

Simultaneous haptic guidance and learning of task parameters during robotic teleoperation

Thibault Poignonec, Florent Nageotte, Nabil Zemiti, Bernard Bayle

▶ **To cite this version:**

Thibault Poignonec, Florent Nageotte, Nabil Zemiti, Bernard Bayle. Simultaneous haptic guidance and learning of task parameters during robotic teleoperation. ICRA 2021 - 38th IEEE International Conference on Robotics and Automation, May 2021, Xi'an, China. pp.3619-3625, 10.1109/ICRA48506.2021.9560938 . lirmm-03251658

HAL Id: lirmm-03251658

https://hal-lirmm.ccsd.cnrs.fr/lirmm-03251658

Submitted on 7 Jun 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Simultaneous haptic guidance and learning of task parameters during robotic teleoperation – a geometrical approach

Thibault Poignonec*[1], Florent Nageotte[1], Nabil Zemiti[2] and Bernard Bayle[1]

***Abstract*— Haptic guidance can improve accuracy and dexterity during the teleoperation of a robot, but only if the model of the task used to provide the assistance is accurate. In medical robotics, the registration of a task from pre-operative planning from medical images to the robot's task-space can be erroneous. Additionally, the deformability of the environment can require online correction of a planned task. Therefore, we propose a method to update the geometry and the registration of a path-following task online. This model is simultaneously used to physically guide the user during the teleoperation. Experimental results obtained on a haptic interface show the validity of the approach for a simulated 2D task.**

I. INTRODUCTION

In surgical robotics, teleoperation allows for dexterous and ergonomic manipulation of surgical tools. Although most systems are used in direct teleoperation, tremendous efforts have aimed at developing more advanced forms of assistance. Among these, haptic guidance is a promising way to enhance the surgeon's skills and to make surgery safer [1]. Classic approaches consist in constraining the user by applying forces to the master robot, either to prevent the slave robot from entering a restricted area or to guide the user along a path. Respectively referred to as *Forbidden-Region Virtual Fixtures* and *Guidance Virtual Fixtures* [2], these methods can be seen as a form of haptic shared-control, which consists in mixing user inputs with a guidance trajectory through physical interaction [3]. In such schemes, a guidance path must be defined, either from pre- or intra-operative planning.

Pre-operative planning can only be used if the task is correctly registered in the slave robot's frame. The deformations and displacements of the organs invalidate off-line planning, and the task must be adapted online. This advocates for adaptive guidance systems that are able to plan or re-plan online from intra-operative images. However, vision-based approaches for surgical task planning have many limitations, such as the presence of occlusions, the sparsity of visual features, or the lack of stereoscopic vision. The resulting task planning and adaptation may therefore be inaccurate. Moreover, the surgeon might base his gestures on information not contained in the images such as knowledge or experience.

Regardless of the origin of the modeling errors, if the model of the task used to generate haptic guidance is incorrect, the user will have to apply forces to the master interface to return the tool toward a more desirable position. Such interventions from the human operator can either be ignored, be facilitated through the reduction of the rendered stiffness, or be exploited to refine the guidance model.

In recent years, these challenges have become very popular, especially in the field of collaborative robotics. Two approaches have been proposed to update task models from user inputs. The first assumes that the user will interact with the interface by rendering a mechanical impedance. If interaction forces are minimized over time, the modeling errors will likewise be minimized. Methods have been proposed to deform trajectories, both parametric [4] and non-parametric [5], [6], from interaction forces. However, these interaction forces only provide limited information for learning.

Another approach consists in assuming that the user can overcome the guidance forces and perform the desired trajectory, even though the guidance is incorrect. In this case, the position of the robot can be seen as a noisy demonstration of the desired trajectory. Existing studies use this assumption to infer the user's intentions in the form of a goal [7] or a preferred policy [8]. But only a few contributions adapt an existing model as in [9], [10], where the trajectory is updated iteratively through consecutive demonstrations. Although iterative learning is pertinent to teach a robot how to perform a task, it is insufficient to update it during its execution. To do so, a recent work uses a window of recent tool poses to update the geometry of virtual fixtures [11]. They rely on the detection of tool/tissue interactions to collect observations about the desired task in order to move the passive Virtual Fixture. But as the method requires a means to measure or estimate distal forces, it cannot be applied to flexible or cable-driven robots, among others.

The present work focuses on path-following tasks, relevant for various Minimally Invasive Surgery (MIS) robotic applications such as optical biopsies [12], dissections [13], or ablations. The registration of the task is a critical issue in surgical robotics and is hence of special interest to us. We propose to use a variable size window of sampled master robot positions to adapt online the geometry of the task model. The geometry of the path, defined by a parametric model, is corrected independently from the dynamical aspect of the task. As in previous work, sample positions are considered to be noisy observations of a desired trajectory [10], [4]. But here, a sliding window of past observations is retained, and the learning is only performed when enough information has been collected. To this end, we propose a criterion that is used to supervise the learning and dynamically adapt the size of the window. The task model is updated while it is simultaneously used to physically guide the user, allowing for *in-situ* correction of the task from only user inputs.

*tpoignonec@unistra.fr. Authors are with the ICube laboratory, University of Strasbourg, Strasbourg, France[1] and the LIRMM, University of Montpellier, CNRS, Montpellier, France [2]

First, the problem, background and used notations are introduced in section II. After detailing the proposed adaptive size window method in section III, experimental results are provided in section IV, demonstrating that the approach can cope with planning errors.

II. Problem formulation

A. Teleoperation and haptic guidance

In the following the teleoperation of a robot is considered in order to perform a remote trajectory-following task. At the master-side, the user manipulates a haptic interface whose planar or 3D Cartesian position $x(t) \in \mathbb{R}^m$ is mapped to the slave workspace to obtain the slave position reference. This position-position mapping, as well as the positioning of the slave robot, are assumed sufficiently accurate so that positions can be considered indifferently in either workspace. In the following, the master side is chosen for the sake of clarity.

The user's desired trajectory is denoted $x_d(t)$. This trajectory is planned from the visual feedback provided by a static camera and performed by manipulating the master interface. In order to guide the user during the task execution, guidance forces $f_g(t)$ are applied by the master using Cartesian impedance control, which allows for compliant tracking of a reference position denoted as $x_g(t)$. The impedance control law is defined as follows:

$$M(t)\ddot{\widetilde{x}}(t) + D_d(t)\dot{\widetilde{x}}(t) + K_d\widetilde{x}(t) = f_h(t) \tag{1}$$

with $\widetilde{x}(t) = x(t) - x_g(t)$, $D_d(t)$, $K_d \in \mathbb{R}^{m\times m}$ define the desired impedance, $M(t) \in \mathbb{R}^{m\times m}$ is the natural Cartesian inertia of the master robot and $f_h(t)$ is the external force, applied by the user to the robot. K_d is set constant and $D_d(t)$ is chosen to impose a critically damped behavior using the *factorization design* method [14], [15]. To impose the behavior, guidance forces $f_g(t)$ are computed considering that the robot's Cartesian dynamical model (after compensation for gravity) is $M(t)\ddot{x}(t) = f_g(t) + f_h(t)$.

As the accelerations are low and the mass of the interface is negligible compared to the human arm and hand, the inertial term is neglected. The guidance forces are

$$f_g(t) = -K_d\widetilde{x} - D_d(t)\dot{\widetilde{x}} \tag{2}$$

B. Task parametric model

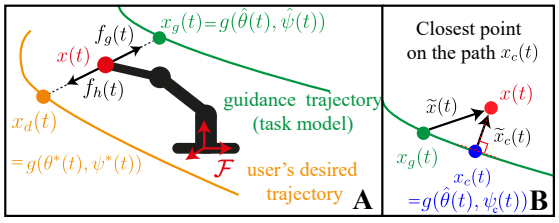


Fig. 1. Desired task, guidance model, and notations used throughout this work. (A) The guidance and the desired trajectories, both defined in the master robot frame of reference, are illustrated along with the guidance force f_g and the one applied by the human f_h. Errors are exaggerated for the purpose of illustration. (B) Positions of interest x_g, x , and x_c. Associated errors $\widetilde{x}$ and $\widetilde{x}_c$ are also illustrated.

A model of the task is necessary to guide the user and, as in previous work [4], it is considered that learning a complete model of the user intentions is intractable. They are restricted and modeled by a predefined family of parameterized curves $g(\theta, \psi)$ including both the desired path x_d and the guidance path x_g. The two parameters $\theta \in \mathbb{R}^n$ and $\psi \in \mathbb{R}$, respectively encode the path geometry and the position along it.

If the user's desired trajectory $x_d(t)$ belongs to this family of curves, then there is a set of parameters θ^* and a function $\psi^*(t)$ that minimize the modeling error $\|x_d(t) - g(\theta, \psi(t))\|$. The haptic guidance should aim at guiding the motion of the robot along this desired trajectory, but the desired parameters θ^* and advancement profile $\psi^*(t)$ are unknown. Although an estimate for these can be obtained from pre-operative planning or from intra-operative data, it is not guaranteed to accurately represent what the user desires. During the execution of the task, the guidance reference $x_g(t)$ is then generated from an estimation of the desired task:

$$x_g(t) = g(\hat{\theta}, \hat{\psi}(t)) \tag{3}$$

These estimates could be erroneous, or become so during the task execution. The online refinement of the different parameters $\hat{\theta}$ and the advancement policy $\hat{\psi}(t)$ is therefore a critical issue.

C. Motion along the path

There are several ways to define the position along the path $\psi(t)$ depending on what is known about the desired velocity profile. A straightforward approach is to simply constrain the motion of the robot on the path, a method referred to as *Guidance Virtual Fixture* [2]. A classic implementation is to define the guidance reference $x_g(t)$ as the point on the guidance path minimizing the distance to robot position $x(t)$:

$$x_g(t) = g(\hat{\theta}(t), \psi_c(t)) \tag{4}$$

$$\psi_c(t) = \arg\min_{\psi} d(x(t), g(\hat{\theta}(t), \psi)) \tag{5}$$

with $d(A, B)$ the distance between the points A and B.

Alternatively, if the velocity profile is known, a function $\hat{\psi}(t)$ can be used to move the reference along the path to actively pull the user forward. This approach is not well adapted to the present case, since the user continuously manipulates the interface and could disagree with such imposed motion. Although methods to stop the advancement [16] or adjust it [17] have been proposed, they lack the interactivity required for teleoperation.

In this work, a local linear model $\hat{\psi}(t) = at + b$ is learned online and is used to compute the guidance reference trajectory $x_g(t)$. First, the closest point on the path is computed from (5), yielding an associated $\psi_c(t)$. The model of $\hat{\psi}(t)$ is then updated by minimizing a cost defined as a linear combination of the squared positioning and velocity errors $\mathcal{L}_\psi = e_\psi^2 + \dot{e}_\psi^2$, with $e_\psi = \psi_c - \hat{\psi}(t)$. The update rule

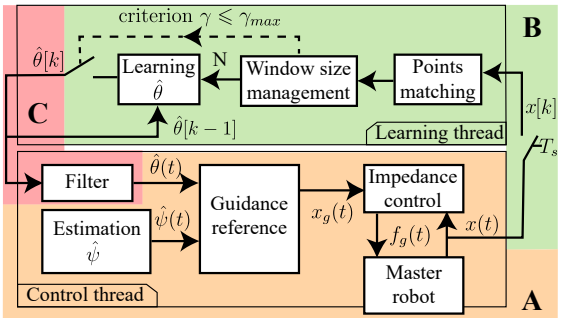


Fig. 2. Synopsis of the proposed method: (A) A haptic guidance is provided from a reference $x_g(t) = g(\hat{\theta}(t), \hat{\psi}(t))$. (B) A variable size window learning algorithm learns the geometrical parameters of the task $\theta[k]$ from a series of observations $x(t) = \hat{x}_d(t)$. The size of the window N is controlled to guarantee that the window contains sufficient information. If this criterion is unmet, the learning is suspended until more sampled user positions are available. At each iteration, the points associations are updated to take parameters changes into account. (C) New parameters estimates $\hat{\theta}[k]$ are sent to the guidance loop and filtered to ensure smooth transitions.

uses the damped Gauss-Newton method [18] as follows:

$$\hat{\psi}(t) = at + b \tag{6}$$

$$\begin{bmatrix} \dot{a} \\ \dot{b} \end{bmatrix} = \alpha \left(J_\psi^T J_\psi \right)^{-1} J_\psi^T \begin{bmatrix} e_\psi \\ \dot{e}_\psi \end{bmatrix} ; J_\psi = \begin{bmatrix} \frac{\partial \hat{\psi}}{\partial a} & \frac{\partial \hat{\psi}}{\partial b} \\ \frac{\partial \dot{\hat{\psi}}}{\partial a} & \frac{\partial \dot{\hat{\psi}}}{\partial b} \end{bmatrix} \tag{7}$$

where $\alpha \in]0;1[$ is a learning factor used to tune the behavior of the guidance. This allows to filter $\hat{\psi}(t)$, smoothing the motion along the path and preventing undesired jerk. It can be observed that if the value of α is close to one, the resulting behavior will be the one obtained with eq. (4) and (5).

III. Method

In this work, we aim to update a task model online to cope with possibly significant changes between pre-operative planning and in-situ realization. These changes can arise from registration errors or be due to a change of the environment. Either way, changes in the desired parameters θ^* might happen, leading to inaccurate haptic guidance. We present a method to update the geometry of a parametric path online from user actions. The general synopsis is illustrated in fig.2, and it is detailed in the following sections.

A. Sliding window learning

It is assumed that in case of conflicts between the guidance and the user, the latter can overcome applied forces to perform the desired trajectory. This hypothesis can be invalid in some cases, since it has been shown that a too large discrepancy between guidance and desired trajectory can impede the execution of the task [19]. Here, applied forces are very low (under 5N) and the user can easily overcome them. The position of the robot $x(t)$ is considered to be a noisy observation of the desired trajectory $x(t) = x_d(t) + \epsilon(t)$ or $x(t) = g(\theta^*, \psi^*(t)) + \epsilon(t)$, with $\epsilon(t)$ the noise. In order to minimize the guidance errors between guidance reference $x_g(t)$ and user desired position over time, the guidance task model parameters must be updated. To do so, the learning of the parameters θ is re-framed as a non-rigid registration problem, fully independent from the motion along the path.

The user inputs $x(t)$ are sampled periodically to form a trace of N observations $x[k-i]$ spanning a finite temporal horizon. With t the current time and T_s the sampling period (see fig. 2), we write $x[k-i] = x(t - iT_s)$. It can be observed that if the task model changes with the update of $\hat{\theta}[k]$, the values of $\psi_c(t)$ computed at times $t = t - iT_s$ will no longer be solutions of (5). Therefore, points associations must be re-computed at each step k of the learning loop according to (5) with the updated parameters $\hat{\theta}[k]$. These point correspondences between user input and guidance path can be used to define errors $\|\widetilde{x}_c\|$, the smallest distances between $x[k-i]$ and the guidance path (see fig. 1). This distance is computed from $\psi_c[k-i]$, the re-evaluation of the position along the path according to (5), such that $\widetilde{x}_c[k-i] = x[k-i] - g(\hat{\theta}[k], \psi_c[k-i])$.

The learning of the task parameters θ^* is formulated as a quadratic optimization problem, minimizing the cost function $\mathcal{L}_\theta$ defined as follows:

$$\mathcal{L}_\theta[k] = \frac{1}{N} \sum_{i=0}^{N-1} \|\widetilde{x}_c[k-i]\|^2 \tag{8}$$

To minimize this cost function over time and subsequently reduce task modeling errors, a damped Gauss-Newton optimization is implemented. This method is highly tractable for online schemes and allows the learning rate to be tuned, a desirable property for online registration-like algorithms [20]. With λ the learning rate, the parameters update rule is

$$\hat{\theta}[k] = \hat{\theta}[k-1] - \lambda \left(\boldsymbol{J_\theta}[k]^T \boldsymbol{J_\theta}[k] \right)^{-1} \mathbf{J_\theta}^T[k] \widetilde{\mathbf{x}}_\mathbf{c}[k] \tag{9}$$

where $\mathbf{J}_\theta[k] \in \mathbb{R}^{Nm \times n}$ and $\widetilde{\mathbf{x}}_\mathbf{c}[k] \in \mathbb{R}^{Nm \times 1}$ are defined as

$$\widetilde{\mathbf{x}}_\mathbf{c}[k] = \begin{bmatrix} \widetilde{x}_c[k] \\ \vdots \\ \widetilde{x}_c[k-(N-1)] \end{bmatrix}, \quad \mathbf{J}_\theta[k] = \begin{bmatrix} J_\theta[k] \\ \vdots \\ J_\theta[k-(N-1)] \end{bmatrix} \tag{10}$$

with $J_\theta[k-i] = \left. \frac{\partial g(\theta, \psi)}{\partial \theta} \right|_{\theta=\hat{\theta}[k], \psi=\psi_c[k-i]}$

A critical issue is the management of the window size N. At the initialization, the window is empty and N is set to zero. As new observations are sampled, the window expands towards maximal size N_{max}, whose choice depends on available computational power. However, N only defines a temporal windowing of the observations, whereas the sampling should also be based on a geometrical criterion: observations must homogeneously span both a temporal and a spatial horizon. Intuitively, a large number of observations do not give any information about the task if no displacement has occurred. Similarly, if the observations are too sparse spatially, they might not be sufficient to learn the correct task. A naive choice of the window size can lead to degenerated cases, impeding the accuracy and stability of the learning. This advocates for an adaptive size window based on the evaluation of the richness of the collected observations.

B. Automatic window size management

In order to adapt the size of the window, it is necessary to have a way of evaluating if the window contains enough information. Let us assume for now that there is a function

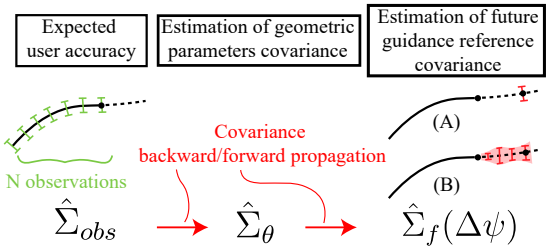


Fig. 3. Illustration of the estimation of future guidance reference position $x_g(t+\Delta t)$ from the N available observations $\hat{x}_d(k)$. (A) Estimation of the covariance $\hat{\Sigma}_f(\Delta t)$ of $x_g(t+\Delta t)$ at $t = t+\Delta t$. (B) Same, but averaged over a finite horizon $t \in [t; t+\Delta t]$.

γ such that if $\gamma \leqslant \gamma_{max}$, the window of size N is sufficient for a meaningful and stable learning. The size of the window should therefore verify the inequality $\gamma \leqslant \gamma_{max}$, with γ_{max} a manually tuned threshold. New observations are sampled one at a time, such that the optimal size for the window varies slowly. We propose to similarly expand or shrink the window used for learning by unitary increments:

$$\begin{cases} N_{min} \mathrel{+}= 1 & if\ \gamma[k] \geqslant s\gamma_{max} \text{ and } N_{min} < N_{max} \\ N_{min} \mathrel{-}= 1 & if\ \gamma[k] < s\gamma_{max} \text{ and } N_{min} > 1 \end{cases} \tag{11}$$

where N_{min} is the minimal size of the window verifying the criterion $\gamma \leqslant \gamma_{max}$ and $s < 1$ is a coefficient used to ensure a margin in the minimal window size N_{min}. Finally, the actual size $N[k]$ used for the learning is chosen such that $N[k] = N_{min}[k]$ and the parameters are only updated if $\gamma[k] \leqslant \gamma_{max}$ (see fig.2).

To implement this scheme, such a criterion $\gamma[k]$ must be devised. One method consists in back-propagating expected observations covariance $\hat{\Sigma}_{obs}$ through the model to estimate the covariance matrix of the parameters $\hat{\theta}$. To this end, the component of the observations' covariance tangent to the guidance path has to be ignored, as these errors have been compensated during the points association stage described in section III-A. Let $\mathcal{P}_i$ be the projection matrix to the plane normal to T_i, the vector tangent to the path defined as $T_i = \frac{\dot{x}_g[k-i]}{\|\dot{x}_g[k-i]\|}$. The Jacobian matrices $J_\theta[k-i]$ associated to each observation contained in the minimal window are projected to the corresponding normal planes such that

$$\boldsymbol{J_{proj}}[k] = \begin{bmatrix} \mathcal{P}_0 & & 0 \\ & \ddots & \\ 0 & & \mathcal{P}_{N_{min}-1} \end{bmatrix} \begin{bmatrix} J_\theta[k] \\ \vdots \\ J_\theta[k-(N_{min}-1)] \end{bmatrix} \tag{12}$$

By assuming that the observation noise, which corresponds to the user accuracy in following the desired path, is isotropic and Gaussian, the variance projected to the normal plane is $\hat{\Sigma}_{obs} = \sigma_0 I_{N(m-1)\times N(m-1)}$. The covariance of the uncertainty on the parameters can be obtained as follows:

$$\hat{\Sigma}_\theta = \left(\boldsymbol{J_{proj}}^T \hat{\Sigma}_{obs}^{-1} \boldsymbol{J_{proj}}\right)^{-1} = \sigma_0 \left(\boldsymbol{J_{proj}}^T \boldsymbol{J_{proj}}\right)^{-1} \tag{13}$$

$\hat{\Sigma}_\theta$ could directly be used to define a criterion, but this would require to know beforehand what uncertainty on the parameters is acceptable – not a trivial question. Furthermore, doing so would yield a task-dependent criterion that would have to be hand-crafted. To cope with these issues, the parameters' estimated covariance matrix is used to predict the uncertainty

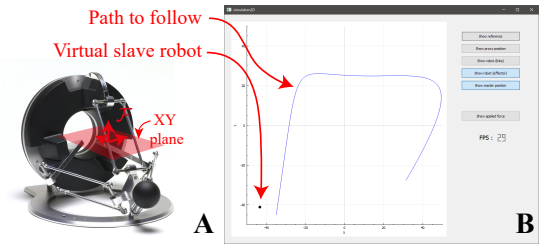


Fig. 4. Experimental setup: (A) 3 DOF haptic interface (Omega 3, Force Dimension) manipulated by the user. The task is defined in the horizontal XY plane. (B) Visual feedback featuring a dot and a curve, respectively representing the XY position of the virtual slave robot and the path that the user is asked to follow. This display of the *"desired path"* allows for an objective assessment of the learning performance.

of the guidance reference $x_g(t)$ further along the path (see fig.3). Let $J_f(\Delta\psi)$ be the Jacobian of $g(\hat{\theta}[k], \hat{\psi}[k] + \Delta\psi)$ with respect to the parameters θ projected to the normal plane as in (12) and $\hat{\Sigma}_f(\Delta\psi)$ the covariance of $g(\hat{\theta}[k], \hat{\psi}[k] + \Delta\psi)$ normal to the path defined as follows:

$$\hat{\Sigma}_f(\Delta\psi) = J_f(\Delta\psi)\hat{\Sigma}_\theta J_f(\Delta\psi)^T \tag{14}$$

The criterion $\gamma[k]$ is computed as the ratio between the accuracy expected for a single observation and for the guidance reference position in the near future. In the case of a 2D task, the projected covariances matrix $\hat{\Sigma}_{obs}$ and $\hat{\Sigma}_f(\Delta\psi)$ are reduced to scalar values, respectively σ_0 and σ_f:

$$\gamma[k] = \frac{\sigma_f[k]}{\sigma_0} = J_f(\Delta\psi)\left(\boldsymbol{J_{proj}}^T \boldsymbol{J_{proj}}\right)^{-1} J_f(\Delta\psi)^T \tag{15}$$

For 3D tasks, $\gamma[k]$ could be defined as a ratio between matrices' determinants or eigen values, among others. In practice, the actual criterion $\gamma[k]$ used in this algorithm is averaged over the interval $[\psi[k], \psi[k] + \Delta\psi[k]]$ to reduce local variations (see fig. 3.B). The threshold γ_{max} is chosen considering that the guidance should at least be as accurate as the user input and if possibly more ($\gamma_{max}[k] \ll 1$) . The criterion γ is computed from eq. (15), where $\Delta\psi[k]$ is defined as in (16) in order for the prediction to cover both a temporal and spatial horizon, respectively denoted as Δt_{min} and Δx_{min}:

$$\Delta\psi[k] = max(\Delta t_{min}\hat{\dot{\psi}}[k],\ \Delta x_{min}\left\|\frac{\partial g(\hat{\theta}[k], \hat{\psi}[k])}{\partial \psi}\right\|^{-1}) \tag{16}$$

IV. Experimental results

The proposed method for online task update is implemented to evaluate its performance on a 2D task. A first experiment shows that the method can correctly learn task parameters online. Another case-study illustrates interesting challenges that can arise in some situations.

A. Setup

The master robot is a 3DOF haptic interface Omega 3 from Force Dimension and a simulated robot and environment serves as the teleoperated robot (fig. 4). The task is constrained to the XY plane through the rendering of a higher stiffness k_{plane} in the Z direction, such that $K_d = diag(k_g, k_g, k_{plane})$ where k_g is the guidance stiffness. The haptic loop runs at 1 kHz, and the learning loop runs at 10 Hz (hence $T_s = 0.1$ s). The method is implemented with the

parameter values given in table IV-A. In order to smooth the transitions introduced by the update of $\hat{\theta}[k]$, the parameters actually used for the guidance $\hat{\theta}(t)$ are filtered (see fig. 2.C) by a first order filter with a time constant of 1 s.

The experiment is as follows. A path is displayed on the screen, along with a dot representing the position of the slave robot (fig. 4). The user is asked to manipulate the haptic interface to follow the path with the virtual "robot", with no constraint on the velocity. The path displayed on the screen is considered to define the ground truth for the geometrical part of the desired task $g(\theta^*, \psi)$.

Section	Name	Value	Unit(s)
II-C	a, b (at $t=0$)	0.0, 0.0	–
	α	0.001	–
III-A	λ	0.2 (set to $2T_s$)	–
III-B	N_{max} (limit)	200	–
	γ_{max}, s	0.2, 0.8	–
	Δt_{min}, Δx_{min}	0.5, 0.01	s, m resp.
IV-A	k_g, k_{plane}	200, 1500	N/m

TABLE I

PARAMETERS VALUES USED FOR THIS EXPERIMENT.

B. Online rigid registration from user inputs

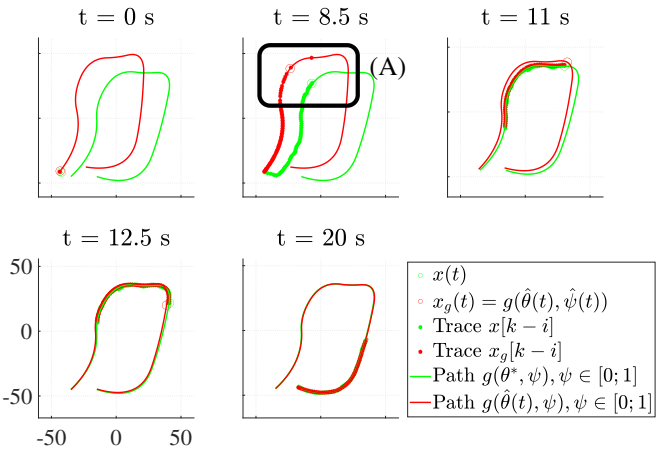


Fig. 5. Snapshots of the different positions of interest at different times throughout the experiment. All positions are expressed in millimeters with the scale displayed in the bottom-left figure. In the area (A), a discontinuity in the closest points search can be observed. The filtering of $\hat{\psi}(t)$ proposed in section II-C prevents the appearance of jerk in the guidance.

Let us first consider the case of a task whose geometry is defined by a spline Γ rigidly registered in the robot's frame of reference such that

$$g(\theta, \psi) = R(\theta_1)\Gamma(\psi) + t(\theta_{2:3}) \qquad (17)$$
$$= \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 \\ \sin\theta_1 & \cos\theta_1 \end{bmatrix} \Gamma(\psi) + \begin{bmatrix} \theta_2 \\ \theta_3 \end{bmatrix}$$

where R and t are planar rotation and translation, $\psi \in [0;1]$ encodes the position along the path and θ_i is the i$^{\text{th}}$ component of the parameters vector θ. The desired parameters are $\theta^* = [0 \quad 5 \quad -5]$ and the estimated parameters are initialized at $\hat{\theta}(t=0) = [9 \quad -10 \quad 5]$ (in [deg. mm mm]).

The execution of one learning step (fig.2.B) takes 20 ms on average, well under the 100 ms needed to run at a frequency

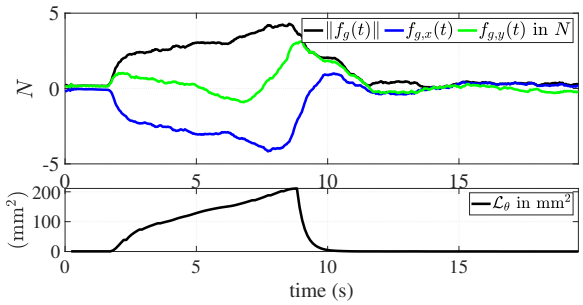


Fig. 6. (Top) In-plane components (x and y) and Euclidean norm of the guidance forces $f_g(t)$. (Bottom) Cost function $\mathcal{L}_\theta[k]$.

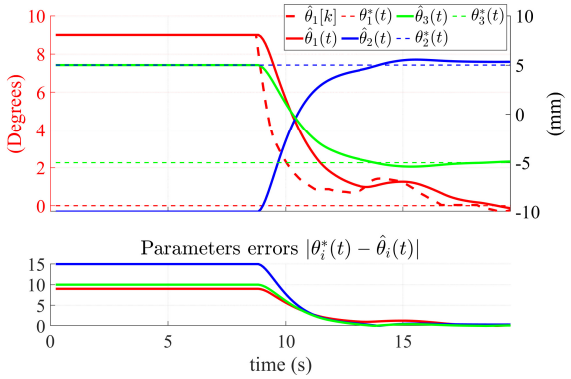


Fig. 7. (Top) Evolution of the estimated parameters and comparison with the ground truth. Only the parameters $\hat{\theta}(t)$ used for generating the guidance reference are shown, except for $\hat{\theta}_1[k]$, included to show the effect of the filter. (Bottom) Absolute error on estimated parameters $|\theta_i^*(t) - \hat{\theta}_i(t)|$.

of 10 Hz. While the initial parameters estimation error is large, as visible on fig.5, it is significantly reduced once the updates are allowed by the window size manager at $t \sim$ 9s. This event is triggered when the criterion defined by (15) is met (fig.8). The update of the estimated parameters is then propagated to the parameters used to guide the user (fig.7). The cost function $\mathcal{L}_\theta$ (fig.6) converges toward $0.2\,\text{mm}^2$. The residual error can be explained by the accuracy of user positioning, as the mean value of $\|\widetilde{x}_c(t)\|$ when the guidance parameters are correct has been evaluated at 0.35 mm. As expected, the intensity of the guidance forces decreases along with the task modeling errors (fig. 6). It can be noted that once geometric parameters are correctly estimated, residual forces are mostly tangent to the guidance path. These forces are created by errors on the estimation of the desired position and velocity along the path $\hat{\psi}(t)$ (see fig.9).

The criterion γ is subject to local variations and consequently, the minimal size of the window N_{min} varies (fig. 8). This variation on γ can partly be imputed to points matching updates, but even when the parameters have converged, some variations remain, suggesting than the cause is the forward propagation of the co-variance. First, the sensitivity of the task model to one parameter or another, mainly the rotation here, changes depending on the location along the path. Second, the derivative of the path is not constant, and a re-parameterization by arc length could reduce the variations.

The model of the velocity profile $\hat{\psi}(t)$ follows $\psi_c(t)$ as

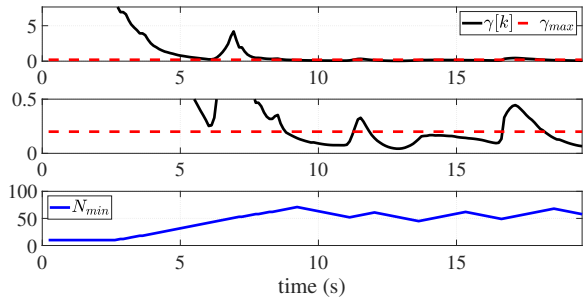


Fig. 8. (Top) Evolution of the criterion $\gamma[k]$ and threshold γ_{max}, displayed on a log-scale. (Center) Zoom of the figure above it. (Bottom) The window shrinks or expands to keep the criterion within a predetermined range. The maximal size of the window $N_{max} = 200$ is never reached.

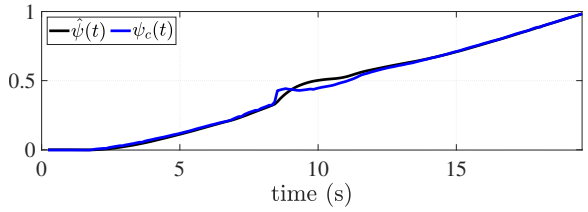


Fig. 9. Advancement along the path: estimated model $\hat{\psi}(t)$ and value of $\psi_c(t)$ associated to the closest point $x_c(t) = g(\hat{\theta}[k], \psi_c(t))$.

intended (see fig.9). The residual tracking error e_ψ has two identified sources: the discontinuities in $\psi_c(t)$ due to local minima and variations in the user's velocity. The latter is low in this experiment, since the input motion is smooth (no stopping or major slowing down). Local minima appear when the position of the robot is further away from the path than the radius of curvature of the path. In this case, the iterative orthogonal projection (see eq.5) is not guaranteed to converge towards the closest point [21]. Since the distance minimization method used is robust and can cope with local minima, the closest point "jumps" and creates discontinuities. This phenomenon, well described in [22], is visible at t $\sim$ 8.4 s in figure 9 and in the area (A) at t = 8.5 s (fig.5). By filtering $\psi_c(t)$ using (7), the jerk tangent to the path is reduced.

C. Second experiment: parametric sinusoidal path

This second task is defined by a sinusoidal path Γ registered in the robot frame of reference such that $g(\theta, \psi) = R(\theta_1)\Gamma(\theta_{4:5}, \psi) + t(\theta_{2:3})$, with R and t defined as in eq. (17), and $\Gamma(\theta_{4:5}, \psi) = \begin{bmatrix}\theta_4\psi & \theta_5 \sin\psi\end{bmatrix}$. Task parameters are $\hat{\theta}(t=0) = [9 \quad -50 \quad 1 \quad 5 \quad 15]$ and $\theta^* = [0 \quad -40 \quad 0 \quad 4 \quad 35]$. θ_4 and θ_5 encode the spatial period and the amplitude of the sinus respectively. The same protocol as in section IV-C is followed. It should be noted that this scenario has been chosen on purpose so as to reveal some of the important issues that need to be discussed. In this experiment, the error on the parameters estimation is well reduced over time and the estimated task is visually correct after about 15 s of interaction (fig.10). The user motion becomes smoother as the desired task parameters are learned, showing that the quality of the guidance improves, and with it the quality of the tracking of the desired path. However, it can be observed in fig. 10 that the parameter $\hat{\theta}_5$ encoding the amplitude of the sinus converges slowly. This is

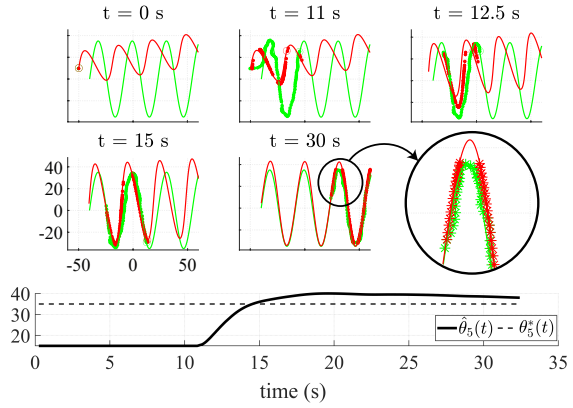


Fig. 10. (Top) Snapshots of the different positions of interest at different times throughout the experiment. See figure 5 for the legend, that is not included here due to limited space. (Bottom) Evolution of the parameter $\hat{\theta}_5(t)$ encoding the amplitude of the sinusoidal task.

due to incorrect point associations between the sample user positions and the path, arising near the crests of the sinus as in the zoom in fig.10. The cause is the same as for the local minima discussed in section IV-B.

V. CONCLUSIONS AND FUTURE WORK

In this work, we have proposed a novel method to update the geometry of a parametric task model from only user position inputs. Contrary to iterative approaches, the task model is learned while simultaneously used to physically guide the user, allowing online correction of the haptic guidance. Experiments show that the path parameters can be learned online when they are initially erroneous. Overall, this method shows great potential to overcome large registration or planning errors in physically guided teleoperation.

The presence of local minima in the distance minimization optimization can create discontinuities in ψ_c that have been filtered. But in some cases, it can cause the point-matching step in the learning of $\hat{\theta}$ to yield temporarily erroneous information. This is shown in the second experiment, where the sinusoidal curve's strong curvature impedes the learning of one of the parameters. However, this could be handled by using more advanced point set registration techniques [23].

The results are already quite satisfactory, and ongoing work aims at improving the method. A next step could involve detecting possible online changes in order to refresh the window. Furthermore, an adaptive stiffness scheme such as the one presented in [9] could improve the capacity of the user to execute the desired trajectory when the parameters estimation error is very large. In this work, as the stiffness of the impedance controller is constant, the question of the passivity has not been discussed. It would be interesting to assess if the guidance system can be passive and if not, to implement a method to enforce it [24], [11].

ACKNOWLEDGMENT

This work was supported by French state funds managed by the ANR within the Investissements d'Avenir programme for the **Labex CAMI** (ANR-11-LABX-0004) and the **IHU of Strasbourg** (ANR-10-IAHU-02).

REFERENCES

[1] N. Enayati, E. D. Momi, and G. Ferrigno, "Haptics in Robot-Assisted Surgery: Challenges and Benefits," *IEEE Reviews in Biomedical Engineering*, vol. 9, pp. 49–65, 2016.

[2] S. A. Bowyer, B. L. Davies, and F. Rodriguez y Baena, "Active Constraints/Virtual Fixtures: A Survey," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 138–157, Feb. 2014.

[3] D. A. Abbink, M. Mulder, and E. R. Boer, "Haptic shared control: smoothly shifting control authority?" *Cognition, Technology & Work*, vol. 14, no. 1, pp. 19–28, Mar. 2012. [Online]. Available: https://doi.org/10.1007/s10111-011-0192-5

[4] D. P. Losey and M. K. O'Malley, "Learning the Correct Robot Trajectory in Real-Time from Physical Human Interactions," *ACM Transactions on Human-Robot Interaction*, vol. 9, no. 1, pp. 1–19, Dec. 2019. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3375676.3354139

[5] D. P. Losey and M. K. O'Malley, "Trajectory Deformations From Physical Human–Robot Interaction," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 126–138, Feb. 2018.

[6] S. S. Restrepo, G. Raiola, P. Chevalier, X. Lamy, and D. Sidobre, "Iterative virtual guides programming for human-robot comanipulation," in *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, July 2017, pp. 219–226, iSSN: 2159-6255.

[7] S. Javdani, H. Admoni, S. Pellegrinelli, S. S. Srinivasa, and J. A. Bagnell, "Shared autonomy via hindsight optimization for teleoperation and teaming," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 717–742, June 2018. [Online]. Available: https://doi.org/10.1177/0278364918776060

[8] G. Raiola, S. S. Restrepo, P. Chevalier, P. Rodriguez-Ayerbe, X. Lamy, S. Tliba, and F. Stulp, "Co-manipulation with a library of virtual guiding fixtures," *Autonomous Robots*, vol. 42, no. 5, pp. 1037–1051, 2018.

[9] T. Kastritsi, F. Dimeas, and Z. Doulgeri, "Progressive Automation with DMP Synchronization and Variable Stiffness Control," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3789–3796, Oct. 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8411480/

[10] F. Dimeas, F. Fotiadis, D. Papageorgiou, A. Sidiropoulos, and Z. Doulgeri, "Towards progressive automation of repetitive tasks through physical human-robot interaction," in *Human Friendly Robotics*. Springer, 2019, pp. 151–163.

[11] M. Selvaggio, G. A. Fontanelli, F. Ficuciello, L. Villani, and B. Siciliano, "Passive Virtual Fixtures Adaptation in Minimally Invasive Robotic Surgery," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3129–3136, Oct. 2018.

[12] O. C. Mora, P. Zanne, L. Zorn, F. Nageotte, N. Zulina, S. Gravelyn, P. Montgomery, M. de Mathelin, B. Dallemagne, and M. J. Gora, "Steerable OCT catheter for real-time assistance during teleoperated endoscopic treatment of colorectal cancer," *Biomedical Optics Express*, vol. 11, no. 3, pp. 1231–1243, Mar. 2020, publisher: Optical Society of America. [Online]. Available: https://www.osapublishing.org/boe/abstract.cfm?uri=boe-11-3-1231

[13] R. Moccia, M. Selvaggio, L. Villani, B. Siciliano, and F. Ficuciello, "Vision-based Virtual Fixtures Generation for Robotic-Assisted Polyp Dissection Procedures," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 7934–7939, iSSN: 2153-0866.

[14] A. Albu-Schaffer, C. Ott, and G. Hirzinger, "A passivity based Cartesian impedance controller for flexible joint robots - part II: full state feedback, impedance design and experiments," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 3, Apr. 2004, pp. 2666–2672 Vol.3, iSSN: 1050-4729.

[15] C. Ott, *Cartesian impedance control of redundant and flexible-joint robots*. Springer, 2008.

[16] A. J. Ijspeert, J. Nakanishi, T. Shibata, and S. Schaal, "Nonlinear dynamical systems for imitation with humanoid robots," in *Proceedings of the IEEE/RAS International Conference on Humanoids Robots (Humanoids2001)*, 2001, pp. 219–226.

[17] Y. Tirupachuri, G. Nava, L. Rapetti, C. Latella, and D. Pucci, "Trajectory Advancement during Human-Robot Collaboration," in *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, Oct. 2019, pp. 1–8, iSSN: 1944-9437.

[18] A. Björck, *Numerical methods for least squares problems*. SIAM, 1996.

[19] J. v. Oosterhout, J. G. W. Wildenbeest, H. Boessenkool, C. J. M. Heemskerk, M. R. d. Baar, F. C. T. v. d. Helm, and D. A. Abbink, "Haptic Shared Control in Tele-Manipulation: Effects of Inaccuracies in Guidance on Task Execution," *IEEE Transactions on Haptics*, vol. 8, no. 2, pp. 164–175, Apr. 2015.

[20] M. Hersch, A. Billard, and S. Bergmann, "Iterative Estimation of Rigid-Body Transformations," *Journal of Mathematical Imaging and Vision*, vol. 43, no. 1, pp. 1–9, May 2012. [Online]. Available: https://doi.org/10.1007/s10851-011-0279-x

[21] K. Ko and T. Sakkalis, "Orthogonal projection of points in CAD/CAM applications: an overview," *Journal of Computational Design and Engineering*, vol. 1, no. 2, pp. 116–127, Apr. 2014.

[22] J. Pegna and F.-E. Wolter, "Surface Curve Design by Orthogonal Projection of Space Curves Onto Free-Form Surfaces," *Journal of Mechanical Design*, vol. 118, no. 1, pp. 45–52, Mar. 1996, publisher: American Society of Mechanical Engineers Digital Collection. [Online]. Available: https://asmedigitalcollection.asme.org/mechanicaldesign/article/118/1/45/431932/Surface-Curve-Design-by-Orthogonal-Projection-of

[23] B. Bellekens, V. Spruyt, R. Berkvens, R. Penne, and M. Weyn, "A benchmark survey of rigid 3D point cloud registration algorithms," *Int. J. Adv. Intell. Syst*, vol. 8, pp. 118–127, 2015.

[24] F. Ferraguti, N. Preda, A. Manurung, M. Bonfè, O. Lambercy, R. Gassert, R. Muradore, P. Fiorini, and C. Secchi, "An Energy Tank-Based Interactive Control Architecture for Autonomous and Teleoperated Robotic Surgery," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1073–1088, Oct. 2015.