Haptic Aids for Bilateral Teleoperators

Alexander Pérez[1,2,⋆] and Jan Rosell[1]

[1] Institute of Industrial and Control Engineering, Universitat Politècnica de Catalunya, Barcelona, Spain
jan.rosell@upc.edu

[2] Escuela Colombiana de Ingeniería Julio Garavito, Bogotá D.C., Colombia
alexander.perez@escuelaing.edu.co

Abstract. Teleoperation of robotic tasks is usually performed in the Cartesian space due to the kinematic differences between the master and the slave. This entails several requirements, like the definition of a proper mapping between workspaces, the need to avoid collisions of the teleoperated robot with the environment, and the use of the inverse kinematics and of a procedure to correctly manage the passing through singularities. Within a bilateral teleoperation framework to teleoperate an industrial robot with a desktop haptic device, the present work proposes a guiding system based on path planning techniques to cope with these issues. The proposed system also includes a reactive behavior to cope with the potential collisions with obstacles. Teleoperation tests on virtual and real scenarios are included to validate the approach.

Keywords: Teleoperation, motion planning, guidance system, haptic devices.

1 Introduction

Teleoperation has evolved in many aspects since its beginning in the late 40s [1], thus enlarging the kind of tasks that can be teleoperated. Recent advances, for instance, allow the design of bilateral teleoperation frameworks using communication channels with variable time delays, like the internet. These bilateral teleoperation frameworks require the use of force feedback devices, called haptic devices, as the master part in the teleoperation. Using these devices, teleoperation tasks are performed at Cartesian level, avoiding incompatibilities with the slave robot from the kinematic point of view. However, this entails some problems that need to be correctly handled, and that have often been overlooked.

The first problem is related to the fact that the workspace of the robot and that of the haptic device have quite different sizes, being the haptic device workspace much smaller than that of the robot. A proper mapping between workspaces is required, and several approaches have been proposed so far, like simple scale factors, drift control [2] or resynchronization [3].

[⋆] This work was partially supported by the Spanish Government through the projects DPI2010-15446 and DPI2011-22471 and by the Generalitat de Catalunya through an AGAUR scholarship.

M.A. Armada et al. (eds.), *ROBOT2013: First Iberian Robotics Conference*,
Advances in Intelligent Systems and Computing 253,
DOI: 10.1007/978-3-319-03653-3_38, © Springer International Publishing Switzerland 2014

The second problem is due to the need to move the robot avoiding collisions with obstacles, which is difficult and non-intuitive because the robot is commanded in Cartesian space. When the task the user wants to teleoperate is known, path planning techniques can be used to guide him/her along a collision-free path, both visually and haptically.

The third problem arises when the robot is steered along paths that include some configuration changes, e.g. from elbow-up to elbow-down as shown in Fig. 1. Some teleoperation papers overlook this problem, others try to eliminate or minimize it by taking advantage of redundancy to assure large convex workspaces for singularity-free operation [4]. This problem can, however, be easily solved if a path is available and can be haptically suggested to the user. In this case, near configurations changes, the haptic suggestion can be strengthened, forcing the robot to cross the singularity in the desired way, i.e. as specified by the path.

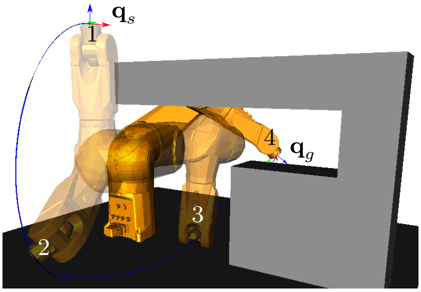


Fig. 1. Four snapshots of a task where where the arm configuration changes from elbow-down ($\mathbf{q}_s$) to elbow-up ($\mathbf{q}_g$)

Within a bilateral teleoperation framework to teleoperate an industrial robot with a desktop haptic device, the present work is focused on the proposal of a guiding system based on path planning techniques to cope with the above stated problems.

1.1 Related Works

The performance of teleoperation tasks can be significatively improved if some aids are provided to the operator. Several proposals in this line have already been published in the literature since the early nineties. A comprehensive overview of visual aids and cues to assist teleoperated tasks can be found in [5], which includes automatic end-effector tracking, computer assisted camera/lighting placement or the use of screen coordinates to solve the stimulus-response mismatch problem. The overview also presents display enhancements (later known as augmented reality) used: a) to highlight non-reachable or dangerous areas, or obstacles where an imminent collision can occur; b) to give visual cues to improve the orientation and positioning of the end effector; c) to present depth cues like virtual views and perspective grids. More recently, stereoscopic visual feedback systems that combine images gathered from two remotely actuated video cameras have been proposed to improve depth perception [6]. In order to improve the performance in complex scenarios with obstacles (although constrained to three *dof* robots), some approaches also proposed to use not only the view of the workspace but also that of the configuration space, where the user manipulates a single point, overcoming some usual human spatial reasoning deficiencies [7].

Besides the visual aids proposed, other aids based on force feedback greatly help to enhance the teleoperation performance, provided that the local site is controlled with a haptic device. Virtual fixtures, for instance, are used to restrain the user motions within a given region or within a subspace of lower dimension, thus allowing him/her to concentrate on the commanding of the motions relevant to the task, resulting in faster and more precise task executions and with lesser operator workloads. Some predefined virtual fixtures can be defined for given actions like approaching motions, or insert or extract operations, as introduced in [8]. Also, virtual fixtures can be specified by geometric constraint solvers, that find a subspace of lower dimension as a parameterized manifold that satisfies a constraints set, that is easily defined by the user from the knowledge of the task to be executed [9].

Force feedback is also used: a) to avoid obstacles, i.e. a force field proportional to the proximity between the obstacles and the end-effector is generated to repel the approaching motions to the obstacles, e.g. by using a map of the environment [10] or an easy recognition of obstacles by means of predefined markers [11]; b) to avoid singularities, i.e. a force field proportional to the proximity of a singularity and to the magnitude and direction of the velocity command is generated to drive the user into the feasible direction defined by the manipulator Jacobian [12]; c) to avoid large contact forces in tasks where there is physical interaction, like assembly tasks, i.e. contact forces computed from a model of the task can be fed back to make the user react on time and avoid huge actual forces on the remote site [13].

Forces have also been used to guide the user motions towards a specific goal, in order to obtain an easier and faster task commanding. The approach in [14] proposes a simple way to guide the operator position during a teleoperated grasping task, by using an attractive potential field obtained from visual information, and the approach in [15] generates an attractive force from the position and orientation errors between the current configuration of the robot griper and the desired one. Other path planning techniques have been used to consider also the avoidance of obstacles, in order to generate forces to constrain the operator along a collision-free path towards a goal configuration, either for commanding 2 *dof* motions of a mobile robot [16], or the motions of an object being manipulated in 2 or 3 *dof* of translation [17] (i.e. without considering the kinematics and the geometry of the robot that manipulates the object), or in virtual 6 *dof* assembly/disassembly tasks [18].

A combination of several of the above mentioned aids are considered in the bilateral teleoperation framework proposed in [19]. However, no approaches include, at the authors' knowledge, with guiding aids to teleoperate robot manipulators.

1.2 Scope, Problem Statement and Solution Overview

Excluding those teleoperated tasks with an exploratory aim carried out in unknown environments, many others are performed in known environments, although this knowledge may be incomplete and/or subject to uncertainty, reason that justifies the need for teleoperation. This knowledge of the task and of the environment (that can be updated as new information is acquired) can be used to support the operator with different aids. It can, for instance, be used by path planning techniques to suggest collision-free paths, as already found in early works of supervisory teleoperation control [20], where the system suggested paths, graphically shown to the user, to avoid collisions with obstacles and also to avoid traversing visually occluded zones by defining them as virtual obstacles. Others approaches, as reviewed in the previous subsection, used path planning techniques to generate forces to attract the motion towards the goal while avoiding collisions of the manipulated object with the environment. We are interested in extending these methods to take also into account potential collisions of the whole robot with the environment, in order to alleviate the operator burden and let him/her concentrate on the task. The proposal is to be framed within a bilateral teleoperation framework to teleoperate an industrial robot with a desktop haptic device.

The use of a desktop haptic device to feed back force to guide the user poses some questions. The human perception of force and torque information is highly integrated, and humans cannot neatly perceive them in a separate way. To evaluate this, a recent work made a thorough study to determine whether force and torque cues interact in haptic discrimination of force, torque and stiffness [21]. These authors found a significant interference across force and torque dimensions, and their experiments proved that they cannot be processed separately. Moreover, the usefulness of torque feedback in haptic manipulation tasks is highly dependant on the task, and for simple tasks it has been demonstrated that force feedback alone approximates the performance obtained with force and torque feedback, thus justifying the popularity of desktop haptic devices that have 6 positioning degrees of freedom but only 3 in force feedback [22]. To evaluate the use of forces to haptically guide the user in teleoperation tasks, we have studied the response of several users when being randomly stimulated by either forces, torques or combinations of both, using a Phantom haptic device with force and torque feedback. Except for the torque around the axis of the stylus, users tended to mistake torques by forces. Taking all this into account, in the present proposal we have decided that no force and torque are to be simultaneously feed back to the user as a guide, because it is very difficult for the users to understand the suggestions felt, and often respond to torques with translations instead of rotations. They can, however, be separately fed back because if users have the chance to select which kind of feedback to receive, either forces or torques, they are prepared to respond accordingly.

Then, the problem to be solved is the implementation of aids to haptically guide an operator in the commanding of a teleoperated task, within a bilateral teleoperation framework involving an industrial robot and a desktop haptic device. The task and the environment are assumed known (although maybe incomplete and subject to uncertainty), and the forces and torques will be feed back separately, as required by the user. Four aids are proposed, which give the teleoperator framework the following features: a) *assisted re-synchronization:* the mapping is based on the re-synchronization method and during the clutching the system haptically guides the user towards the best configuration to relocate the haptic device; b) *collision-free guidance:* a sampling-based planner is used in Configuration Space to plan a collision-free path that, through the direct kinematics and the workspace mapping, is used to generate forces to suggest the user the motions required to teleoperate the task; c) *robust guidance:* the control module guarantees that the solution path is exactly followed at the configuration changes, guaranteeing a safe singularity crossing; d) *reactive behavior:* the damping values of the control algorithm at the remote site are increased when potential collisions with either fixed or mobile obstacles may occur, slowing down the teleoperation and preventing collisions.

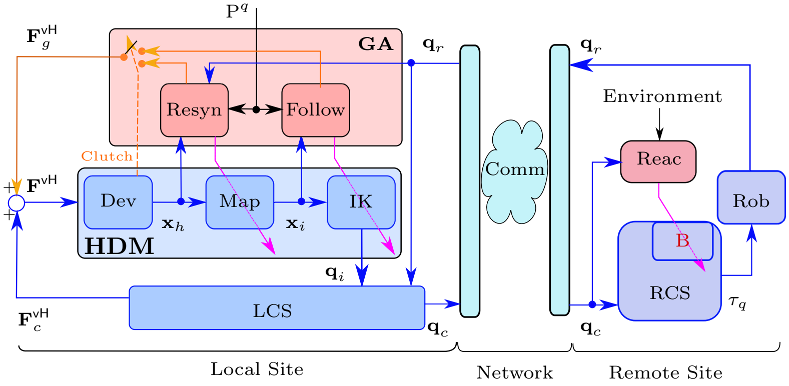
The paper is organized as follows. After this introduction, Section 2 sketches the assisted bilateral teleoperation framework proposed and Section 3 describes in detail the haptic aids. The theoretical contribution is complemented with performance evaluation in Section 4. Finally, Section 5 summarizes and concludes the proposal.

2 Teleoperation Framework

The bilateral teleoperation framework used is composed of the local site, the remote site and the communication layer, as shown in Fig. 2. The basic blocs, drawn in blue, comprise at the local site the haptic device management (HDM) and the local control system (LCS), and at the remote site the robot (Rob) and the remote control system (RCS). The bilateral control scheme and the workspace mapping between the haptic device and the robot are briefly introduced in Sections 2.1 and 2.2, respectively. The aids proposed as assistance correspond to the blocs drawn in red and comprise the resynchronization and path following guiding aids at the local site and the reactive behaviour at the remote site. The assisted teleoperation procedure is sketched in Section 2.3, and the aids used are fully detailed in Section 3.

2.1 Control

The blue signals between blocks in Fig. 2 define the basic control loop, based on the impedance strategy, that uses the position input to provide force feedback. The controller is a P+d (proportional plus damping) that was proved to be stable and robust in front of time delays [24]. Moreover, this controller was proven to provide stiff force reflections of the remote environment, and for small time-delays a better transparency than other alternatives. For these advantages, this control scheme has been chosen in the present work.



$\mathbf{F}_g^{\mathsf{vH}}$: Guiding force	$\mathbf{F}_c^{\mathsf{vH}}$: Control force	$\mathbf{F}^{\mathsf{vH}}$: Haptic force
$\mathbf{x}_h, \mathbf{x}_i$: Cur./Map. haptic pos.	$\mathbf{q}_r$: Real robot config.	P^q: Guided path
$\mathbf{q}_i$: Guided config.	$\mathbf{q}_c$: Control config.	τ_q: Control torque

HDM:	Haptic Device Management	**GA**:	Guiding Aids
LCS:	Local Control System	RCS:	Remote Control System
Dev:	Haptic Device	Rob:	Robot
Map:	Workspaces Mapping Module	Reac:	Reactive Module
IK:	Inverse Kinematics Module	B:	Damping Factor
Resyn:	Resynchronization Module		
Follow:	Path Following Module		

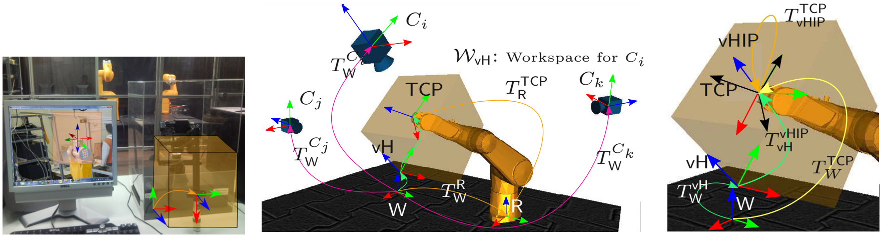
Fig. 2. General scheme of the teleoperation framework and its information flow

2.2 Workspace Mapping

The teleoperation framework requires the mapping between the workspace of the haptic device and that of the robot, i.e. the mapping between the movements of the Haptic Interface Point (HIP) and those of the robot Tool Center Point (TCP). The mapping used is based on the procedure proposed by the authors in [23] and summarize below, that uses the camera that feeds back the video from the scene as a common point between the local and the remote sites, as shown in Fig. 3a. This procedure maps the workspace of the haptic device ($\mathcal{W}_H$, with reference frame H) into that of the robot ($\mathcal{W}_W$, with reference frame W) to obtain the virtual haptic workspace ($\mathcal{W}_{\mathsf{vH}}$, with reference frame vH), following two guidelines: a) the orientation of vH is the same as that of the camera used to feed back the video (Fig. 3b), and b) the origin of the HIP within vH, called vHIP, is made coincident with that of the TCP (Fig. 3c).

The location of the HIP with respect to H, i.e. the transform $T_{\mathsf{H}}^{\mathsf{HIP}}$, is continually read from the haptic device, and the transform $T_{\mathsf{vH}}^{\mathsf{vHIP}}$ is made coincident with it or modified by a simple scale factor. Then, during the teleoperation activity the location of the TCP with respect to W is obtained as (Fig. 3c):

$$T_W^{\mathsf{TCP}} = T_{\mathsf{W}}^{\mathsf{vH}} \cdot T_{\mathsf{vH}}^{\mathsf{vHIP}} \cdot T_{\mathsf{vHIP}}^{\mathsf{TCP}}, \tag{1}$$


C_i
$T_W^{C_i}$
$\mathcal{W}_{\mathsf{vH}}$: Workspace for C_i
C_j
$T_W^{C_j}$
TCP
$T_{\mathsf{R}}^{\mathsf{TCP}}$
C_k
$T_W^{C_k}$
vH
$T_{\mathsf{W}}^{\mathsf{R}}$
W
R
$T_{\mathsf{vHIP}}^{\mathsf{TCP}}$
vHIP
TCP
$T_{\mathsf{vH}}^{\mathsf{vHIP}}$
vH
T_W^{TCP}
$T_{\mathsf{W}}^{\mathsf{vH}}$
W

(a) Haptic workspace.

(b) Virtual haptic workspace.

(c) Frames and transformations.

Fig. 3. Bilateral teleoperation framework equipped with an industrial Stäubli TX90 robot, a Phantom haptic device and three cameras: a) Physical devices and video feedback; b) Model of the remote site showing the virtual haptic workspace aligned with camera i; c) Virtual coupling attained by making the origins of frames vHIP and TCP coincident. Figures reproduced from [23].

where the transforms $T_{\mathsf{W}}^{\mathsf{vH}}$ and $T_{\mathsf{vHIP}}^{\mathsf{TCP}}$ are fixed and set at the synchronization instants as:

$$\begin{aligned} T_{\mathsf{vHIP}}^{\mathsf{TCP}} &= \left(\begin{array}{c|c} \mathcal{R}_{\mathsf{vHIP}}^{\mathsf{TCP}} & \mathbf{0} \\ \hline 0 & 1 \end{array}\right) \\ \mathcal{R}_{\mathsf{vHIP}}^{\mathsf{TCP}} &= (\mathcal{R}_{\mathsf{vH}}^{\mathsf{vHIP}})^{-1} \cdot (\mathcal{R}_{\mathsf{W}}^{\mathsf{Ci}})^{-1} \cdot \mathcal{R}_{\mathsf{W}}^{\mathsf{TCP}} \\ T_{\mathsf{W}}^{\mathsf{vH}} &= T_W^{\mathsf{TCP}} \cdot (T_{\mathsf{vHIP}}^{\mathsf{TCP}})^{-1} \cdot (T_{\mathsf{vH}}^{\mathsf{vHIP}})^{-1} \end{aligned} \qquad (2)$$

Re-synchronization is done by pressing the clutch button on the haptic device stylus each time the user reaches the limits of the haptic workspace or when he/she wants to change the camera used.

2.3 Assisted Teleoperation Procedure

Teleoperation assistance is introduced in the teleoperation framework by the modules shown as red blocs in Fig. 2. The Path Following Module allows the generation of forces for the haptic device ($\mathbf{F}_g^{\mathsf{vH}}$) that suggest the following of a collision-free path, and its connection with the Inverse Kinematic Module permits a robust and smooth configuration change, when necessary. When the robot is detached from the haptic device, the Resynchronization Module is used to generate forces to assist the re-synchronization. The Reactive Module in the remote site is used to increase the damping gain if potential collisions are detected.

The proposed teleoperation assistance is based on the assumption that the task to be teleoperated is known (i.e. the initial and goal configurations, q_s and the q_g respectively, as well as the model of the robot and of the workspace). Then, the teleoperation assistance is designed based on the availability of a collision-free path computed by a sampling-based planner (Probabilistic RoadMap – PRM) on the configuration space of the robot ($\mathcal{C}$), connecting q_s and q_g:

– Let $\mathbf{P}^q$ be such a path: $\mathbf{P}^q = \{\mathbf{q}_s, \mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_g\} \in \mathcal{C}$, where q_i are the configurations that have been collision-checked to validate the edges of the

Algorithm 1. Assisted teleoperation procedure

```
Require:
  $\mathbf{q}_s, \mathbf{q}_g$: Start and goal configurations
  $\delta$: Step size of the sampling-based planner

  $\mathbf{P}^q$ = Path_Planner($\mathbf{q}_s, \mathbf{q}_g, \delta$)
  $\mathbf{P}^x$ = FK($\mathbf{P}^q$)
  while teleoperation do
    $\mathbf{x}_i$ = MapHaptic()
    if clutch then
      $\mathbf{F}^{\mathsf{vH}}$ = Re-synchronization($\mathbf{x}_i$)
    else
      $\mathbf{q}_r$ = SubscribeRobotConfig()
      $\mathbf{F}_g^{\mathsf{W}}$ = ComputeForce($\mathbf{x}_i$)
      $\mathbf{F}_g^{\mathsf{vH}}$ = ChangeFrame($\mathbf{F}_g^{\mathsf{W}}$)
      $\mathbf{q}_i$ = IK_filtered($\mathbf{x}_i$)
      [$\mathbf{F}_c^{\mathsf{vH}}, \mathbf{q}_c$] = Control($\mathbf{q}_i, \mathbf{q}_r$)
      PublishRobotCommand($\mathbf{q}_c$)
      $\mathbf{F}^{\mathsf{vH}} = \mathbf{F}_g^{\mathsf{vH}} + \mathbf{F}_c^{\mathsf{vH}}$
    end if
    Apply $\mathbf{F}^{\mathsf{vH}}$ to the haptic device
  end while
```

path, i.e. $\mathbf{q}_j$ and $\mathbf{q}_{j+1}$ are close configurations separated a suficient small distance ϵ (i.e. no collision is assumed possible between two free configurations ϵ apart). The path $\mathbf{P}^q$ is piecewise linear, and when two consecutive configurations in $\mathbf{P}^q$ correspond to different kinematic configurations, then a linear interpolation is done and the resulting intermediate (singular) configuration, is added to $\mathbf{P}^q$.

- Let $\mathbf{P}^x = \{\mathbf{x}_s, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_g\} \in SE3$ be the path followed by the robot TCP corresponding to $\mathbf{P}^q$, computed by the forward kinematics of the robot, i.e. $\mathbf{x}_j$ is the position and orientation of the TCP when the robot is at $\mathbf{q}_j$. Since the configurations of the path are very close to each other, the path $\mathbf{P}^x$ can also be assumed to be piecewise linear, and each segment of the path $\overline{\mathbf{x}_j\mathbf{x}_{j+1}}$ is labeled with the kinematic configuration of $\mathbf{q}_j$.

Algorithm 1 sketches the cycle of the proposed assistance-based teleoperation procedure. It first computes the paths $\mathbf{P}^q$ and $\mathbf{P}^x$ and then the teleoperation loop starts. The algorithm uses the following functions:

- Path_Planner($\mathbf{q}_s, \mathbf{q}_g, \delta$): Returns a collision-free path $\mathbf{P}^q$ from $\mathbf{q}_s$ to $\mathbf{q}_g$ using a step size δ.
- FK($\mathbf{P}^q$): Uses the forward kinematics of the robot to obtain the path $\mathbf{P}^x$ from $\mathbf{P}^q$.

- MapHaptic: Uses Eq. (1) to compute the robot TCP, as detailed in Section 2.2.
- Re-sychronization($\mathbf{x}_i$): Uses Eq. (2) to set a new mapping between workspaces and returns the guiding forces generated to suggest the user the best place to re-synchronize, as introduced in Section 3.1.
- SubscribeRobotConfig: Receives the robot configuration ($\mathbf{q}_r$) from the remote site.
- ComputeForce($\mathbf{x}_i$): Computes the guiding force to be applied at $\mathbf{x}_i$ to follow a collision-free path, as introduced in Section 3.2.
- ChangeFrame($\mathbf{F}_g^{\mathsf{W}}$): Transforms $\mathbf{F}_g^{\mathsf{W}}$ from the robot frame to the vH frame.
- IK_filtered($\mathbf{x}_i$): Computes the robot command $\mathbf{q}_i$ from $\mathbf{x}_i$ using the inverse kinematics and a robust crossing of singular configurations, as introduced in Section 3.3.
- Control($\mathbf{x}_i, \mathbf{q}_r$): Runs the bilateral control algorithm.
- PublishRobotCommand($\mathbf{q}_c$): Sends the command to the robot at the remote site.

This teleoperation cycle can start again at user's will, i.e. a new path from the configuration where the robot may be can be queried at any time and used to assist the teleoperation form there.

3 Teleoperation Aids

This section describes the four aids proposed, that have been incorporated to the basic bilateral teleoperation framework. The focus is set at the generation of forces; the description of the planner is out of the scope of the present paper.

3.1 Re-Synchronization Aid

Since the haptic device workspace is usually much smaller than that of the robot, the mapping between workspaces needs to be continually updated. At user requirement (e.g. by pressing the button located at the stylus of the haptic device), the robot is detached from the haptic device. Let HIP_{curr} be the position of the HIP at this instant of time. Then, the **Re-synchronization** function used in Algorithm 1:

1. Computes the new position of the HIP (called HIP_{new}) from where the teleoperation can resume and proceed for as long as possible. This is done by computing the largest bounding box that satisfies that it contains the current position of the vHIP and as much points of the path $\mathbf{P}^x$ as possible, and that fits within the virtual haptic workspace. Then $\mathcal{W}_{\mathsf{vH}}$ is placed such that the bounding box is centered in it, and the corresponding new HIP position is computed (HIP_{new}).

2. Haptically guides the user from HIP_{curr} to HIP_{new}, i.e. generates forces along the direction defined by the position vector of T:

$$T = \left[T_{\mathsf{H}}^{\mathsf{HIP_{curr}}}\right]^{-1} T_{\mathsf{H}}^{\mathsf{HIP_{new}}} \tag{3}$$

The re-synchronization aid is implemented in the Re-synchronization Module shown in Fig. 2. Finer details of the procedure can be found in [23].

3.2 Guiding of Collision-Free Paths

The guiding aid has been conceived to fulfill the following requirements:

- No force and torque are to be simultaneously feed back to the user. The user must have the chance to select which kind of feedback to receive, either forces or torques, and hence be prepared to respond accordingly.
- The forces/torques fed back must guide the user toward the path, disappearing within a dead-zone around it (i.e. by giving up to the suggestions felt, the haptic device has to be correctly positioned/oriented on the path). Optionally, when the user is in the dead zone or near it, he/she must have the chance to select weather to receive or not a pushing force to indicate the sense of the path.

The following nomenclature needs to be defined. Let:

- $\mathbf{x} = \left(\begin{array}{c|c} \mathcal{R} & \mathbf{p} \\ \hline \mathbf{0} & 1 \end{array}\right)$ describe the configuration (position and orientation) of the TCP, $\mathbf{x}_i$ be the current one and $\mathbf{x}_d$ the configuration on the path closest to $\mathbf{x}_i$.
- $\mathbf{d}_t$ be the vector $\mathbf{d}_t = \mathbf{p}_d - \mathbf{p}_i$ and $(\mathbf{d}_r, \theta_r)$ the axis-angle representation of $\mathcal{R}_i^{-1}\mathcal{R}_d$.
- d_t be the translational distance between $\mathbf{x}_i$ and $\mathbf{x}_d$, i.e. $d_t = |\mathbf{d}_t|$, and d_r the rotational one defined as $d_r = \theta_r$.
- $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ be the nodes of $\mathbf{P}^x$ such that $\mathbf{x}_d$ lies between them, and $\mathbf{s}$ be the vector $\mathbf{s} = \mathbf{p}_{k+1} - \mathbf{p}_k$.
- ϵ_{t_n} and ϵ_{r_n} be, respectively, translational and rotational distance thresholds, with $n = \{1, 2, 3\}$.

Then, the **ComputeForce**($\mathbf{x}_i$) function used in Algorithm 1 computes a generalized guiding force $\mathbf{F}_g^{\mathsf{W}} = (\mathbf{f}_g^{\mathsf{W}}, \tau_g^{\mathsf{W}})^T$ from two components $\mathbf{F}_m$ and $\mathbf{F}_p$ such that:

- $\mathbf{F}_m = (\mathbf{f}_m, \tau_m)^T$ attracts the TCP to the path. Force and torque are separately fed back provided that the distances d_t and d_r lie above given thresholds.
- $\mathbf{F}_p = (\mathbf{f}_p, 0)^T$ pushes along $\mathbf{s}$ towards the next node in the path. This generalized force is generated if d_t lies below a given threshold and the user requests it.

The expressions of $\mathbf{f}_g^{\mathsf{W}}$ and τ_g^{W} are:

$$\mathbf{f}_g^{\mathsf{W}} = \begin{cases} \mathbf{f}_p \cdot \mathrm{A} & d_t < \epsilon_{t_1} \\ \mathbf{f}_m + \mathbf{f}_p \cdot \mathrm{A} & \epsilon_{t_1} < d_t < \epsilon_{t_2} \\ \mathbf{f}_m & d_t > \epsilon_{t_2} \end{cases}$$

$$\tau_g^{\mathsf{W}} = \begin{cases} 0 & d_r < \epsilon_{r_1} \\ \tau_m & d_r > \epsilon_{r_1} \end{cases}$$

$$\begin{aligned} \mathbf{f}_m &= \frac{\mathbf{d}_t}{|\mathbf{d}_t|} \min\left(F_{max}, F_{max} \frac{d_t - \epsilon_{t_1}}{\epsilon_{t_3} - \epsilon_{t_1}}\right) \\ \tau_m &= \frac{\mathbf{d}_r}{|\mathbf{d}_r|} \min\left(\tau_{max}, \tau_{max} \frac{d_r - \epsilon_{r_1}}{\epsilon_{r_3} - \epsilon_{r_1}}\right) \\ \mathbf{f}_p &= \text{Constant force along } \mathbf{s} \\ \mathrm{A} &= 1/0 \text{ (Enable/Disable active force)} \\ F_{max} &= \text{Max. force exerted by the device} \\ \tau_{max} &= \text{Max. torque exerted by the device} \end{aligned}$$

In order to apply the guided force to the teleoperator using the haptic device, the generalized force $\mathbf{F}_g^{\mathsf{W}}$ is mapped to the frame vH, by applying the virtual work principle. This is done by the **ChangeFrame**($\mathbf{F}_g^{\mathsf{W}}$) function in Algorithm 1:

$$\mathbf{F}_g^{\mathsf{vH}} = \begin{bmatrix} \mathbf{f}_g^{\mathsf{vH}} \\ \tau_g^{\mathsf{vH}} \end{bmatrix} = \begin{bmatrix} \mathcal{R}_{\mathsf{vH}}^{\mathsf{W}} & \mathbf{0} \\ \mathbf{0} & \mathcal{R}_{\mathsf{vH}}^{\mathsf{W}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{f}_g^{\mathsf{W}} \\ \tau_g^{\mathsf{W}} \end{bmatrix} \tag{4}$$

The guiding aid is implemented in the Path Following Module shown in Fig. 2.

3.3 Aid to Change the Kinematic Configuration

The **IK_filtered** function in Algorithm 1 computes the set point $\mathbf{q}_i$ for the control algorithm, from the input $\mathbf{x}_i$ set by the user with the haptic device, using the inverse kinematics solution closer to the guided path. The function includes a robust and smooth crossing of points where the kinematic configuration changes. This is done by modifying the configuration commanded with a saturation that produces a funnel effect that enforces the crossing of the singular configuration in the desired way, i.e. following the solution path.

Let Fig. 4 illustrate the nomenclature required to describe the procedure:

$\mathbf{q}_0$: is a configuration where the kinematic configuration changes.

$\mathbf{q}_p$,$\mathbf{q}_n$: are, with respect to $\mathbf{q}_0$, the previous and the next configurations in the path, respectively.

$\mathbf{u}_{p/n}$: are the unitary vectors with origin at $\mathbf{q}_0$ and pointing towards $\mathbf{q}_p$ and $\mathbf{q}_n$, respectively.

$\mathbf{u}_j$: is the unitary vector of the jth coordinate of the configuration space (that which corresponds to the jth joint).

$\mathbf{q}_i$: is the robot configuration corresponding to the user position $\mathbf{x}_i$. It is computed from $\mathbf{x}_i$ using the inverse kinematics and choosing the kinematic configuration of the segment of the path to which $\mathbf{x}_d$ (the point on the path closest to $\mathbf{x}_i$) belongs to.

$\mathbf{v}_i$: is the vector from $\mathbf{q}_0$ to $\mathbf{q}_i$.

H_ρ: is the hyper-ball of radius ρ centered at $\mathbf{q}_0$, it bounds the region where the funnel effect is felt.

$\mathbf{u}_{p_j/n_j}$**:** are the unitary vectors defined as:

$$\begin{aligned}\mathbf{u}_{p_j} &= \text{sign}(\mathbf{u}_p \cdot \mathbf{u}_j)\mathbf{u}_j \;\; \forall j \in 1,\ldots,d \\ \mathbf{u}_{n_j} &= \text{sign}(\mathbf{u}_n \cdot \mathbf{u}_j)\mathbf{u}_j \;\; \forall j \in 1,\ldots,d\end{aligned} \tag{5}$$

$R_{P/N}$**:** are the regions inside H_ρ that are defined by the positive linear combination of the vectors $\mathbf{u}_{p_j} \forall j \in 1,\ldots,d$ and $\mathbf{u}_{n_j} \forall j \in 1,\ldots,d$, respectively.

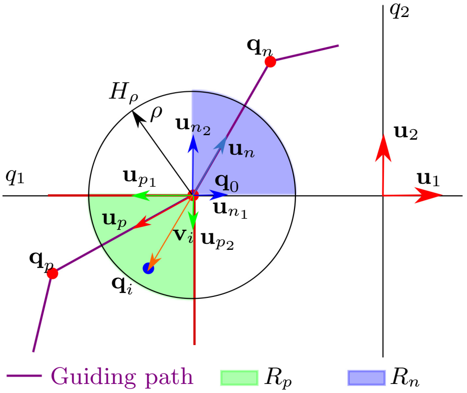


Fig. 4. Zones defined to correctly change the kinematic configuration in the 2 d.o.f. example of an RR robot shown

The aid is automatically activated when the configuration $\mathbf{q}_i$ lies within region H_ρ, modifying its value so as to make it lie within $R_P \cup R_N$, i.e. the coordinates of $\mathbf{q}_i$ are modified according to:

$$q_{i_j} = \begin{cases} q_{0_j} + v_{i_j} & \text{if } \mathbf{v}_i \cdot \mathbf{u}_{p_j} > 0 \text{ or } \mathbf{v}_i \cdot \mathbf{u}_{n_j} > 0 \\ q_{0_j} & \text{otherwise} \end{cases} \tag{6}$$

Fig. 5 shows an example where the guiding path crosses a point where the kinematic configuration changes, due to a change in the sign of q_2. The use of the proposed aid is as follows. Let the guiding path be $\mathbf{P}^q = \{\mathbf{q}_A, \mathbf{q}_B, \mathbf{q}_C, \mathbf{q}_D, \mathbf{q}_E\} \in \mathcal{C}$ and $\mathbf{P}^x$ be the corresponding path in $SE3$. Let also $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6\} \in SE3$ be the path commanded by the user with the haptic device and $\{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4, \mathbf{q}_5, \mathbf{q}_6\}$ the path in $\mathcal{C}$ computed using the same inverse kinematic solution as the corresponding closest point in $\mathbf{P}^x$. Configurations $\mathbf{q}_a$, $\mathbf{q}_b$, $\mathbf{q}_d$ and $\mathbf{q}_e$ already belong to either R_P or R_N and are therefore kept unchanged. However, $\mathbf{q}_3$ and $\mathbf{q}_4$ are first mapped, using Eq. (6), to $\mathbf{q}'_3$ and $\mathbf{q}'_4$ respectively, i.e. the path commanded to the robot is $\{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}'_3, \mathbf{q}'_4, \mathbf{q}_5, \mathbf{q}_6\}$. Therefore, the robot crosses the critical point as specified by the guiding path.

This aid is implemented in the Inverse Kinematic Module shown in Fig. 2.

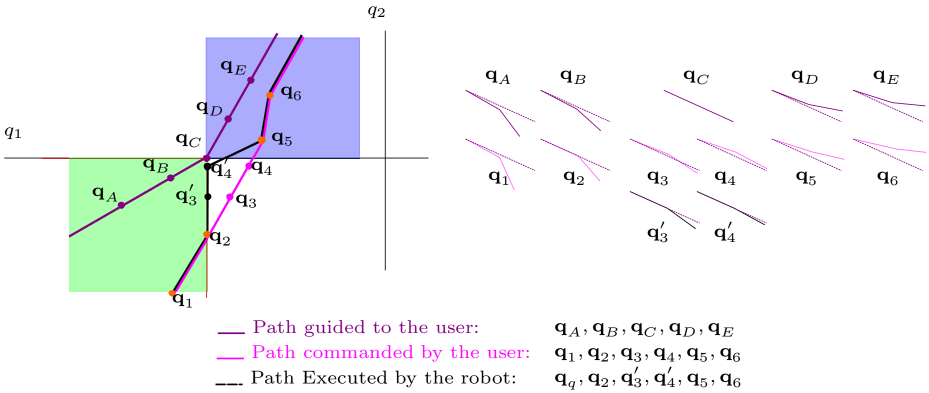

q_2
q_1
$\mathbf{q}_E$
$\mathbf{q}_D$
$\mathbf{q}_C$
$\mathbf{q}_B$
$\mathbf{q}_A$
$\mathbf{q}_6$
$\mathbf{q}_5$
$\mathbf{q}_4'$
$\mathbf{q}_4$
$\mathbf{q}_3'$
$\mathbf{q}_3$
$\mathbf{q}_2$
$\mathbf{q}_1$
$\mathbf{q}_A$ $\mathbf{q}_B$ $\mathbf{q}_C$ $\mathbf{q}_D$ $\mathbf{q}_E$
$\mathbf{q}_1$ $\mathbf{q}_2$ $\mathbf{q}_3$ $\mathbf{q}_4$ $\mathbf{q}_5$ $\mathbf{q}_6$
$\mathbf{q}_3'$ $\mathbf{q}_4'$
Path guided to the user: $\mathbf{q}_A, \mathbf{q}_B, \mathbf{q}_C, \mathbf{q}_D, \mathbf{q}_E$
Path commanded by the user: $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4, \mathbf{q}_5, \mathbf{q}_6$
Path Executed by the robot: $\mathbf{q}_q, \mathbf{q}_2, \mathbf{q}_3', \mathbf{q}_4', \mathbf{q}_5, \mathbf{q}_6$

Fig. 5. Guiding, commanded and executed paths both in configuration space (left) and physical space (right)

3.4 Reactive System

The proposed assistance is completed with the Reactive Module in the remote site, as shown in Fig. 2. This module is designed to regulate the remote damping of the control system, as a function of the potential collisions with the environment.

As demonstrated in [24], the use of the P+d controller guarantees that the velocities and the position error are bounded, provided that the control gains (the proportional and the damping gains at the local and the remote site, $K_l, B_l, K_r, B_r > 0$), are set according to:

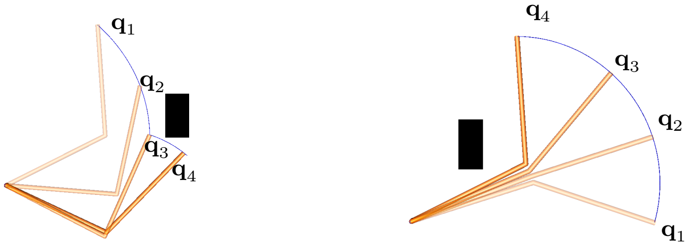
$$4B_lB_r > ({}^*T_l + {}^*T_r)^2K_lK_r, \tag{7}$$

assuming passive the human operator and the environment, and variable time-delays with known upper bounds ($T_i(t) \leq {}^*T_i < \infty$) and with time derivatives that do not vary faster than time itself ($|\dot{T}_i(t)| < 1$).

According to Eq. (7), assuming the local damping B_l and the gains K_l and K_r have a fixed value, the control scheme will be stable if the damping B_r is above a lower bound, B_r^{min}:

$$B_r^{min} = \frac{({}^*T_l + {}^*T_r)^2K_lK_r}{4B_l} \tag{8}$$

An increase of B_r above B_r^{min} will slow down the teleoperation, i.e. the user will feel an increased difficulty in moving the robot, and the system will remain stable.



(a) Without configuration change

(b) With configuration change

Fig. 6. Two instances of a 2D problem where the task to be teleoperated requires or not a configuration change in order to avoid the black obstacle

The Reactive Module has information of the obstacles of the environment, and therefore can compute the distance d between them and the robot, and make the value of B_r to decrease with d:

$$B_r = \begin{cases} B_r^{max} & d < d_{th} \\ B_r^{max} + \frac{B_r^{min} - B_r^{max}}{d_{cov} - d_{th}}(d - d_{th}) & d_{th} < d < d_{cov} \\ B_r^{min} & d > d_{cov} \end{cases} \tag{9}$$

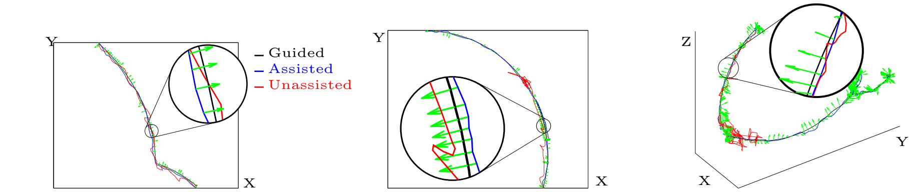
Where d_{th} is a fixed minimum allowable distance to the obstacles, $B_r^{max} > B_r^{min}$ the highest damping value set to slow down the motions as much as possible when the distance threshold d_{th} is exceeded, and d_{cov} the distance from where the effect of the obstacles disappears.

This proposal does not generate repulsive forces, and hence does not interfere with the guiding, while generating a reactive behaviour in front of potential collisions.

4 Evaluation

In order to test the performance of the guiding system, two scenarios have been defined with two tasks to be performed in each one, one requiring a configuration change and another not requiring it. The first scenario involves a RR 2 link planar robot (Fig. 6), and the second one the industrial robot Stäubli TX90 (Fig. 1), that has been used both in simulation and in real experiments. All the tasks have been teleoperated with and without using the guiding aid. To ease the visualization of results, the guiding aid has been restricted to force feedback (i.e. the torque feedback is disabled). The threshold values used for 2D task are $\epsilon_{t1} = 1$, $\epsilon_{t2} = 5$, $\epsilon_{t3} = 10$ and $\rho = 5$ and for the 3D task are $\epsilon_{t1} = 5$, $\epsilon_{t2} = 30$, $\epsilon_{t3} = 50$ and $\rho = 30$ (all values are in mm.).

Fig. 7a shows the data acquired while teleoperating the task in Fig. 6a, where it can be appreciated that the path followed with the guiding aid is smoother.



(a) 2D task without configuration change (b) 2D task with configuration change (c) 3D task with configuration change

Fig. 7. Guided and executed paths in Cartesian space, with and without aid, for the examples of Fig. 6 and 1. The guiding forces that drive the user to follow the guided path are shown with green arrows

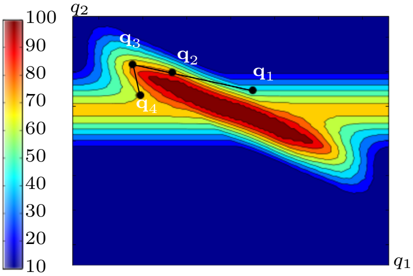


Fig. 8. Values of B_r computed by Eq. (9) on the configuration space corresponding to the task shown in Fig. 6a. The guided path is shown in black

Moreover, it could be executed a 27% faster (mean of 3 users). This figure also shows with green arrows how the generated forces drive the user to follow the guiding path. Fig. 8 illustrates for this example the values of B_r computed using $B_r^{max} = 100$, $B_r^{min} = 10$, $d_{th} = 2$ and $d_{cov} = 50$.

Fig. 7b shows the data acquired while teleoperating the task in Fig. 6b. In this case the task cannot be completed without guiding because it requires a change in the kinematic configuration. Using the guiding aid the task is easily completed, i.e. the user is able to change the configuration of the robot by simply passing near to the singular point, because the system aids to cross it correctly.

The 3D problem with the TX90 Staübli robot is more complex to command because the user has also to handle the orientation of the TCP, which is difficult and requires some training. Fig. 7c shows the data acquired while teleoperating the task in Fig. 1 with a real robot. In this case the task cannot be completed without guiding because a change in the kinematic configuration is also required.

5 Conclusions

The execution of teleoperated task with robots can be improved with guiding systems in order to aid the user to increase not only the velocity but also the safety. Within a bilateral teleoperation framework where an industrial robot is teleoperated by a desktop haptic device, a guiding system has been proposed based on a previously planned path and on the generation of forces to lead the user towards the path and to move along it. The guiding forces fed back are soft, i.e. the user is by no means restricted to move exactly along the path, and moreover any new path from the configuration where the robot may be can be queried at any time.

Also, the assistance proposed helps in the crossing of singularities often required during the execution of the task, which could not be done otherwise because the teleoperation is carried out at the cartesian level.

The bilateral teleoperation used is based on a proportional plus damping control approach that guarantees stability besides variable time delays, provided the coefficients satisfy a given condition. The assisted teleoperation framework proposed uses the damping factor at the remote site as a safety factor in front of possible collisions, i.e. teleoperation is slowed down by increasing this factor (above the minimum value that satisfies stability) when potential collisions with static or dynamic obstacles can occur. This reactive behavior does not interfere with the guiding.

The proposal has been validated with several users, showing that the assistance provided effectively increases the velocity and safety of the task executions, and eases the teleoperation burden.

References

1. Basañez, L., Suárez, R.: Teleoperation. In: Nof, S. (ed.) Springer Handbook of Automation, pp. 449–4685. Springer (2009)
2. Conti, F., Khatib, O.: Spanning large workspaces using small haptic devices. In: Proc. of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, pp. 183–188 (2005)
3. Azorin, J.M., Sabater, J.M., Sabater, J.M., Paya, L., Garcia, N.: Kinematics correspondence & scaling issues in virtual telerobotics systems. In: Proc. of the World Automation Congress, pp. 383–388 (2004)
4. Peer, A., Stanczyk, B., Buss, M.: Haptic Telemanipulation with Dissimilar Kinematics. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 3493–3498 (2005)
5. Brooks, T., Ince, I.: Operator vision aids for telerobotic assembly and servicing in space. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, vol. 1, pp. 886–891 (1992)
6. Portilla, H., Basañez, L.: Augmented reality tools for enhanced robotics teleoperation systems. In: 3DTV Conference, pp. 1–4 (May 2007)
7. Ivanisevic, I., Lumelsky, V.: Configuration space as a means for augmenting human performance in teleoperation tasks. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 30(3), 471–484 (2000)
8. Payandeh, S., Stanisic, Z.: On application of virtual fixtures as an aid for telemanipulation and training. In: Proc. of the 10th Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems, pp. 18–23 (2002)
9. Rodríguez, A., Nuño, E., Palomo, L., Basañez, L.: Nonlinear control and geometric constraint enforcement for teleoperated task execution. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 5251–5257 (2010)

10. Diolaiti, N., Melchiorri, C.: Obstacle avoidance for teleoperated mobile robots by means of haptic feedback. In: Proc. of the IEEE 1st Int. Workshop on Advances in Service Robotics (2003)
11. Park, H., Lim, Y.A., Pervez, A., Lee, B.C., Lee, S.G., Ryu, J.: Teleoperation of a multi-purpose robot over the internet using augmented reality. In: Proc. of the Int. Conf. on Control, Automation and Systems, pp. 2456–2461 (2007)
12. Maneewarn, T., Hannaford, B.: Augmented haptics of manipulator kinematic condition. In: Proc. SPIE, vol. 3840 (1999)
13. Rosell, J., Vázquez, I.: Haptic rendering of compliant motions using contact tracking in C-space. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 4223–4228 (2005)
14. Hua, J., Gu, D.: Haptically Guided Teleoperation System Based on Virtual Attractive Force. In: Int. Conf. on Mechatronics and Automation, ICMA 2009, pp. 2777–2782 (August 2009)
15. Hua, J., Wang, Y., Xi, N.: Attractive Force Guided Grasping for a Teleoperated Manipulator System Based on Monocular Vision. In: IEEE Int. Conf. on Control and Automation, ICCA 2007, May 30-June 1, pp. 2453–2457 (2007)
16. Jarvis, R.: Terrain-aware path guided mobile robot teleoperation in virtual and real space. In: Proc. of the 3rd Int. Conf. on Advances in Computer-Human Interactions, pp. 56–65 (2010)
17. Rosell, J., Vázquez, C., Pérez, A., Iñiguez, P.: Motion planning for haptic guidance. Journal of Intelligent and Robotic Systems 53(3), 223–245 (2008)
18. Ladeveze, N., Fourquet, J., Puel, B., Taix, M.: Haptic Assembly and Disassembly Task Assistance using Interactive Path Planning. In: IEEE Virtual Reality Conference, VR 2009, pp. 19–25 (March 2009)
19. Basañez, L., Rosell, J., Palomo, L., Nuño, E., Portilla, H.: A framework for robotized teleoperated tasks. In: Proc. of ROBOT 2011 Robótica Experimental, pp. 573–580 (2011)
20. Park, J., Sheridan, T.: Supervisory teleoperation control using computer graphics 1, 493–498 (1991)
21. Wu, B., Klatzky, R., Hollis, R.: Force, Torque, and Stiffness: Interactions in Perceptual Discrimination. IEEE Trans. on Haptics 4(3), 221–228 (2011)
22. Verner, L., Okamura, A.: Force & Torque Feedback vs Force Only Feedback. In: Third Joint EuroHaptics Conf. and Symp. on Haptic Interfaces for Virtual Environment and Teleoperator Systems, World Haptics 2009, pp. 406–410 (March 2009)
23. Pérez, A., Rosell, J.: An assisted re-synchronization method for robotic teleoperated tasks. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 886–891 (2011)
24. Nuño, E., Basañez, L., Prada, M.: Asymptotic stability of teleoperators with variable time-delays. In: Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 4332–4337 (2009)