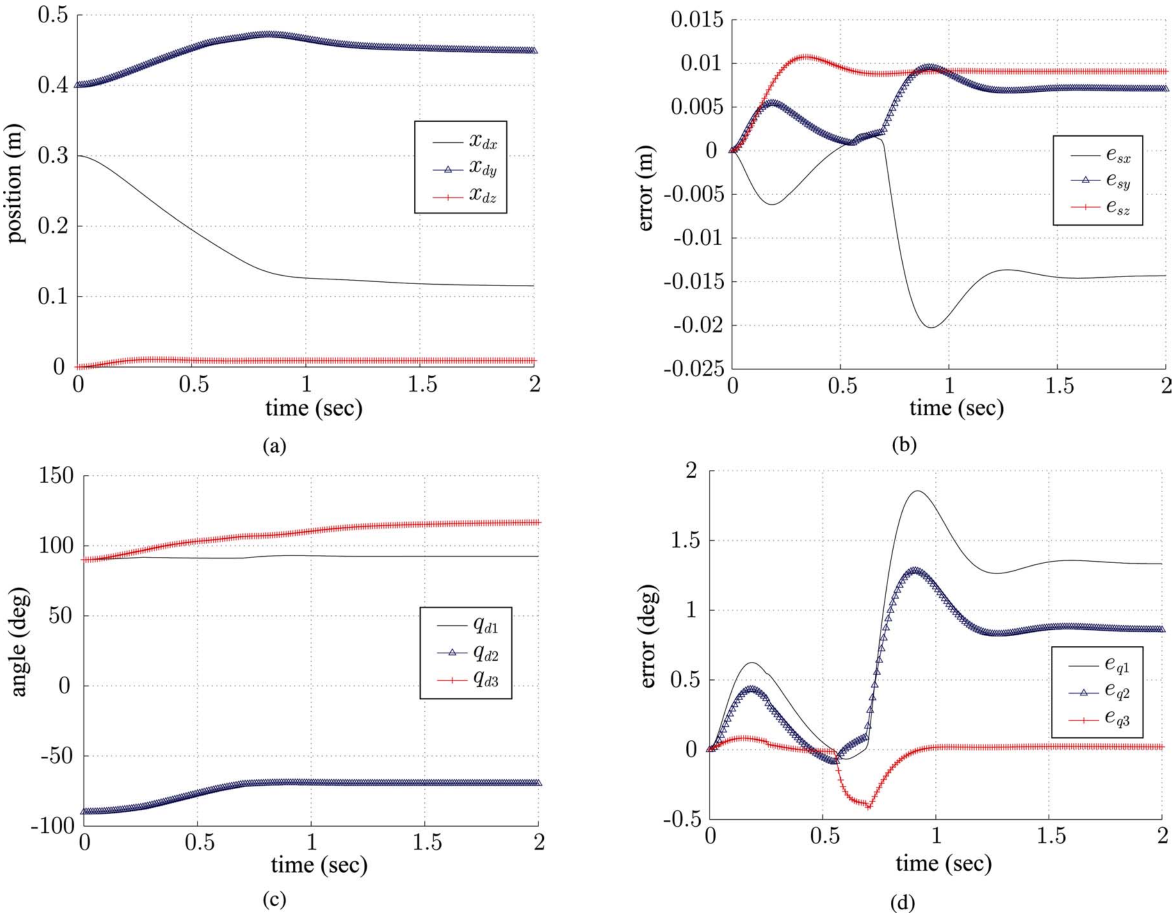


Fig. 8. Simulation procedure.

We have built a custom simulator using MATLAB Simulink 6.4, Virtual Reality Toolbox 4.3, and SimMechanics Tool box 2.4, which are available as a multimedia supplement to this paper. Table II summarizes the kinematic and dynamic parameters, as well as the control gains, used in the simulation. In this simulation, there are two sphere obstacles at $[-0.012, 0.1, 0]^T$ (in meters) and $[0.1, 0.25, 0]^T$ (in meters), both with the radius of 0.01 m and a wall along the line passing through two points $[0.3, 0.43]^T$ and $[0.0, 0.5]^T$. The stiffness and the damping coefficients of the obstacles are 10^5 N/m and 10^2 N/m/s, respectively. The wall stiffness and damping coefficients are 10^3 N/m and 10^2 N/m/s, respectively. For the RSP matrices, both reflection force gains K_G and



(a) (b) (c) (d)

Fig. 9. Simulation results. (a) Desired pose $\boldsymbol{x}_d$. (b) Pose error $\boldsymbol{e}_s = \boldsymbol{x}_d - \boldsymbol{x}_s$. (c) Desired angle $\boldsymbol{q}_d$. (d) Angle error $\boldsymbol{e}_q = \boldsymbol{q}_d - \boldsymbol{q}_s$.

K_E are 100.0. The angle error threshold used to determine the joint collision ε is $0.5°$.

At $t = 0.28$ s, the first link hits an obstacle. Since the slave robot has 1-DOF redundancy, it can follow the desired pose in x–y plane using the proposed IK solutions in (15). At $t = 0.55$ s, the end-effector hits a wall. The sum of $\boldsymbol{F}_{EF}$ and $\boldsymbol{F}_{RG}$ are transferred to the master side. At $t = 0.73$, the second link hits an obstacle, and the end-effector is dropped from the wall. Although the force sensor cannot detect the joint constraints, the RSP algorithm detects restriction space. $\boldsymbol{F}_{RE}$ calculated from the proposed RSP method is transferred to the master side. The slave robot configuration is illustrated in Fig. 8.

From the simulation results summarized in Figs. 9 and 10, the following are discussed:

1) the calculated IRS_G and the corresponding reflection force $\boldsymbol{F}_{RG}$;
2) the validity of the proposed IK solution (15) when $0.28 < t < 0.55$;
3) the force reflection when the collision is detected by the force sensor when $0.55 < t < 0.73$;
4) the calculated IRS_E and the corresponding reflection force $\boldsymbol{F}_{RE}$ when the force sensor does not detect joint constraints from $t > 0.73$;
5) the comparison of the reflection forces calculated by the proposed RSP method and the conventional position error feedback method.

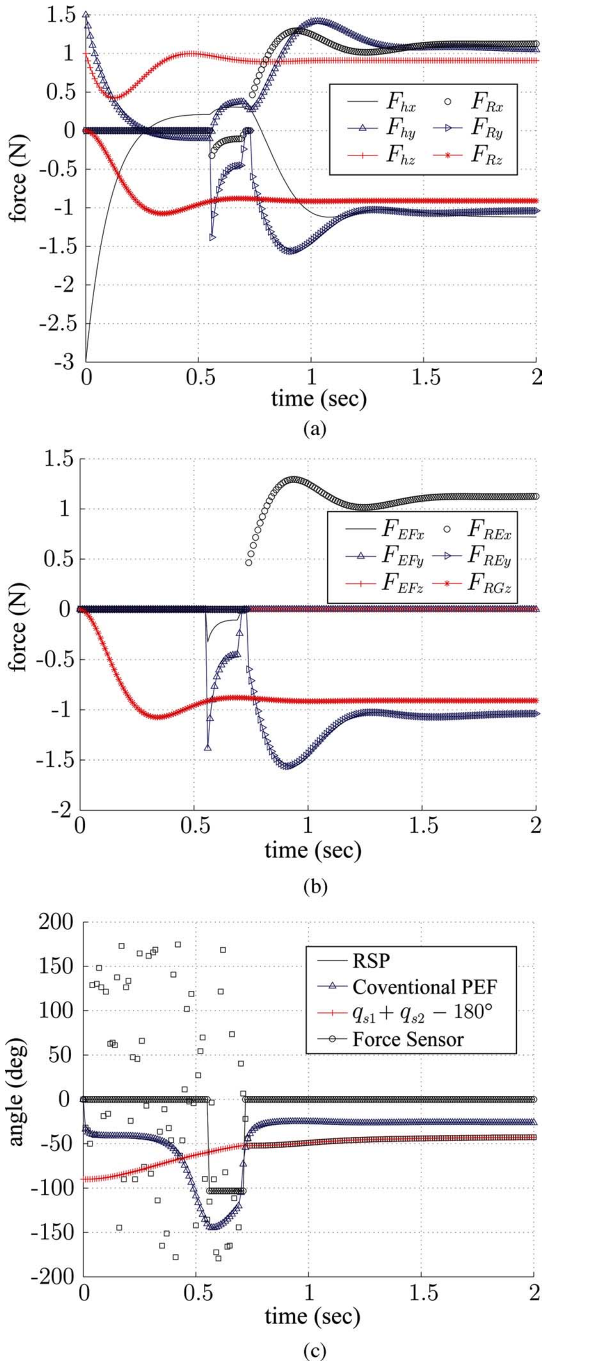
1) *Calculation of IRS_G:* Since the slave robot motion spans only the x–y plane, z-direction forms the restriction space, i.e., IRS_G. In Fig. 10(a), in order to move the master device to the target pose, F_{hz} is applied to the master device. However, the IRS_G is calculated using the proposed RSP method, and F_{Rz} restricts the motion in the z-direction. As a result, the z-direction motion of the master device is constrained with negligible error [see x_{dz} in Fig. 9(a) and e_{sz} in Fig. 9(b)].

2) *Validity of the Proposed IK Solution:* At $t = 0.28$ s, the first link collides with an obstacle, and the motion is constrained. However, the IK solution in (15) avoids the motion of the constrained joint. The first joint moves in a way to minimize the joint error, while the second and the third joint move to follow the master device pose. As a result, the slave robot follows the desired pose maintaining negligibly small tracking error, as shown in Fig. 9(b) and (d), until $t = 0.55$.

3) *Force Reflection Using Force Sensor Signals:* When $0.55 < t < 0.73$, the force sensor detects the collision of the end-effector. In Fig. 10, the force sensor detects the collision, and the reflection force is calculated as $\boldsymbol{F}_R = \boldsymbol{F}_{EF} + \boldsymbol{F}_{RG}$.

4) *Calculation of IRS_E:* At $t = 0.73$, both the first and the second link are constrained by obstacles. The slave robot has IMS in the normal direction of link 3. Fig. 9(b) and (d) shows that the pose error, as well as the angle error, increases significantly. Then, the proposed RSP method calculates the IRS_E and reflects the force $\boldsymbol{F}_R$ $(= \boldsymbol{F}_{RG} + \boldsymbol{F}_{RE})$ to the master device, as shown in Fig. 10(a), even though the force sensor cannot detect the constraints. As a result, the pose errors converge into small values where $\boldsymbol{F}_h$ and $\boldsymbol{F}_R$ are balanced.

5) *Comparison of RSP and Position Error Feedback:* Fig. 10(c) shows the direction of the force reflection calculated by the proposed RSP method and the conventional position error feedback



(a)

(b)

(c)

Fig. 10. Simulation results. (a) $\boldsymbol{F}_h$ and $\boldsymbol{F}_R$ $(= \boldsymbol{F}_{RG} + \boldsymbol{F}_{RE} - \boldsymbol{F}_{EF})$. (b) Components of $\boldsymbol{F}_R$: $\boldsymbol{F}_{EF}$, $\boldsymbol{F}_{RE}$, and $\boldsymbol{F}_{RG}$. (c) Direction of the force reflection sensed by the force sensor calculated by the proposed method and the conventional position error feedback method and the direction of link 3. When $t < 0.73$ s, the magnitude of F_{RE} is negligible so that the direction of F_{RE} is not meaningful for the RSP method.

method. When $t > 0.73$, since the first and the second joint are constrained, $\mathrm{IRS_E}$ is the direction of the third link. Therefore, the angle of the force reflection should be $q_1 + q_2 + q_3 - \pi$. While the proposed RSP method represents the accurate direction of the restriction space, the conventional position error feedback method does not. Note that the direction of $\boldsymbol{F}_{RE}$ is not meaningful since the magnitude of $\boldsymbol{F}_{RE}$ is negligibly small when $t < 0.55$, and the RSP method is not used when $0.55 < t < 0.73$ since the force sensor detects the obstacles.

TABLE III
COMPARISON OF FORCE REFLECTIONS USING THE PROPOSED RSP AND FORCE SENSORS

	RSP	Force Sensor
Pros	- No force sensor is required (less expensive). - Robust collision detection using only position sensors like encoders.	- The amplitude as well as the direction of the force can be detected.
Cons	- Only the direction of the force can be calculated.	- It is too expensive to distribute enough number of force sensors to detect every possible collision. - There is possibility to miss the collision with limited number of force sensors in unknown environment.

The results demonstrate that the proposed RSP framework successfully describes the IRS.

VI. Discussion and Concluding Remarks

This paper developed the position-sensor-based force reflection framework called the RSP method by using the concept of IRS. It describes the accurate restriction space at the slave side, regardless of the KDS conditions of multi-DOF bilateral teleoperation system. In unstructured environments, the slave robot can collide with unexpected obstacles at any point on the body of the slave robot and not just at the tip point of the end-effector, as typically assumed by bilateral teleoperation algorithms. As demonstrated in this paper, arbitrary collision may lead to grossly incorrect force feedback in the conventional position-sensor-based bilateral teleoperation algorithms. This paper proposed a simple and effective solution to this problem without the need for a force sensor.

The proposed method guarantees the accurate direction of the restriction space, but it cannot calculate the amplitude of the interaction force at the slave side. In the proposed method, the amplitude needs to be adjusted by multiplying the projected position error by a scalar force gain, as is typically done in position-error-based force feedback algorithm.

If the force sensors placed on the robot are able to detect every possible collision, including unexpected link collision, it promises results that are better than those obtained using position error force feedback method (see Table III for a comparison). The proposed method can also utilize information from force sensors, if the force sensor can detect the obstacle collision. Note that the proposed RSP method is developed to enhance the conventional position-sensor-based force feedback and not to replace the force-sensor-based force feedback.

The limitation of the proposed method is that it can detect the restriction space for the rigid obstacles since the constraint joint is selected using the joint angle errors. It is a tradeoff between the sensitivity of the force amplitude estimation and the cost of the force sensors.

From examples and simulation, we can conclude that the proposed RSP framework generates accurate direction of force reflection in various constraint conditions, while the conventional position-sensor-based force reflection method does not.

References

[1] D. A. Lawrence, “Stability and transparency in bilateral teleoperation,” *IEEE Trans. Robot. Autom.*, vol. 9, no. 5, pp. 624–637, Oct. 1993.

[2] R. Oboe and P. Fiorini, “A design and control environment for internet based telerobotics,” *Int. J. Robot. Res.*, vol. 17, no. 4, pp. 443–449, 1998.

[3] M. C. Cavusoglu, A. Sherman, and F. Tendick, "Design of bilateral teleoperation controllers for haptic exploration and telemanipulation of soft environment," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 641–647, Aug. 2002.

[4] T. Imaida, Y. Yokokohji, T. Doi, M. Oda, and T. Yoshikawa, "Ground-space bilateral teleoperation of EST-VII robot arm by direct bilateral coupling under 7-s time delay condition," *IEEE Trans. Robot. Autom.*, vol. 20, no. 3, pp. 499–511, Jun. 2004.

[5] B. Hannaford, "A design framework for teleoperators with kinesthetic feedback," *IEEE Trans. Robot. Autom.*, vol. 5, no. 4, pp. 426–434, Aug. 1989.

[6] R. J. Anderson and M. W. Spong, "Bilateral control of teleoperators with time delay," *IEEE Trans. Autom. Control*, vol. 34, no. 5, pp. 494–501, May 1989.

[7] G. Niemeyer and J. Slotine, "Stable adaptive teleoperation," *IEEE J. Ocean. Eng.*, vol. 16, no. 1, pp. 152–162, Jan. 1991.

[8] Y. Yokokohji and T. Yoshikawa, "Bilateral control of master–slave manipulators for ideal kinesthetic coupling-formulation and experiment," *IEEE Trans. Robot. Autom.*, vol. 10, no. 5, pp. 605–620, Oct. 1994.

[9] J. E. Colgate and J. M. Brown, "Factors affecting the z-width of a haptic display," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1994, pp. 3205–3210.

[10] R. J. Adams and B. Hannaford, "Stable haptic interaction with virtual environment," *IEEE Trans. Robot. Autom.*, vol. 15, no. 3, pp. 465–474, Jun. 1999.

[11] B. Hannaford and J. H. Ryu, "Time-domain passivity control of haptic interfaces," *IEEE Trans. Robot. Autom.*, vol. 18, no. 1, pp. 1–10, Feb. 2002.

[12] K. H. Zaad and S. E. Salcudean, "Transparency in time-delayed systems and the effect of local force feedback for transparent teleoperation," *IEEE Trans. Robot. Autom.*, vol. 18, no. 1, pp. 109–114, Feb. 2002.

[13] H. Kazerooni, T. Tsay, and K. Hollerbach, "A controller design framework for telerobotic systems," *IEEE Trans. Control Syst. Technol.*, vol. 1, no. 1, pp. 50–62, Mar. 1993.

[14] K. Kim, W. Chung, and S. Nam, "Accurate force reflection method for a multi-DOF haptic interface using instantaneous restriction space without a force sensor in an unstructured environment," *Adv. Robot.*, vol. 21, no. 1, pp. 87–104, 2007.

[15] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC, 1994.

[16] A. Liegeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, no. 12, pp. 868–871, Dec. 1977.

[17] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*. Reading, MA: Addison-Wesley, 1991.

Online Segmentation and Clustering From Continuous Observation of Whole Body Motions

Dana Kulić, Wataru Takano, and Yoshihiko Nakamura

***Abstract*—This paper describes a novel approach for incremental learning of human motion pattern primitives through online observation of human motion. The observed time series data stream is first stochastically segmented into potential motion primitive segments, based on the assumption that data belonging to the same motion primitive will have the same underlying distribution. The motion segments are then abstracted into a stochastic model representation and automatically clustered and organized. As new motion patterns are observed, they are incrementally grouped together into a tree structure, based on their relative distance in the model space. The tree leaves, which represent the most specialized learned motion primitives, are then passed back to the segmentation algorithm so that as the number of known motion primitives increases, the accuracy of the segmentation can also be improved. The combined algorithm is tested on a sequence of continuous human motion data that are obtained through motion capture, and demonstrates the performance of the proposed approach.**

***Index Terms*—Humanoid robots, incremental learning, learning from observation, motion segmentation and clustering.**

I. Introduction

Learning by observing and imitating humans is an attractive approach for improving the cognitive abilities of humanoid robots, thus taking advantage of the similarity in body structure between humanoids and humans. This process commonly includes the abstraction of human behaviors and their reuse for reproduction as well as human motion recognition and understanding. Many algorithms have been proposed in the literature [3], [4], most of which consider offline learning, where the data are collected, segmented, and sorted into the motion groups to be learned *a priori*. Following the offline learning stage, the robot can recognize and perform learned motions, but no subsequent learning takes place.

A robot operating in the human environment should be capable of continuous learning of demonstrated actions online during colocation and possible interaction with the human teacher. A robot with such abilities could adapt to changes in the environment requiring a change in how a task is executed by observing the changes in the human demonstration, autonomously detect that a novel task is being executed and should be learned, adapt to a change of interaction partner, and improve skill performance over time, thereby significantly improving robot adaptability and useability. The ability to learn continuously from online observation would enable robots to learn from nonexpert

Manuscript received June 4, 2008; revised February 4, 2009. First published July 28, 2009; current version published October 9, 2009. This paper was recommended for publication by Associate Editor F. Lamiraux and Editor J.-P. Laumond upon evaluation of the reviewers' comments. This work was supported by the Japanese Society for the Promotion of Science under Grant 18.06754 and by the Category S Grant-in-Aid for Scientific Research under Grant 15100002. This paper was presented in part at the 2008 IEEE International Conference on Robotics and Automation and in part at the 2008 IEEE/Robotics Society of Japan International Conference on Intelligent Robots and Systems.

D. Kulić was with the Department of Mechano-Informatics, University of Tokyo, Tokyo 113-8656, Japan. She is now with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: dkulic@ece.uwaterloo.ca).

W. Takano and Y. Nakamura are with the Department of Mechano-Informatics, University of Tokyo, Tokyo 113-8656, Japan (e-mail: takano@ynl.t.u-tokyo.ac.jp; nakamura@ynl.t.u-tokyo.ac.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2009.2026508