Robotica (2010) volume 28, pp. 267–277. © Cambridge University Press 2009
doi:10.1017/S0263574709990415

Virtual fixtures with autonomous error compensation for human–robot cooperative tasks

Raúl A. Castillo-Cruces* and Jürgen Wahrburg

Center for Sensor Systems (ZESS), University of Siegen, Germany

(Received in Final Form: July 23, 2009. First published online: September 2, 2009)

SUMMARY

This paper presents a control strategy for surgical interventions, applied on a human–robot cooperative system, which facilitates the sharing of responsibilities between surgeon and robot. The controller utilizes virtual fixtures to constrain the movements of the end-effector into a predefined path or region. Possible deviation error can be compensated in two different ways: (a) manual compensation and (b) autonomous compensation. With manual compensation, the system defines both virtual fixtures and error compensation directions, but the surgeon must apply manual forces himself/herself in order to generate end-effector motion. With autonomous compensation, a clear distribution of responsibilities between surgeon and robotic system is present, meaning the surgeon has complete control of the end-effector along the preferred directions, while the robot autonomously compensates for any deviation along the non-preferred directions.

1. Introduction

A natural and seamless integration of robotic systems in surgical interventions remains a significant challenge in robotic surgery. The interaction between surgeon and a robot should be as intuitive as possible; i.e., the surgeon should have complete control over the flow of the operation, while at the same time, the reliability and safety of the operation process must be ensured. A cooperative interaction, where the surgeon can grab and move the robot's end-effector directly, is a step towards achieving such integration. This paper discusses the concept of virtual fixtures (VFs), i.e., where defined fixtures are used to keep the movements of the end-effector within a predefined path or region in which the specific task is to be executed.

Robotic systems operating in collaboration with humans have been an active topic of research during the last two decades. Various control systems have been proposed by Kazerooni *et al.*[16,17] to generate the motion based on the intentional force. Cooperative tasks for industrial applications have also been proposed; examples include the cooperative manipulation of objects[18,19] and peg-in-hole tasks.[20] In surgical robotics, cooperative remains a key control topic of ongoing research. Work in this area has previously been conducted in relation to robotic systems, examples of which include the JHU Steady Hand Robot at the Johns Hopkins University, designed for microsurgical procedures,[1,8] an the special-purpose hands-on robot, Acrobot, applied in knee surgery[5] and minimal invasive neurosurgery,[14] that uses active constraints to limit the motion to a predefined region. In these system types, the robot is an active mechanism that behaves as a passive manipulator, by means of an admittance controller. This allows the movement of the end-effector to be expressed as a function of the user's applied force, measured by a force–torque (FT) sensor mounted at the end-effector of the robot. VFs for such robots have been developed by Bettini *et al.*[2,3] and Li *et al.*[9,10] Prade *et al.*[25] advanced this research by applying VFs to the surgical training environment. Two types of force fields were defined within the VFs: attractive and reactive. In this work only translation was considered. Other system types, based on passive mechanisms, have also been implemented; examples include Cobots,[11] which allow the coupling of degrees of freedom (DOFs) by means of variable transmissions, and PADyC,[12] which consists of a two freewheels mounted in opposite directions and associated with two motors at each joint in order to provide the different desired constraint effects. The working principle of these passive devices is considered to be out of the scope of this work and, as such, will not be discussed further in this paper.

The modular interactive computer-assisted surgery (modiCAS) project based at the Center of Sensor System (ZESS), in Siegen, Germany, focused on providing an integrated solution to different surgical problems through integrating a navigation system and a robotic arm with hands-on capabilities.[7] One of the projects main concerns was the development of a virtual constrained hands-on interface for cooperative tasks. The VFs developed here are of the admittance type and are based predominantly on the work done by Bettini *et al.*,[3] yet with additional modifications in the manner of how deviation errors are compensated. In Bettini's work, the virtually defined preferred directions and the error compensation directions are combined into a new preferred direction which is then applied to the admittance controller. This approach implies a manual compensation, as the surgeon-applied forces are always required to compensate the error. In our approach, the error compensation is autonomously controlled, independent of the VFs. A clear distribution of responsibilities between surgeon and robotic system is present, meaning that the surgeon has complete

* Corresponding author. E-mail: castillo-cruces@zess.uni-siegen.de

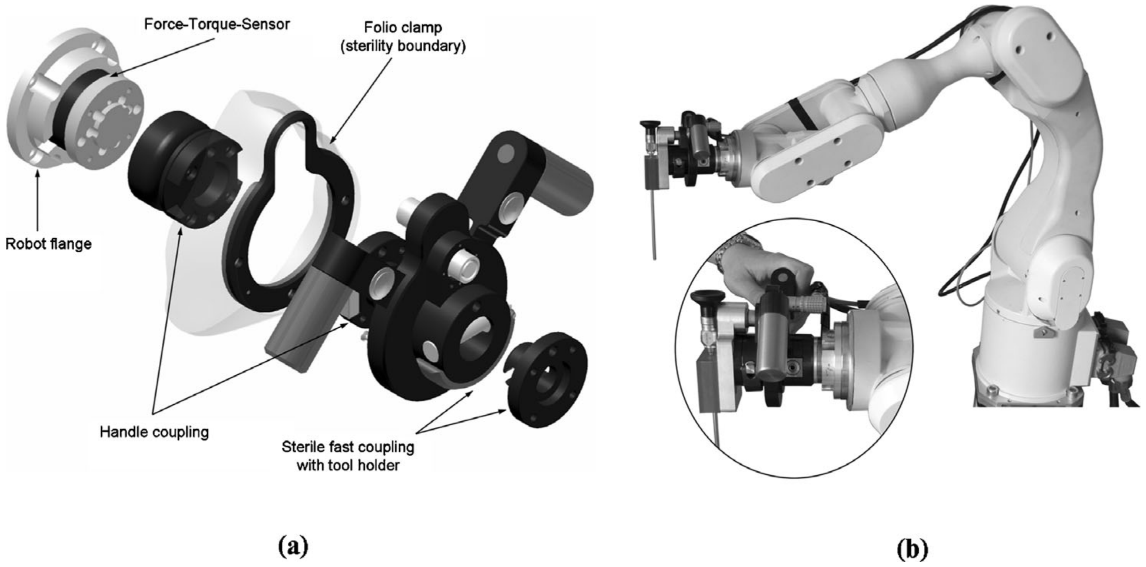

Fig. 1. (a) Handle system with rapid tool-exchange mechanism. (b) Robotic system with hands-on capability.

control of the end-effector along the preferred directions, while the robot autonomously compensates for any deviation along the non-preferred directions.

In the following sections, further insight about the VFs with manual and autonomous compensation is given. Section 2 describes the hands-on interface used within the surgical system. In Section 3, the determination of VFs and their corresponding subspaces for preferred and non-preferred directions is explained. In Section 4, the admittance controller with manual and autonomous error compensation is introduced. Both approaches are discussed in this section. An experimental setup is presented in Section 5 in order to evaluate the performance of both approaches. The results are discussed in Section 6, and finally the conclusions are presented in Section 7.

2. Hands-On Interface

The modiCAS system is equipped with a hands-on interface that consists of a six-DOF FT sensor mounted on the robot's end-effector (see Fig. 1). The force applied to the tool is measured by the FT sensor mounted just behind the holding mechanism. At the software level, the data acquisition and processing of the incoming signals, such as voltage-to-force transformation, mean value calculation, drift compensation, filtering, and compensation of the influence of gravity on the tool when its orientation is changed, are executed.

3. VF description

The VFs, also found in the literature as synthetic fixtures,[23] virtual mechanisms,[24] and virtual tools,[19] are software-generated force and position signals applied to human operators via robotic devices. They help humans perform robot-assisted manipulation tasks by limiting movement into restricted regions and/or influencing movement along desired paths.[1] The VF is essentially the separation of the three-dimensional working space into two complementary subspaces, one containing all the preferred directions and the other containing the non-preferred ones.[13] A VF can be composed of one or more directions, the combination of which permits different isotropic movements. Each of these directions is hereby defined as single *virtual unit*.

We first distinguish between two types of virtual units, namely, the linear virtual unit $\mathbf{l}$ and the angular virtual unit $\boldsymbol{\alpha}$. The linear virtual unit is a vector in $\Re^3$ that defines a specific direction in the Cartesian space along which the displacement of the robot's end-effector is permitted. The angular virtual unit, also a vector in $\Re^3$, specifies an arbitrary axis in the Cartesian space, about which a rotation of the end-effector is possible. We now define a subspace U of $\Re^6$, which contains all preferred directions for both translation and rotation. Let $\boldsymbol{S}^l$ and $\boldsymbol{S}^\alpha$ be two subsets of $\Re^3$, comprising the linear independent set of vectors that span U for position and orientation respectively:

$$\begin{aligned} \boldsymbol{S}^l &= \{\mathbf{l}_1, \dots, \mathbf{l}_p\},\\ \boldsymbol{S}^\alpha &= \{\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_k\}, \end{aligned} \tag{1}$$

where $p,\ k \leq 3$. Now, let $\boldsymbol{D}$ denote the $6 \times (p+k)$ instantaneous preferred direction matrix comprising the elements of S^l and S^α,

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{S}^l & 0_{3\times k} \\ 0_{3\times p} & \boldsymbol{S}^\alpha \end{bmatrix} = \begin{bmatrix} (\mathbf{l}_1|\dots|\mathbf{l}_p) & 0_{3\times k} \\ 0_{3\times p} & (\boldsymbol{\alpha}_1|\dots|\boldsymbol{\alpha}_k) \end{bmatrix}, \tag{2}$$

such that

$$\boldsymbol{P}_U = \mathrm{Ran}(\boldsymbol{D}) = \boldsymbol{D}(\boldsymbol{D}^T\boldsymbol{D})^{-1}\boldsymbol{D}^T. \tag{3}$$

The orthogonal projection $\boldsymbol{P}_U$ acts as the identity of U; i.e., any vector $\mathbf{x}$ in this subspace has $\boldsymbol{P}_U\mathbf{x} = \mathbf{x}$. The subspace U is the exact range of this projection. Furthermore, there exists an orthogonal complementary subspace V that contains all the non-preferred directions (see Fig. 2). Every vector $\mathbf{x}$ in V has $\boldsymbol{P}_U\mathbf{x} = 0$. This is the *null space*, also called the *kernel* of the projection. Its corresponding projection operator is given

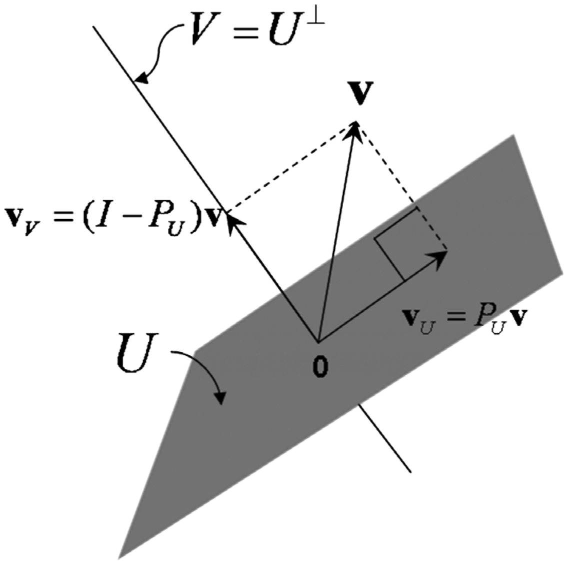
$V = U^{\perp}$
$\mathbf{v}$
$\mathbf{v}_V = (I - P_U)\mathbf{v}$
U
0
$\mathbf{v}_U = P_U\mathbf{v}$

Fig. 2. Projection onto the subspaces U and V of preferred and non-preferred directions of a virtual plane, respectively.

by

$$\boldsymbol{P}_V = \mathrm{Ker}(\boldsymbol{D}) = \boldsymbol{I} - \boldsymbol{P}_U. \tag{4}$$

The resulting projections, $\boldsymbol{P}_U$ and $\boldsymbol{P}_V$, create a mechanism which can be used within the system control law to determine whether the applied forces at the end-effector are pointing in a preferred direction or not. These measured forces are expressed directly in the robot's end-effector coordinate system. This means the virtual units in $\boldsymbol{D}$ must also be defined with respect to this frame. Nevertheless, if the robot kinematic and the relationship of the different possible reference frames (with respect to the robot base frame) are well known, it is then possible to define each virtual unit with respect to one of the different coordinate systems. However, this implies that the calculation of $\boldsymbol{D}$ must be executed at every cycle time.

4. Admittance Controller

The control strategy for the cooperative mode essentially consists of two control loops: an inner velocity control loop at joint level and an outer admittance controller that modulates the end-effectors' linear and angular velocities as a function of the applied forces. These velocities are then mapped to the joint space and further forwarded to the inner loop (see Fig. 3). The general form of an admittance controller is

$$\dot{\mathbf{x}} = c\gamma, \tag{5}$$

where $\dot{\mathbf{x}} = [\,\dot{\mathbf{p}}^T\ \boldsymbol{\omega}^T\,]^T$ represents the linear and angular velocity of the end-effector. The scalar admittance gain, $c \in [0,1]$, establishes the compliance level of the system. The vector $\boldsymbol{\gamma} = [\,\mathbf{f}^T\ \boldsymbol{\tau}^T\,]^T$ contains the forces/moments applied at the end-effector.

In Eq. (5), the robot compliance has an isotropic behavior, since the gain c affects all directions in the task space in the same way. The objective of the VFs is to provoke anisotropic behavior with different levels of compliance between the preferred directions and the non-preferred ones. Therefore, the projection operators expressed in Eqs. (3) and (4) are incorporated into Eq. (5), together with the additional gains $c_U, c_V \in [0, 1]$. This leads into the following expression:

$$\dot{\mathbf{x}} = c_U(\boldsymbol{P}_U\boldsymbol{\gamma} + c_V\boldsymbol{P}_V\boldsymbol{\gamma}) = c_U(\boldsymbol{P}_U + c_V\boldsymbol{P}_V)\boldsymbol{\gamma}. \tag{6}$$

The gain c_V only regulates the amount of compliance of V. The resulting effect is a *guidance VF* that helps the user to move the end-effector along a desired path or surface defined by U. Different values of c_V will influence the level of guidance. If $c_V = 0$, the subspace V is completely eliminated;

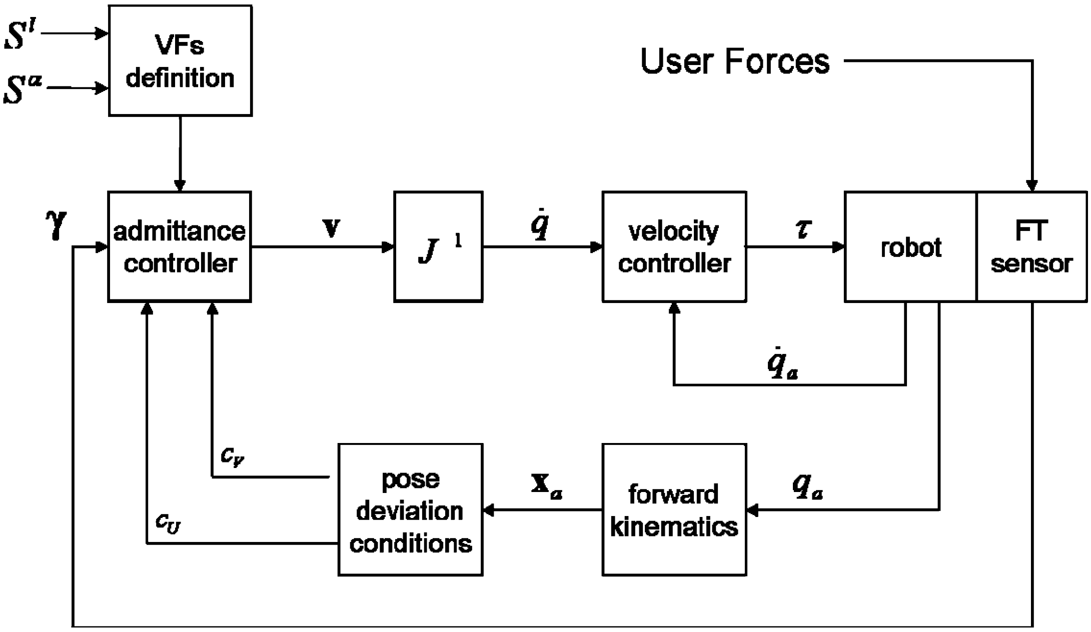

Fig. 3. Control loop for cooperative robot system.

https://doi.org/10.1017/S0263574709990415

i.e., a hard guidance level along U is present. At the other extreme, where $c_V = 1$, there is no distinction between preferred and non-preferred directions; i.e., no guidance is present. Values in between both extremes will create the effect of soft guidance. The global compliance of the system against applied force can be regulated by means of c_U. This is useful when defining boundaries along preferred directions.

The admittance gains (c_U, c_V) are scalars that affect all Cartesian components of the end-effector equally. These gains can be substituted by a matrix form $(\boldsymbol{C}_U, \boldsymbol{C}_V)$, where each matrix is a 6×6 diagonal matrix containing the different components of velocity expressed in the tool frame:

$$C = \begin{bmatrix} c_x & 0 & 0 & 0 & 0 & 0 \\ 0 & c_y & 0 & 0 & 0 & 0 \\ 0 & 0 & c_z & 0 & 0 & 0 \\ 0 & 0 & 0 & c_\alpha & 0 & 0 \\ 0 & 0 & 0 & 0 & c_\beta & 0 \\ 0 & 0 & 0 & 0 & 0 & c_\varphi \end{bmatrix}, \tag{7}$$

where c_x, c_y, c_z and $c_\alpha, c_\beta, c_\varphi$ are position and orientation components, respectively. The matrices $\boldsymbol{C}_U, \boldsymbol{C}_V$ allow the establishment of an appropriate level of guidance to each velocity component. This is useful, for example, where the compliance behavior of the translational and rotational components have to be controlled independent of each other.

4.1. Deviation error

Expression (6) allows the user to move the end-effector in preferred directions despite its actual position and orientation. However, there is normally a desired reference pose (position and orientation) to which the VF is referred. Any deviation of the end-effector from the reference along the non-preferred directions is considered an error and has to be compensated for. This error compensation is regarded as a reaching target problem acting on V. The reference target pose defines the position and orientation that is to be reached and maintained. This pose is also the center reference point of the VF definition. If the target pose varies within the time, the VF moves with it.

Let $\mathbf{e}_p$ and $\mathbf{e}_r$ be the position error and orientation error vectors respectively. These vectors quantify the deviation of the actual tool center point (TCP) pose ${}^{\text{Base}}\boldsymbol{T}_{\text{TCP}}$ from the desired target pose ${}^{\text{Base}}\boldsymbol{T}_{\text{TAR}}$.† Both homogenous transformations have the form

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{R}_{3\times3} & \mathbf{p} \\ 0_{1\times3} & 1 \end{bmatrix}, \tag{8}$$

where $\boldsymbol{R}$ is the rotation matrix and $\mathbf{p} = (x\ y\ z)^T$ is the position vector. It is assumed that target pose $\boldsymbol{T}_{\text{TAR}}$ is already defined with respect to the base frame and, by using the robot kinematics, $\boldsymbol{T}_{\text{EE}}$ can be calculated in a straightforward manner as follows (see Fig. 4):

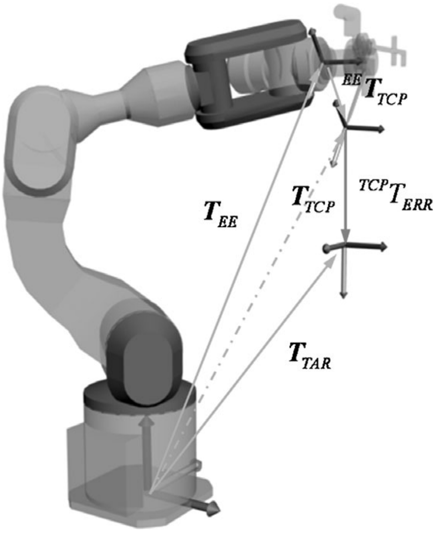

Fig. 4. System reference frames.

$$\boldsymbol{T}_{\text{TCP}} = \boldsymbol{T}_{\text{EE}}{}^{\text{EE}}\boldsymbol{T}_{\text{TCP}}, \tag{9}$$

where ${}^{\text{EE}}\boldsymbol{T}_{\text{TCP}}$ is the constant homogeneous transformation matrix from the TCP to the robot's end-effector and $\boldsymbol{T}_{\text{EE}}$ defines the end-effector with respect to the robot base frame and is calculated using the forward kinematics relationship of the robot arm.

The translational error $\mathbf{e}_p$ is calculated by subtracting the translational vector of the homogenous transformations $\boldsymbol{T}_{\text{TCP}}$ and $\boldsymbol{T}_{\text{TAR}}$ as follows:

$$\mathbf{e}_p = \mathbf{p}_{\text{TAR}} - \mathbf{p}_{\text{TCP}}. \tag{10}$$

In the case of the orientation error, $\mathbf{e}_r$ represents the axis of rotation in which the error is to be compensated and its norm represents the angle of rotation. This can be calculated using quaternion representation which (contrary to other notations such as the Euler angles) is numerically stable and free of singularities.[22] The error in terms of rotation matrices is defined as

$$\boldsymbol{R}_{\text{ERR}} = \boldsymbol{R}_{\text{TAR}}^{-1}\boldsymbol{R}_{\text{TCP}} = \boldsymbol{R}_{\text{TAR}}^{T}\boldsymbol{R}_{\text{TCP}}. \tag{11}$$

Applying quaternion representation, the orientation error can be expressed as

$$\begin{aligned} \phi_{\text{ERR}} &= \bar{\phi}_{\text{TAR}}\phi_{\text{TCP}} \\ &= \begin{bmatrix} \eta_{\text{TAR}} & \boldsymbol{\mu}_{\text{TAR}}^{T} \\ -\boldsymbol{\mu}_{\text{TAR}} & \eta_r \boldsymbol{I}_{3\times3} - \boldsymbol{S}\left(\boldsymbol{\mu}_{\text{TAR}}\right) \end{bmatrix} \begin{bmatrix} \eta_{\text{TCP}} \\ \boldsymbol{\mu}_{\text{TCP}} \end{bmatrix}. \end{aligned} \tag{12}$$

Since (12) is expressed with unit quaternions, an axis of rotation $\boldsymbol{\varepsilon}'$and an angle of rotation φ can be derived through the expression $\phi = [\cos\varphi\ \boldsymbol{\varepsilon}'\ \sin\varphi]^T$. Finally the orientation error $\mathbf{e}_r$ is calculated as follows:

$$\mathbf{e}_r = \varphi\boldsymbol{\varepsilon}'. \tag{13}$$

†In the remaining of this paper, for the sake of notation simplicity, when the reference frame is the base of the robot, the upper prefix of the transformation is omitted, e.g., $\boldsymbol{T}_{\text{EE}} = {}^{\text{Base}}\boldsymbol{T}_{\text{EE}}$. The same applies for rotation matrices $\boldsymbol{R}$ and position vectors $\mathbf{p}$.

https://doi.org/10.1017/S0263574709990415

The orientation error $\mathbf{e}_r$ is a vector indicating the axis of rotation, and $\|\mathbf{e}_r\|$ describes the magnitude of the rotation about this axis.

Any deviation from $\boldsymbol{T}_{\text{TAR}}$, within the subspace U, is not considered an error, since it occurs along a preferred direction. However, deviations along V do represent an error. Thus, only the error along the non-preferred direction is calculated using the projection operator P_V:

$$\mathbf{e}_V = \boldsymbol{P}_V \begin{bmatrix} \mathbf{e}_p \\ \mathbf{e}_r \end{bmatrix} = \begin{bmatrix} \mathbf{e}_{Vp} \\ \mathbf{e}_{Vr} \end{bmatrix}. \tag{14}$$

4.2. Manual error compensation

The manual error compensation is based on the previous work by Bettini *et al.*[2] and relies on the input forces applied by the user to compensate for the deviation errors. Essentially, in the presence of an error, the virtual preferred directions are redefined to consider such error, creating a new VF that corrects for the error. Thus, a new instantaneous preferred direction $\boldsymbol{D}_e$ is defined. This considers the directions required to compensate any translational and rotational deviation from U:

$$\boldsymbol{D}_e = (1 - k_d)\boldsymbol{P}_U\boldsymbol{\gamma} + k_d\,\|\boldsymbol{\gamma}\|\,\mathbf{e}_V, \quad 0 < k_d < 1. \tag{15}$$

The combination of the applied forces $\boldsymbol{\gamma}$ pointing in the preferred direction (obtained by means of the projection operator $\boldsymbol{P}_U$) and the error vector $\mathbf{e}_V$ yields a virtual direction that returns the TCP to the subspace U. The constant k_d regulates how strong the influence of the error vector $\mathbf{e}_V$ is in the new virtual preferred direction, i.e., how quickly the error is compensated. When the TCP lies within the subspace U, the second term of Eq. (15) vanishes. Now, using the new preferred direction $\boldsymbol{D}_e$ to recalculate the projection operators (3) and (4), and introducing them into the control law of Eq. (6), results in a law equivalent to a pure subspace motion constraint:[4]

$$\dot{\mathbf{x}} = c_U(\boldsymbol{P}_{Ue} + c_V\boldsymbol{P}_{Ve})\boldsymbol{\gamma}, \tag{16}$$

where

$$\boldsymbol{P}_{Ue} = \text{Ran}(\boldsymbol{D}_e) = \boldsymbol{D}_e\big(\boldsymbol{D}_e^T\boldsymbol{D}_e\big){-}1\boldsymbol{D}_e^T,$$
$$\boldsymbol{P}_{Ve} = \text{Ker}(\boldsymbol{D}_e) = \boldsymbol{I} - \boldsymbol{P}_{Ue}.$$

Notice however that the definition of the new preferred direction to compensate the error is not sufficient to guarantee that the error is minimized; i.e., the surgeon is still able to apply a force in the negative error direction which would increase the error. Therefore, the applied forces pointing towards the negative error direction are filtered out using the following condition:

$$\boldsymbol{\gamma}_e = \begin{cases} \boldsymbol{P}_U\gamma, & \mathbf{e}_V\cdot\gamma < 0, \\ \gamma & \text{otherwise}. \end{cases} \tag{17}$$

Substituting $\boldsymbol{\gamma}$ with $\boldsymbol{\gamma}_e$ in Eq. (16) guarantees that only applied forces that really compensate the error are effective without affecting the forces pointing in the preferred directions.

4.2. Autonomous error compensation

The error compensation presented in Eq. (16) depends on the applied forces $\boldsymbol{\gamma}$. This means the error is compensated for only if the user applies a force in the $\mathbf{e}_V$ direction; otherwise the error remains present. This compensation occurs intuitively in some cases. For example, in translation movements along a predefined direction, the user automatically compensates for any possible error by pushing in the path direction; however, there may be cases in which no compensation is induced at all, such as when making pivot rotations about the TCP at a constant target position. Ideally, the position remains fixed, but in reality slight deviations in the position might occur. Although the error is detected by the system and $\boldsymbol{D}_e$ is defined, the compensation effect occurs only after the proper force is applied, though the act of rotating rather demands applied moments than forces. Consequently, the error remains present and even increases before the user can observe it and apply a compensation force.

Nevertheless, by adding one term to Eq. (6), an automatic compensation of the deviation error can be achieved independent of the applied forces without affecting the VFs. The new expression looks as follows:

$$\dot{\mathbf{x}} = c_U(\boldsymbol{P}_U + c_V\boldsymbol{P}_V)\boldsymbol{\gamma} + k_V\mathbf{e}_V. \tag{18}$$

The error term in Eq. (18) does not depend on the input forces $\boldsymbol{\gamma}$ anymore. Notice that rather than combining the error vector with the VF definition D as in manual compensation (Eq. (15)), it is compensated with a simple linear control law ($k_V\mathbf{e}_V$). The gain k_V modulates the rate of response of the compensation. Through this approach, the surgeon has still complete control inside U, while the robot assures that the reference target pose is maintained.

5. Experimental Evaluation

The experimental evaluation of the VFs presented below was conducted using the modiCAS system. All tests were executed with the six-DOF PA10-6C robot arm, with the mini45 FT sensor from ATI Industrial Automation (Apex, NC, USA) mounted on the robot's end-effector. The inner velocity control loop was directly implemented in the servo driver of the robot system and ran with a frequency of 1538 Hz. The outer admittance control loop, running at 200 Hz, was implemented on the real-time (RT) target running the LabVIEW-RT module. The position and orientation errors were calculated independently as follows:

$$\begin{aligned} \|\mathbf{e}_p\| &= \sqrt{\mathbf{e}_{xpV}^2 + \mathbf{e}_{ypV}^2 + \mathbf{e}_{zpV}^2}, \\ \|\mathbf{e}_r\| &= \sqrt{\mathbf{e}_{xrV}^2 + \mathbf{e}_{yrV}^2 + \mathbf{e}_{zrV}^2}. \end{aligned} \tag{19}$$

The following experiments compared the behavior of both manual and autonomous compensation. Firstly, the behavior of manual error compensation was analyzed upon changes of the coefficient k_d in Eq. (15). Furthermore, a comparison between manual compensation and autonomous

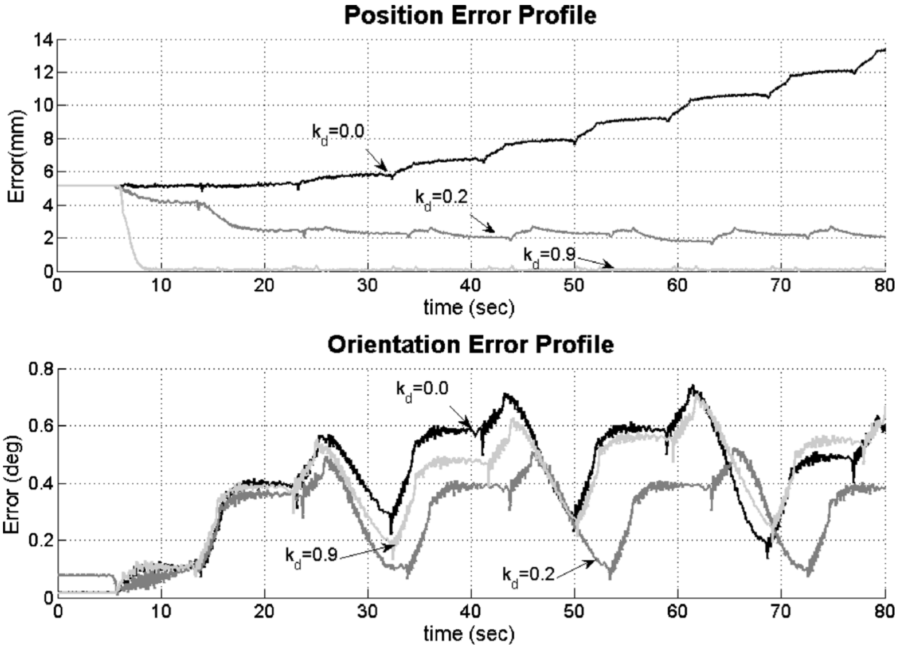

Fig. 5. Influence of gain k_d on the error norm of manual compensation while moving the end-effector along the y-axis with respect to world coordinates.

compensation was conduced for both a translational VF and a rotational VF. A hard guidance level, $c_V = 0$, was used during manual compensation, since minimization of the deviation from the actual path was sought. Any soft guidance level, $0 < c_V < 1$, would allow deviations along the non-preferred subspace.

5.1. Manual error compensation

The following experiment analyzed the behavior of manual compensation (expression (16)) in the presence of position deviation. A target position $\boldsymbol{T}_{\text{TAR}}$ was defined at $\mathbf{p}_{\text{TAR}} = [\,621\ 0\ 548\,]^T$ (mm) and $\mathbf{rpy}_{\text{TAR}} = [\,-90\ 0\ -90\,]^T$ (deg.), where "rpy" denotes the roll–pitch–yaw notation of the orientation of the TCP with respect to the robot base frame. The TCP was defined exactly at the end-effector of the robot; i.e., ${}^{\text{EE}}\boldsymbol{T}_{\text{TCP}}$ contains an identity rotation matrix and position coordinates equal to zero. The robot's end-effector was located at $\mathbf{p}_{\text{EE}} = [\,621\ 0\ 554\,]^T$ and $\mathbf{rpy}_{\text{EE}} = \mathbf{rpy}_{\text{TAR}}$, which represented a deviation of 6 mm along z-axis from $\boldsymbol{T}_{\text{TAR}}$. A VF was created in order to move the end-effector in the preferred direction along y-axis with respect to the base frame while keeping orientation constant, i.e., $S^l = \{\boldsymbol{R}^T_{\text{TCP}}\mathbf{l}_1\}$ and $S^\alpha = \{0\}$, where $\mathbf{l}_1 = [\,0\ 1\ 0\,]^T$. The experiment consisted of moving the end-effector back and forth by hand along the preferred direction, which was already known for the user. Several trials were executed with different values of gain $k_d \in [0{,}1]$ for manual compensation. Figure 5 shows the influence of k_d on Eq. (15). No error compensation occurred when $k_d = 0$. Increasing the value of k_d yielded a faster compensation of the position error. Notice that the end-effector orientation deviated considerably from the desired one despite the values of k_d. Although the orientation components were not influenced by k_d during translational movements, the position error was reduced when incrementing the value of k_d. A notable enhancement in performance occurred for values up to $k_d = 0.9$, while higher values produced no significant improvement.

5.2. Manual compensation versus autonomous compensation

5.2.1. Translational case. The setup of this experiment was similar to the one just explained above in Section 5.1. The user was asked to move the tool back and forth along the preferred direction on the y-axis. Notice that in this experiment, the initial TCP position was equal to the target position, i.e., $\boldsymbol{T}_{\text{TCP}} = \boldsymbol{T}_{\text{TAR}}$. The objective was to analyze the efficiency of manual and autonomous controllers (expressed in Eqs. (16) and (18), respectively) to keep the error at minimum along the non-preferred directions while moving along a preferred direction. Therefore a translational VF was defined as follows: ($S^l = \{\boldsymbol{R}^T_{\text{TCP}}\mathbf{l}_1\}$ and $S^\alpha = \{0\}$, where $\mathbf{l}_1 = [\,0\ 1\ 0\,]^T$). The gain $k_d = 0.9$ of the manual controller was chosen after the analysis of the experiment presented in Section 5.1. The gain $k_v = 5$ for the autonomous controller was experimentally determined. Additionally, an attempt with no error compensation was included to provide an additional benchmark for the comparison of the results.

The position behavior, orientation behavior, and instantaneous quadratic error norm are presented in Figs. 6–8, respectively. Both the manual and autonomous controllers exhibited a similar behavior relating to the position; i.e., both had a position error of the same order. However, it is in the orientation error where a great difference arose. While the orientation error behavior of the manual compensation looked very similar to the case in which no compensation at all occurred, the error was strongly reduced when applying autonomous error compensation. Table I shows the mean position and orientation error for the three cases: no compensation, manual compensation, and autonomous compensation.

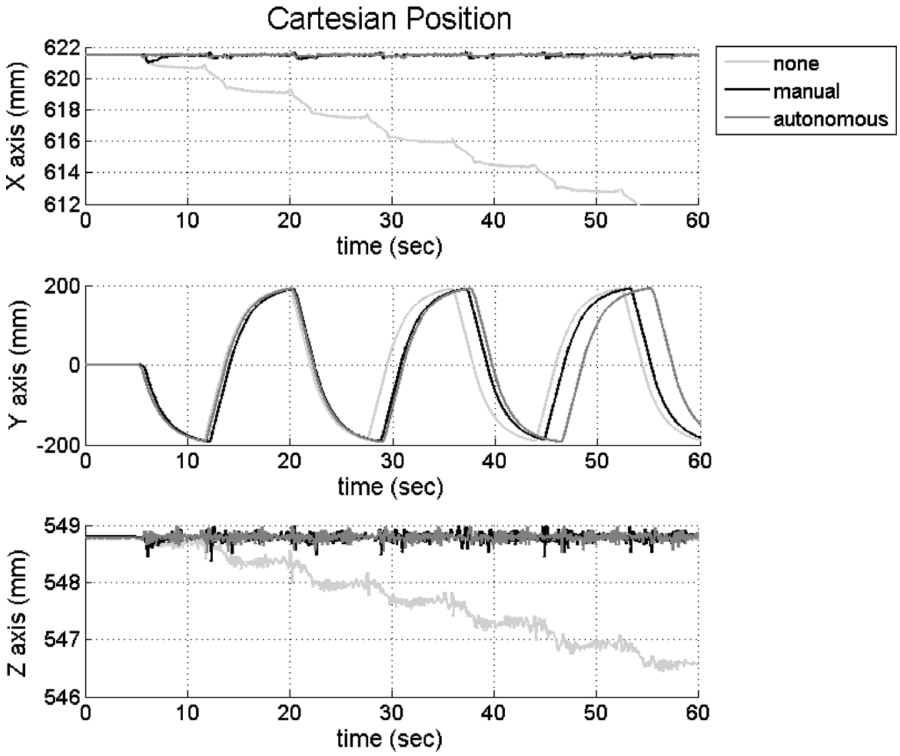

Fig. 6. End-effector position while moving along the y-axis with respect to the world coordinates. (b) End-effector orientation while moving along the y-axis with respect to the world coordinates.

5.2.2. Rotational case. This experiment evaluated the error compensation when a rotational VF was applied. Three responses were analyzed: no compensation, manual compensation (Eq. (16)), and autonomous compensation (Eq. (18)). The rotational VF consisted of a pivot rotation of the end-effector 360° back and forth while keeping a constant inclination ($\beta = 45°$) of the tool with respect to the rotation axis. This axis was positioned at a specific point in the space parallel to the z-axis of the robot base frame. In this experiment, the TCP was located at the tool's tip

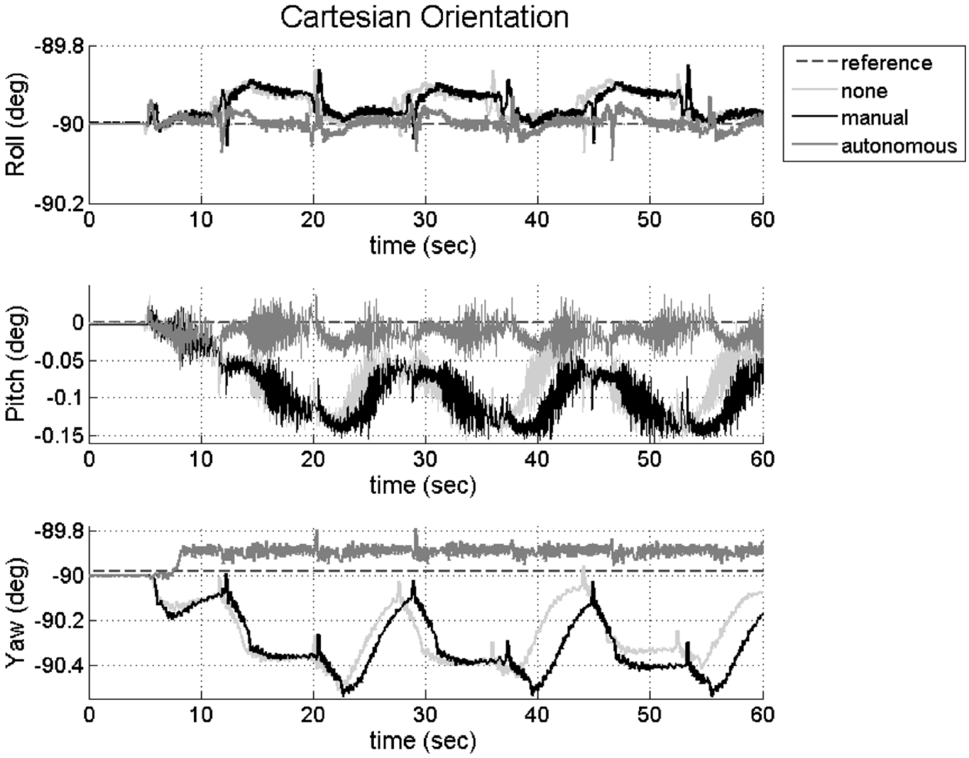

Fig. 7. End-effector orientation while moving along the y-axis with respect to the world coordinates.

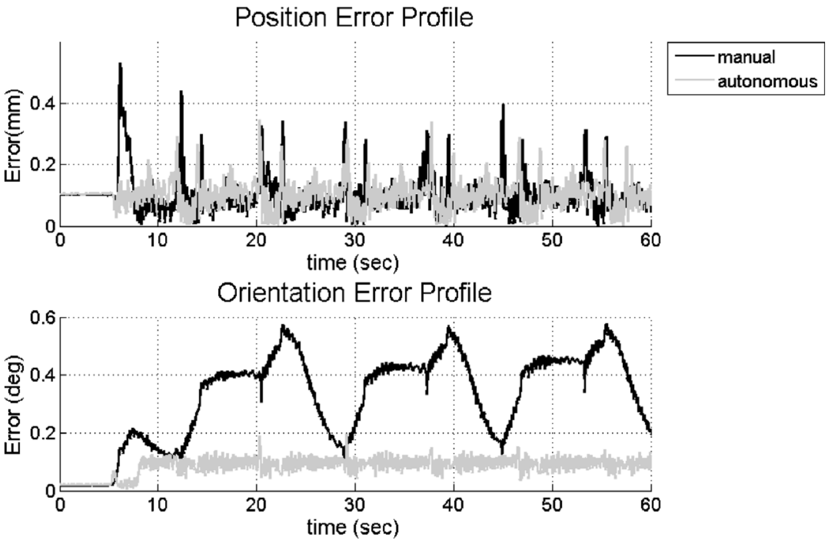

Fig. 8. Error profile of manual and autonomous error compensation while moving along the *y*-axis with respect to the world coordinates.

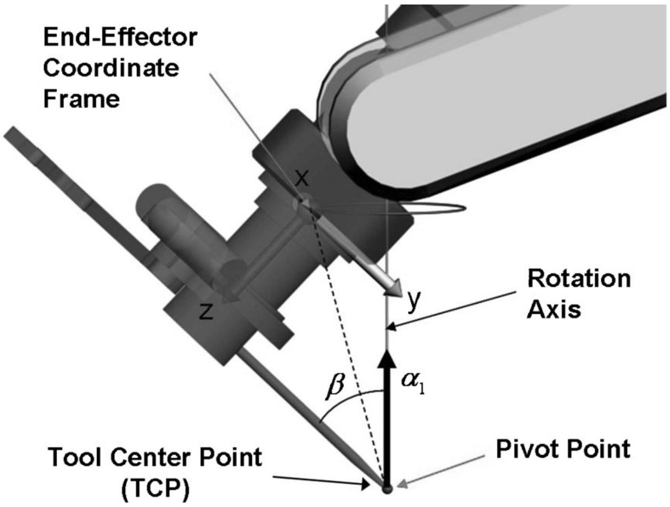

Fig. 9. Experimental setup for rotational case.

(see Fig. 9). The transformation ${}^{EE}\boldsymbol{T}_{\text{TCP}}$ from TCP to end-effector was defined by means of a calibration procedure using the navigation system.[15] A VF was created in order to rotate the tool around the *z*-axis with respect to the base frame while keeping the relative angle relationship β constant, i.e., $S^l = \{0\}$ and $S^\alpha = \{\boldsymbol{R}^T_{\text{TAR}}\boldsymbol{\alpha}_1\}$, where $\boldsymbol{\alpha}_1 = [0\;0\;1]^T$. Contrary to the experiment in Section 5.2, $\boldsymbol{R}_{\text{TAR}}$ was used instead of $\boldsymbol{R}_{\text{TCP}}$ to define the rotational VF. The latter rotation matrix would mean that the VF would stay constant with respect to the tool coordinates, while this experiment required a VF that stays constant with respect to the base coordinates.

Table I. Mean value of position and orientation error of translational VF.

Control	Mean position error (mm)	Mean orientation error (deg.)
None	6.4582	0.3044
Manual	0.1006	0.3492
Autonomous	0.1012	0.0831

The results of this experiment revealed that the decoupling of position and orientation during manual compensation occurred in a similar way as in the translational case. During the experiment, orientation movements were executed and the position was kept constant. However, Fig. 10 shows that in the cases of no compensation and manual compensation the position of the TCP presented strong deviations from the reference pivot point, both of them having the same patron. Autonomous compensation, on the contrary, reduced the position error. The orientation error was reduced in a similar manner in both manual and autonomous compensation, while the error continuously increased when no compensation was executed. Finally, the quadratic error norm plotted in Fig. 11 corroborated the behavior just explained above. The corresponding mean errors are presented in Table II.

6. Discussion

The two methods presented in this paper were compared for error compensation during cooperative manipulation of the tool along virtually constrained subspaces: the manual compensation and autonomous compensation. The philosophy behind manual compensation states that the user is the only one able to generate any kind of motion, while the robotic system is more like a passive system with the sole job of constraining the possible movements into an allowed subspace. This is done by the so-called VFs. In the presence of a deviation error, the manual controller redefines such VFs to include the direction needed to compensate for such an error, thus having one new direction that guarantees the required error compensation. The experiments presented above demonstrate that the compensation takes place as long as an input force induced by the user is applied. Unfortunately, the translational and rotational movements are not directly coupled with each other. This means that when performing one of these two types of movements, any deviation error appearing on the other type of movement may not necessarily be compensated for. The reason is that due to the nature of the movement, despite the error being

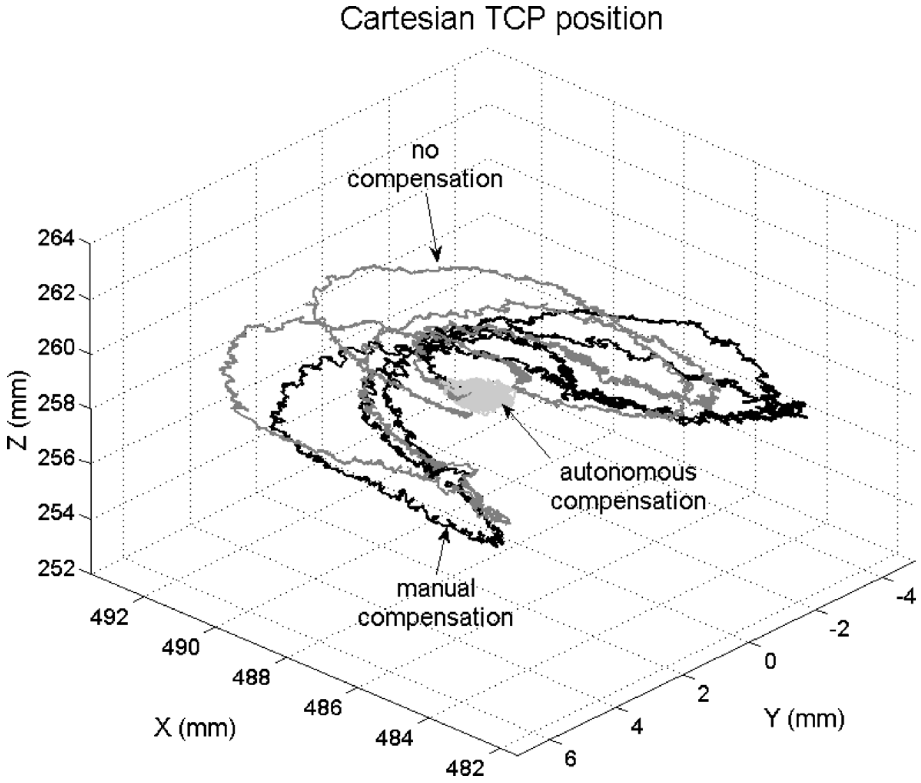

Fig. 10. Tool tip position in three-dimensional while pivot rotation about the z-axis.

detected by the controller, and a new VF being redefined to compensate for the error, the user may not realize that the generation of such motions is required. Thus, if no movement is induced by the user in such direction, the compensation does not occur.

The concept of VFs with autonomous error compensation is then proposed to deal with this drawback of manual compensation. The main idea is to give the robotic system the responsibility of error compensation, while the user keeps complete control inside the allowed subspace. This has a disadvantage in that the robot itself is able to generate motion which may be undesired for the sake of safety. For instance, suppose that a VF is defined by mistake on a target pose $\boldsymbol{T}_{\mathrm{TAR}}$ which is far away from the end-effector's current position, and the autonomous compensation is active. At the moment when the user activates the hands-on interface, the robot would automatically begin to compensate for the error, producing an unexpected and even more undesirable movement which could lead to serious consequences. This is not the case if manual compensation is active, where the controller redefines the VF and the robot waits until the user compensates the error himself/herself. This is intrinsically

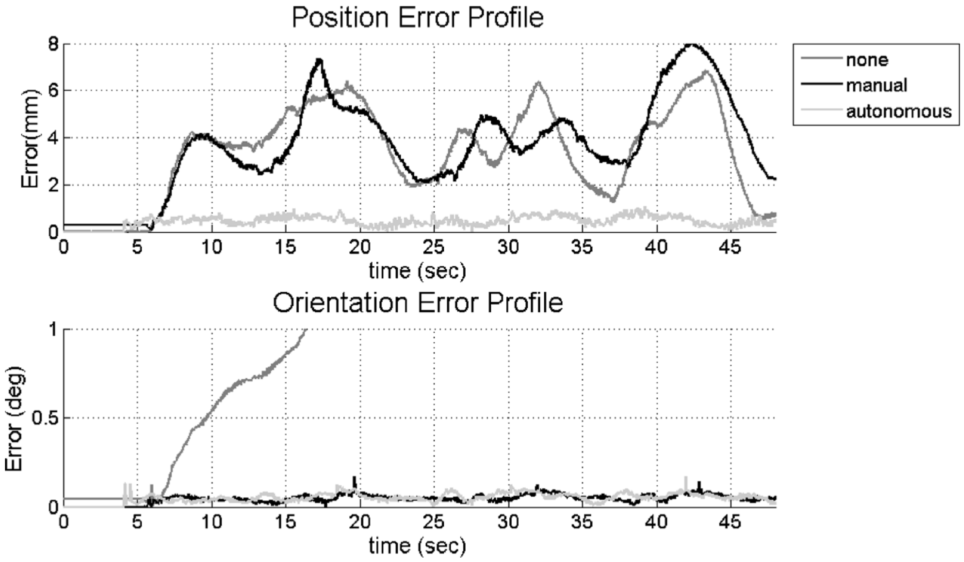

Fig. 11. End-effector error profile while pivot rotation about the z-axis.

Table II. Mean value of position and orientation error of rotational VF.

Control	Mean position error (mm)	Mean orientation error (deg.)
None	3.1758	2.4977
Manual	3.4864	0.0440
Autonomous	0.435	0.0495

safer than with the autonomous controller. The combination of both controllers is proposed as a solution to this safety issue by establishing an error threshold above which the manual compensator becomes active, while autonomous compensation can only run below this value. This means that the main objective of autonomous compensation is to keep the TCP inside the preferred subspace rather than getting the TCP into it. Once the preferred subspace is reached, i.e., the error is below the threshold value, autonomous compensation becomes active.

The main advantage of autonomous compensation, in comparison with manual compensation, is that the decoupling nature of translation and rotation is no longer a problem, as while doing movements of one type, possible deviation error of the other type is automatically compensated.

7. Conclusions

This paper introduces a cooperative system for robotic-assisted surgery. Special attention is placed on the interaction of the robotic system with the surgeon, in which a cooperative approach appears to be a good candidate to achieve a suitable integration of the robot within surgical interventions. However, such cooperation implies extra safety measures due to the human element involved. The concept of virtual constraints is used to assure safety during operation by limiting the allowed working space. This is realized in the form of VFs which guide the tool along a predefined direction or path. Previous work related to VFs applied admittance control to create the virtually constrained subspace. This controller type relied on the user-applied force to generate the end-effector's motion, where even deviation error compensation depends on such applied forces (here known as manual error compensation). In such an approach, when deviation error occurs, the virtual preferred directions are redefined to consider such errors, creating a new VF that makes it possible to suitably compensate for it. However, it has been shown that manual compensation does not necessarily compensate for all deviations, especially when the VF is translational and the deviation error is of orientation, or vice versa. In order to solve this problem, the present work proposes another admittance controller with autonomous error compensation, which has a clear division of responsibilities between user and robotic system during cooperative tasks. While the user retains complete control on the movements along the preferred directions, the robotic system takes care of the error compensation independent of the applied forces. Such approach demonstrates considerable minimization of error when compared to the manual compensation.

The methods presented in this paper make a significant contribution toward making manually guided robot movements during cooperative tasks safer and more accurate. They increase the assistive functionality of robotic systems and the level of integration within surgical interventions. The interaction between surgeon and robot becomes friendlier and more intuitive. Moreover, the concept of VFs improves safety measurements during such cooperative tasks, while the surgeon maintains full control over the operation procedure.

References

1. J. J. Abbott, P. Marayong and A. M. Okamura, "Haptic Virtual Fixtures for Robot-Assisted Manipulator, in Robotics Research," **In**: *Proceedings of the 12th International Symposium of Robotics Research, 2005* (S. Thrun, H. Durrant-Whyte and R. Brooks, eds.), Springer Tracts in Advanced Robotics, vol. 28 (Springer, 2007) pp. 49–64.
2. A. Bettini, S. Lang, A. Okamura and G. Hager, "Vision Assisted Control for Manipulation Using Virtual Fixtures: Experiments at Macro and Micro Scales," *Proceedings of the IEEE International Conference on Robotics and Automation*, Washington, DC (11–15 May 2002) pp. 3354–3361.
3. A. Bettini, P. Marayong, S. Lang, A. M. Okamura and G. D. Hager, "Vision assisted control for manipulation using virtual fixtures," *IEEE Int. Trans. Rob. Automat.* **20**(6), 953–966 (Dec. 2004).
4. A. Kapoor, M. Li and R. H. Taylor, "Spatial Motion Constraints for Robot Assisted Suturing Using Virtual Fixtures," *IEEE International Conference on Robotics & Automation*, Taiwan (Sept. 2003) pp. 1954–1959.
5. B. L. Davies, S. J. Harris and F. Rodriguez y Baena, "Hands-On Robotic Surgery: Is This The Future?" *Medical Imaging and Augmented Reality, Proceedings of the 2nd International Workshop, MIAR 2004*, Beijing, China (Aug. 2004) pp. 27–37.
6. G. D. Hager, "Human–Machine Cooperative Manipulation with Vision-Based Motion Constraints," *IEEE/RSJ International Conference on Intelligent Robots and Systems, Workshop on Visual Servoing*, EPFL, Lausanne, Switzerland (30 Sept. to 4 Oct. 2002).
7. J. Wahrburg, S. Pieck, I. Gross, P. Knappe, S. Kuenzler and F. Kerschbaumer, "A navigated mechatronic system with haptic features to assist in surgical interventions," *J. Comp. Aid. Surg.* **8**, 292–299 (2003).
8. R. Kumar, P. Berkelman, P. Gupta, A. Barnes, P. S. Jensen, L. L. Whitcomb and R. H. Taylor, "Preliminary Experiments in Cooperative Human/Robot Force Control for Robot Assisted Microsurgical Manipulation," *Proceedings of the IEEE International Conference Robotics and Automation*, vol. 1, San Francisco, CA, USA (24–28 April 2000) pp. 610–617.
9. M. Li and R. H. Taylor, "Spatial Motion Constraints in Medical Robot Using Virtual Fixtures Generated by Anatomy," *Proceedings of IEEE International Conference on Robotics & Automation*, New Orleans, LA (26 April to 1 May 2004 ICRA) pp. 1270–1275.
10. M. Li, A. Kapoor and R. H. Taylor, "A Constrained Optimization Approach to Virtual Fixtures," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Canada (02–06 August 2005) pp. 1408–1413.
11. M. A. Peshkin et al., "COBOT architecture," *IEEE Trans. Rob. Automat.* **17**(4), 3777 (Aug. 2001).
12. O. Schneider and J. Troccaz, "A six-degree-of-freedom passive arm with dynamic constraints (PADyC) for cardiac surgery

https://doi.org/10.1017/S0263574709990415

application: Preliminary experiments," *Comp. Aid. Surg.* **6**, 340–351 (2001).

13. P. Marayong, M. Li, A. M. Okamura and G. D. Hager, "Spatial Motion Constraints: Theory and Demonstrations for Robot Guidance Using Virtual Fixtures," *Proceedings of the IEEE International Conference on Robotics and Automation*, Taipei, Taiwan (Sept. 2003) pp.14–19.
14. S. Starkie and B. L. Davies, "Advances in Active Constraints and Their Application to Minimally Invasive Surgery," *Fourth International Conference on Medical Image Computing and Computer-Assisted Intervention* (*MICCAI'01*), Lecture Notes in Computer Science, vol. 2208, Utrecht (Oct. 2001) pp. 1316–1321.
15. P. Knappe, S. Pieck and J. Wahrburg, "Komponenten und Architektur eines navigierten Assistenzroboters für chirurgische Anwendungen," *Automatisierungstechnik* **53**(12), 615–626 (Dec. 2005).
16. H. Kazerooni, "Human/Robot Interaction Via the Transfer of Power and Information Signals Part I: Dynamics and Control Analysis," *IEEE Conference on Robotics and Automation*, vol. 3, Scottsdale, Arizona (May 1989) pp. 908–909.
17. H. Kazerooni, "Human–robot interaction via the transfer of power and information signals," *IEEE Trans. Syst. Man Cybernet.* **20**(2), 450–463 (Mar./Apr. 1990).
18. K. Kosuge, H. Yoshida and T. Fukuda, "Dynamic Control for Robot–Human Collaboration," *Proceedings of the 2nd IEEE International Workshop on Robot and Human Communication*, Tokyo, Japan (Nov. 03-05, 1993) pp. 398–399.
19. K. Kosuge, Y. Fujisawa and T. Fukunda "Control of Robot Directly Maneuvered by Operator," *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Yokohama, Japan (26–30 Jul. 1993) pp. 49–54.
20. K. Kosuge and N. Kayamura, "Control of a Robot Handling an Object in Cooperation with a Human," *Proceedings of the 6th IEEE International Workshop on Robot and Human Communication*, Sendai, Japan (29 Sept. to 1 Oct. 1997) pp. 142–147.
21. T. Tsumugiwa, R. Yokogawa and K. Hara, "Variable Impedance Control with Virtual Stiffness for Human–Robot Cooperative Task," *Proceedings of the 41st SICE Annual Conference*, vol. 4, Osaka, Japan (5–7 August 2002) pp. 2329–2334.
22. R. Mukundan, "Quaternions: From Classical Mechanics to Computer Graphics, and Beyond," *Proceedings of the 7th Asian Technology conference in Mathematics*, Melaka, Malaysia (18–21 December 2002) pp. 97–106.
23. C. Sayers, *Remote Control Robotics* (Springer, New York, 1999). ISBN 0387985972.
24. R. V. Mayorga and A. C. K. Wong, "A Singularities Avoidance Approach for the Optimal Local Path Generation of Redundant Manipulators," *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, Philadelphia, Pennsylvania (24–29 April 1988) pp. 49–54.
25. R. Prada and S. Payandeh, "A Study on Design and Analysis of Virtual Fixtures for Cutting in a Training Environment," *Proceedings of World Haptics*, Pisa, Italy (18–20 March 2005).