Semi-autonomous Trajectory Generation for Mobile Robots with Integral Haptic Shared Control

Carlo Masone, Paolo Robuffo Giordano, Heinrich H. Bülthoff, and Antonio Franchi

Abstract— A new framework for semi-autonomous path planning for mobile robots that extends the classical paradigm of bilateral shared control is presented. The path is represented as a B-spline and the human operator can modify its shape by controlling the motion of a finite number of control points. An autonomous algorithm corrects in real time the human directives in order to facilitate path tracking for the mobile robot and ensures i) collision avoidance, ii) path regularity, and iii) attraction to nearby points of interest. A haptic feedback algorithm processes both human's and autonomous control terms, and their integrals, to provide an information of the mismatch between the path specified by the operator and the one corrected by the autonomous algorithm. The framework is validated with extensive experiments using a quadrotor UAV and a human in the loop with two haptic interfaces.

I. INTRODUCTION

Mobile robots can be deployed to remote or dangerous locations that are not accessible by human workers, thus making them valuable resources for many tasks such as exploration, monitoring and damage assessment. For example, in [1] underwater robots were teleoperated to inspect critical infrastructures by human pilots who could rely on a force feedback to perceive the environment.

The previous example is indicative of the fact that in real world applications robots are usually operated by human pilots, because they lack the sufficient cognitive capabilities to take complex decisions or to cope with unstructured environments. The example shows also that the human operator must be provided with cues that are informative of the state of the robots and of the remote environment [2]. In this regard, the adoption of a 'bilateral' interaction with haptic cues appears very promising, because: i) it was proven to increase the situation awareness and performance of the operator, see e.g., [3], [4]; ii) it requires little bandwidth in comparison to video streaming (see [5]), thus making it a good solution when there is limited bandwidth available, e.g., for intercontinental control of mobile robots over the internet [5] or remote control of underwater vehicles [1].

In accordance with these considerations, the *bilateral haptic shared control* framework [6] has been successfully applied to mobile robots, see e.g., [7], [8], [9], [10]. The classical bilateral teleoperation paradigm adopted in these works

C. Masone, H. H. Bülthoff and A. Franchi are with the Max Planck Institute for Biological Cybernetics, Spemannstraße 38, 72076 Tübingen, Germany {carlo.masone,hhb}@tuebingen.mpg.de.

P. Robuffo Giordano is with the CNRS at Irisa and Inria Rennes Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes Cedex, France prg@irisa.fr.

H. H. Bülthoff is also with the Department of Brain and Cognitive Engineering, Korea University,Seoul, 136-713 Korea.

A. Franchi is also with Centre National de la Recherche Scientifique (CNRS), Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS), 7 Avenue du Colonel Roche, 31077 Toulouse CEDEX 4, France. antonio.franchi@laas.fr

operates with a circular structure: the human operator directly commands the *current*[1] desired state of the robot(s) (e.g., position, velocity and acceleration); the robots try to execute the command and exploit their limited autonomy to achieve or preserve some desired objective (e.g., desired shape of the formation, obstacle avoidance); the force feedback closes the loop by giving back to the operator an information of the mismatch between commanded and actual motion. The drawback of this paradigm is that navigation of the robot(s), e.g., for a monitoring task, requires the human operator to always give motion directives, thus being very demanding.

In order to reduce the human's commitment, we introduced in [11] a novel shared control framework for persistent tasks in which a person modifies the 'closed' path that is autonomously followed by a single mobile robot, rather than piloting the robot itself. The human is assisted by an autonomous algorithm based on reactive deformations which ensures that the path is collision free or generates an alternative path. Reactive deformations allow for real-time implementation while the presence of the human alleviates the problems (e.g., local minima) implicit of purely reactive methods such as [12], [13], [14]. In this paper we extend [11] with several contributions:

1) The autonomous algorithm is extended to ensure path regularity and to prevent the presence of cusps.
2) The haptic feedback is better detailed.
3) The algorithm that generates alternative paths includes a new step (*Expansion*).
4) The framework is validated with experiments using a quadrotor and with both closed and open paths (in [11] the it was only tested in simulation with closed paths).

Note that in this paper we focus on path corrections and in practice we generate the trajectory by using a timing-law that modulates the traveling speed with the curvature of the path. A description of the timing-law algorithm is omitted because it is not the focus of the paper, however the problem of timing-law generation is widely treated in literature, see, e.g., [15]. The rest of the paper is organized as follows. Section II introduces the model of the path and of the environment. The outline of the framework is described in Sec. III and its components are detailed in Secs. III-A to III-C. The algorithm that autonomously plans new paths in presence of obstacles is detailed in Sec. IV. The experimental setup and results are presented in Sec. V.

II. PRELIMINARIES

We consider a single mobile robot with a characteristic point that is capable of traveling a sufficiently smooth and regular

---

[1]i.e., the state at the current time or in the very next future.

planar[2] path by always keeping a non-zero speed. This assumption relates to the concept of differential flatness [15] and applies to the large variety of mobile robots that are differentially flat with the characteristic point as (a part of) its flat output [16] or feedback linearizable with the characteristic point taken as linearizing output [17].

The robot is tasked to follow a path that is described by a B-spline [18], a linear combination of control points $\boldsymbol{x} = \left(\boldsymbol{x}_1^T \;\; \cdots \;\; \boldsymbol{x}_n^T\right)^T \in \mathbb{R}^{2n}$ and basis functions $B_j^\lambda : S \to \mathbb{R}$ for $j = 1, \ldots, n$ given as

$$\boldsymbol{\gamma}(\boldsymbol{x}, \cdot) : S \to \mathbb{R}^2$$
$$s \mapsto \sum_{j=1}^n \boldsymbol{x}_j B_j^\lambda(\boldsymbol{s}, s) = \boldsymbol{B}_{\boldsymbol{s}}(s)\boldsymbol{x} \quad (1)$$

where S is a compact subset of $\mathbb{R}$. The degree $\lambda > 0$ and knots $\boldsymbol{s} = (s_1, \ldots, s_l)$ are constant parameters chosen such that $\boldsymbol{\gamma}(\boldsymbol{x}, \cdot)$ is sufficiently smooth w.r.t. s. Further details on these parameters and on the computation of the basis functions can be found in [18], and in [11] for closed B-splines. The path corresponding to the B-spline curve $\boldsymbol{\gamma}(\boldsymbol{x}, \cdot)$ is

$$\boldsymbol{\gamma}_S(\boldsymbol{x}) = \{\boldsymbol{\gamma}(\boldsymbol{x}, s) \in \mathbb{R}^2 \,|\, s \in S\}, \quad (2)$$

i.e., the set of points $\boldsymbol{\gamma}(\boldsymbol{x}, s)$ obtained by varying the coordinate s within S. Note that the shape of the path is determined by $\boldsymbol{x}$. Regarding the regularity of $\boldsymbol{\gamma}_S(\boldsymbol{x})$, it is defined as follows.

Definition 1. *A point $\boldsymbol{\gamma}(\boldsymbol{x}, s)$ with $\boldsymbol{x} \in \mathbb{R}^{2n}$, $s \in S$ and such that $\left.\frac{\partial \boldsymbol{\gamma}}{\partial s}\right|_{(\boldsymbol{x}, s)} = (0\; 0)^T \in \mathbb{R}^2$ is called a singularity of $\boldsymbol{\gamma}_S(\boldsymbol{x})$. A path $\boldsymbol{\gamma}_S(\boldsymbol{x})$ without singularities is called regular.*

Geometrically, a singularity is a point where $\boldsymbol{\gamma}_S(\boldsymbol{x})$ could have a cusp, i.e., where the direction of motion (the unit tangent vector) vanishes. The following definition will be instrumental to ensure path regularity.

Definition 2. *Consider a regular path $\boldsymbol{\gamma}_S(\boldsymbol{x})$ with $S \subset \mathbb{R}$ and $\boldsymbol{x} = (\boldsymbol{x}_1^T \, \boldsymbol{x}_2^T \ldots \boldsymbol{x}_n^T)^T \in \mathbb{R}^{2n}$, and indicate with $\boldsymbol{x}_i^\star(\boldsymbol{x}, s) \in \mathbb{R}^2$, where $s \in S_i = \{s \in [s_{i-\lambda}, s_i) : \frac{d B_i^\lambda(\boldsymbol{s}, s)}{d s} \neq 0\}$[3], a point such that $\boldsymbol{\gamma}\left((\boldsymbol{x}_1^T \ldots \boldsymbol{x}_{i-1}^T \, \boldsymbol{x}_i^\star(\boldsymbol{x}, s)^T \, \boldsymbol{x}_{i+1}^T \ldots \boldsymbol{x}_n^T)^T, s\right)$ is a singularity. The 'singular curve' of the control point $\boldsymbol{x}_i \in \mathbb{R}^2$ is the collection of points $\Omega_i(\boldsymbol{x}) = \{\boldsymbol{x}_i^\star(\boldsymbol{x}, s) \,|\, s \in S_i\}$.*

From (1), it is straightforward to see that

$$\boldsymbol{x}_i^\star(\boldsymbol{x}, s) = -\frac{\sum_{j=1, j \neq i}^n \boldsymbol{x}_j \frac{d B_j^\lambda(\boldsymbol{s}, s)}{d s}}{\frac{d B_i^\lambda(\boldsymbol{s}, s)}{d s}}. \quad (3)$$

The singular curve Ω^i describes how the control point $\boldsymbol{x}_i$ of a regular $\boldsymbol{\gamma}_S(\boldsymbol{x})$ can be modified without creating singularities. These concepts are illustrated by the example of Fig. 1.

---

[2]The framework generalizes to $\mathbb{R}^3$, but we formulate our machinery in $\mathbb{R}^2$ to simplify the exposition and because the planar case is relevant *per se* in many real-world scenarios, e.g., an aircraft monitoring the earth surface while flying at a constant altitude.

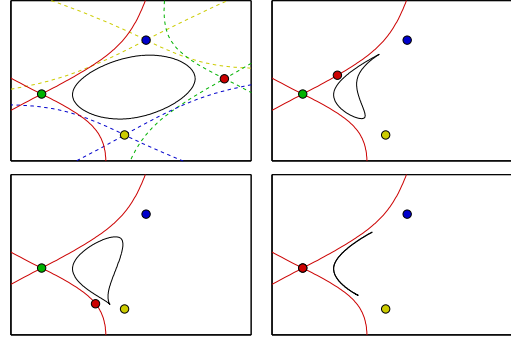[3]If $i - \lambda \leq 0$, use s_1 instead of $s_{i-\lambda}$.

Fig. 1: Example of a B-spline (black line) of degree $\lambda = 3$, with 4 control points (colored points). By moving one control point, the B-spline is made non-regular. Top-Left: initial regular B-spline and singular curves (colored lines) of the control points (with the same color pattern). The dashed lines are the singular curves of the fixed control points. Other boxes: the B-spline becomes non-regular when one control point (red one) is moved onto its singular curve.

The framework also accounts for the presence in the environment of points of interest (PoIs) and static obstacles. The points of interests are locations that are meaningful for the task, e.g., meeting points, fixed stations for data transfer, or victims to be rescued. These locations are modeled as a finite set of points $\mathcal{R} \in \mathbb{R}^{2 \times n_{\mathcal{R}}}$. The obstacles are modeled as balls centered in a finite set of static points $\mathcal{O} \in \mathbb{R}^{2 \times n_{\mathcal{O}}}$ and such that the robot is not in collision if the characteristic point lies outside all the obstacle balls. To simplify the notation, hereinafter we consider balls having all radius $R_{\mathcal{O}}$.

III. OVERVIEW OF THE PROPOSED FRAMEWORK

We assume that an initial regular path $\boldsymbol{\gamma}_S(\boldsymbol{x}_0)$ is given by a preliminary planning algorithm tailored for the task at hand[4]. Path modifications are achieved by introducing a time dependency in $\boldsymbol{x}$ so that $\boldsymbol{\gamma}_S(\boldsymbol{x}(t))$ becomes a time-varying path. The signal $\boldsymbol{x}(t)$ is generated online according to the following dynamical system

$$\dot{\boldsymbol{x}} = \boldsymbol{N}\left(\boldsymbol{u}_h + \boldsymbol{u}_a\right), \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0, \qquad \boldsymbol{N} \in \mathbb{R}^{2n \times 2n} \quad (4)$$

where $\boldsymbol{u}_h \in \mathbb{R}^{2n}$ is a term influenced by the human operator (described in Sec. III-A), and $\boldsymbol{u}_a \in \mathbb{R}^{2n}$ and $\boldsymbol{N} \in \mathbb{R}^{2n \times 2n}$ are two control terms generated by the autonomous algorithm (described in Sec. III-B). The proposed framework that computes the control terms in (4) is organized with a circular structure (see Fig. 2):

a) Human guidance: it steers $\boldsymbol{\gamma}_S(\boldsymbol{x})$ towards a desired path $\boldsymbol{\gamma}_S(\boldsymbol{x}_h)$ that is modified by an human operator using an actuated multi-DoFs tool as input device

b) Autonomous corrector: an autonomous algorithm corrects, if necessary, the human commands so that the actual path $\boldsymbol{\gamma}_S(\boldsymbol{x})$ remains regular, collision free and is attracted by nearby points of interest.

c) Haptic feedback: another algorithm closes the interaction-loop between human operator and autonomous correction algorithm by controlling the force exerted by the actuated input device. The force feedback physically informs the operator about the changes brought by the autonomous

---

[4]For example, it can be an exploration algorithm planning the next move based on the current partial map.
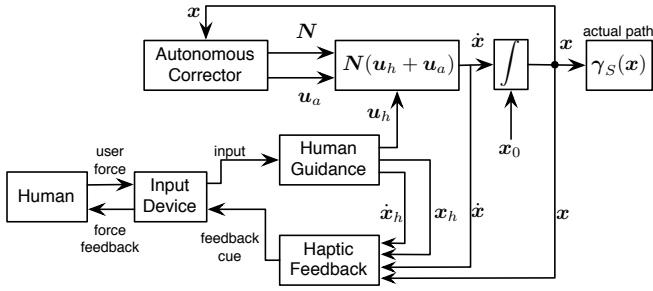
Fig. 2: Schematics of the framework.

(a) Translation

(b) Scaling w.r.t. $\bar{\boldsymbol{p}}_i$

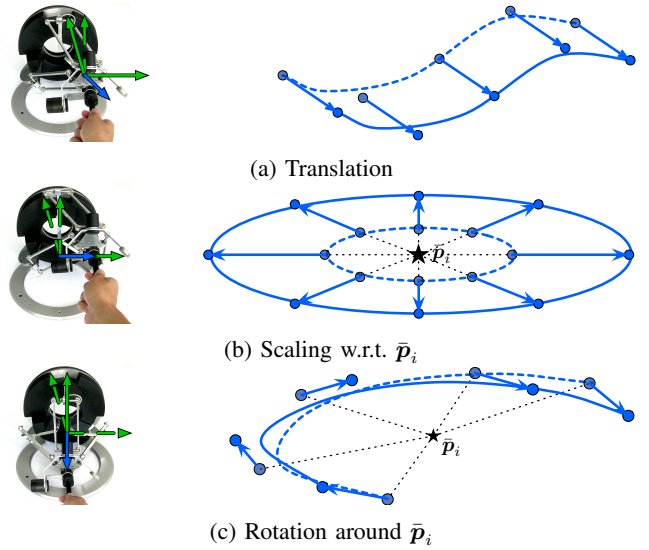(c) Rotation around $\bar{\boldsymbol{p}}_i$

Fig. 3: Examples of canonical path transformations applied to three different kind of paths. Green arrows represent the DoFs. Depending on the transformation the device motion is artificially restricted (continuous green arrows): 2 DoFs for translation, 1 DoFs for scaling, 1 DoFs for rotation. Blue arrows represent the commands and corresponding motion of the control points.

correction to his/her suggested modifications of the current path.

The three parts of the framework are thoroughly illustrated in Sections III-A, III-B and III-C respectively.

A. Human Guidance

The human operator interacts with the path by means of one or more input devices with $1 \leq m \leq 2n$ fully-actuated DoFs[5]. The devices are modeled as generic mechanical systems

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \boldsymbol{\tau} + \boldsymbol{\tau}_h \qquad (5)$$

where $\boldsymbol{q} \in \mathbb{R}^m$ is the configuration vector of the device, $M(\boldsymbol{q}) \in \mathbb{R}^{m \times m}$ is the positive-definite and symmetric inertia matrix, $C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} \in \mathbb{R}^m$ are the Coriolis and centrifugal terms, and $\boldsymbol{\tau}, \boldsymbol{\tau}_h \in \mathbb{R}^m$ are the control and human forces, respectively[6]. The computation of $\boldsymbol{\tau}$ is done by the haptic feedback algorithm described in Sec. III-C.

The operator controls the desired shape $\boldsymbol{\gamma}_S(\boldsymbol{x}_h)$ of the path, according to the following dynamical system

$$\dot{\boldsymbol{x}}_h = \boldsymbol{Q}(\boldsymbol{x}_h)\boldsymbol{K}\boldsymbol{q}, \qquad \boldsymbol{x}_h(0) = \boldsymbol{x}_0, \qquad (6)$$

where $\boldsymbol{K} \in \mathbb{R}^{m \times m}$ is a diagonal matrix of positive gains and $\boldsymbol{Q} : \mathbb{R}^{2n} \to \mathbb{R}^{2n \times m}$ is a nonlinear mapping. The term $\boldsymbol{u}_h$ in (4) steers the actual $\boldsymbol{x}$ towards the desired $\boldsymbol{x}_h$ by implementing a feedforward/proportional-like action

$$\boldsymbol{u}_h = \dot{\boldsymbol{x}}_h + k_h(\boldsymbol{x}_h - \boldsymbol{x}), \qquad (7)$$

with $k_h > 0$.

Matrix $\boldsymbol{Q}$ in (6) defines the path modifications that the operator can command. In order to provide an intuitive interface for the operator we design $\boldsymbol{Q}$ as the juxtaposition of l elementary matrices $\boldsymbol{Q}_i(\boldsymbol{x}_h) \in \mathbb{R}^{2n \times \nu_i}$ with $i = 1 \ldots l$ that implement 'canonical' transformations

$$\boldsymbol{Q}(\boldsymbol{x}_h) = \left(\boldsymbol{Q}_1(\boldsymbol{x}_h) \mid \quad \ldots \quad \mid \boldsymbol{Q}_l(\boldsymbol{x}_h) \right), \qquad (8)$$

where $1 \leq l \leq m$, $1 \leq \nu_i \leq m$, and $\sum_{i=1}^{l} \nu_i = m$. Partition (8) induces a corresponding partition of $\boldsymbol{q}$

$$\boldsymbol{q} = \left(\boldsymbol{q}_1^T \mid \quad \ldots \quad \mid \boldsymbol{q}_l^T \right)^T$$

[5]In practice, the number n of control points is much larger than the number of fully-actuated DoFs (m) because of mechanical limitations while the number n of control points that specifies a path easily reaches the hundreds even in simple cases.

[6]In the case of multiple input devices $\boldsymbol{q}$, $\boldsymbol{\tau}$ and $\boldsymbol{\tau}_h$ are obtained by stacking in columns the corresponding vectors of each device while M and C are block diagonal matrices.

with $\boldsymbol{q}_i \in \mathbb{R}^{\nu_i}$ for $i = 1 \ldots l$, so that each $\boldsymbol{q}_i$ is mapped by a a different elementary matrix $\boldsymbol{Q}_i$. Among the several choices allowed by this general formulation, we provide in the following three significant examples of elementary matrices $\boldsymbol{Q}_i$, corresponding to three canonical motions:

Translation: The configuration $\boldsymbol{q}_i \in \mathbb{R}^2$ is mapped to a translation of $\boldsymbol{\gamma}_S(\boldsymbol{x}_h)$ (see Fig. 3a) by the following elementary matrix

$$\boldsymbol{Q}_i(\boldsymbol{x}_h) = \boldsymbol{I}_{2n} \qquad (9)$$

where $\boldsymbol{I}_{2n}$ is the $2n \times 2n$ identity matrix. Notice that matrix $\boldsymbol{Q}_i$ does not depend on $\boldsymbol{x}_h$, since the velocity applied to each control point is a fixed scaled version of $\boldsymbol{q}_i$.

Scaling: The single DoF $\boldsymbol{q}_i \in \mathbb{R}$ commands a scaling of the path w.r.t. a given fixed point $\bar{\boldsymbol{p}}_i \in \mathbb{R}^2$ by means of the following elementary matrix

$$\boldsymbol{Q}_i(\boldsymbol{x}_h) = \boldsymbol{x}_h - \boldsymbol{1}_n \otimes \bar{\boldsymbol{p}}_i \qquad (10)$$

where $\boldsymbol{1}_n$ is an n-dimensional column vector of ones and $\otimes$ denotes the Kronecker product.

Rotation: The single DoF $\boldsymbol{q}_i \in \mathbb{R}$ commands a rotation of the control points around a given fixed point $\bar{\boldsymbol{p}}_i \in \mathbb{R}^2$ by means of the following elementary matrix

$$\boldsymbol{Q}_i(\boldsymbol{x}_h) = \operatorname{diag}(\underbrace{\bar{\boldsymbol{I}}_2, \ldots, \bar{\boldsymbol{I}}_2}_{n \text{ times}}) (\boldsymbol{x}_h - \boldsymbol{1}_n \otimes \bar{\boldsymbol{p}}_i) \qquad (11)$$

where $\bar{\boldsymbol{I}}_2 = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$.

B. Autonomous Corrector

The autonomous corrector must ensure that the actual path $\boldsymbol{\gamma}_S(\boldsymbol{x})$ satisfies the following requirements:

Objective 1. *The distance between any obstacle point $\boldsymbol{o} \in \mathcal{O}$ and $\boldsymbol{\gamma}_S(\boldsymbol{x})$ is always greater than $R_{\mathcal{O}}$.*

Objective 2. $\boldsymbol{\gamma}_S(\boldsymbol{x})$ *is regular.*
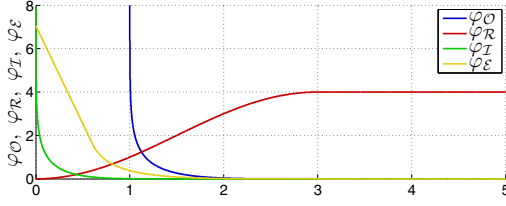
Fig. 4: Example of the artificial potentials $\varphi_{\mathcal{O}}$, $\varphi_{\mathcal{R}}$ and $\varphi_{\mathcal{I}}$ that are used to compute $\boldsymbol{u}_a$, and of potential $\varphi_{\mathcal{E}}$ used in Sec. IV

Objective 3. $\boldsymbol{\gamma}_S(\boldsymbol{x})$ *is attracted by nearby PoIs.*

In order to satisfy these Objectives $\boldsymbol{u}_a$ in (4) is designed as the sum of three terms

$$\boldsymbol{u}_a = \boldsymbol{u}_{a,\mathcal{O}}(\boldsymbol{x}, \mathcal{O}) + \boldsymbol{u}_{a,\mathcal{I}}(\boldsymbol{x}) + \boldsymbol{u}_{a,\mathcal{R}}(\boldsymbol{x}, \mathcal{R}). \quad (12)$$

Obstacle avoidance: $\boldsymbol{u}_{a,\mathcal{O}}$ is chosen as

$$\boldsymbol{u}_{a,\mathcal{O}} = -\sum_{o \in \mathcal{O}} \int_S \frac{\partial \boldsymbol{\gamma}(\boldsymbol{x}(t), s)}{\partial \boldsymbol{x}(t)}^\dagger \underbrace{\frac{\partial \varphi_{\mathcal{O}}(\|\boldsymbol{\gamma}(\boldsymbol{x}(t), s) - \boldsymbol{o}\|)}{\partial \boldsymbol{\gamma}(\boldsymbol{x}(t), s)}}_{\dot{\boldsymbol{p}}_o(s)} ds \quad (13)$$

where $\varphi_{\mathcal{O}} : \mathbb{R}_{\geq R_{\mathcal{O}}} \to \mathbb{R}_{\geq 0}$ is a smooth distance-based artificial potential function such that

$$\begin{array}{ll} \varphi_{\mathcal{O}} = 0 & \text{if } \|\boldsymbol{\gamma}(\boldsymbol{x}, s) - \boldsymbol{o}\| \geq \bar{R}_{\mathcal{O}} \\ \varphi_{\mathcal{O}} \to \infty & \text{if } \|\boldsymbol{\gamma}(\boldsymbol{x}, s) - \boldsymbol{o}\| \to R_{\mathcal{O}}^+ \end{array}$$

with $\bar{R}_{\mathcal{O}} > R_{\mathcal{O}}$. Furthermore, $\varphi_{\mathcal{O}}$ is strictly monotonic in $[R_{\mathcal{O}}, \bar{R}_{\mathcal{O}}]$ (see Fig. 4 for an example).

For every obstacle $\boldsymbol{o} \in \mathcal{O}$, $\boldsymbol{u}_{a,\mathcal{O}}$ applies to every point $\boldsymbol{\gamma}(\boldsymbol{x}(t), s) \in \boldsymbol{\gamma}_S(\boldsymbol{x})$ a repulsive velocity $\dot{\boldsymbol{p}}_o(s)$ that goes to infinity as $\boldsymbol{\gamma}(\boldsymbol{x}(t), s)$ goes towards the boundary $R_{\mathcal{O}}$ around $\boldsymbol{o}$. The repulsive velocity is then mapped to a velocity in the $\mathbb{R}^{2n}$ space of control points by inverting the relation $\dot{\boldsymbol{p}}_o(s) = \frac{\partial \boldsymbol{\gamma}(\boldsymbol{x}, s)}{\partial \boldsymbol{x}} \dot{\boldsymbol{x}}$ with the pseudo-inverse $\frac{\partial \boldsymbol{\gamma}(\boldsymbol{x}, s)}{\partial \boldsymbol{x}}^\dagger$. Finally, the line integral in (13) evaluates the effect of the artificial potential over all the points of the path.

Path regularity: $\boldsymbol{u}_{a,\mathcal{I}}$ is designed as

$$\boldsymbol{u}_{a,\mathcal{I}} = -\sum_{i=1}^n \sum_{j=1}^n \int_{S_i} \underbrace{\frac{\partial \varphi_{\mathcal{I}}(\|\boldsymbol{x}_i(t) - \boldsymbol{x}_i^\star(\boldsymbol{x}(t), s)\|)}{\partial \boldsymbol{x}_j(t)}}_{\dot{\boldsymbol{p}}_{\boldsymbol{x}_j}} ds \quad (14)$$

where and $\varphi_{\mathcal{I}} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a smooth distance-based artificial potential function such that

$$\begin{array}{ll} \varphi_{\mathcal{I}} = 0 & \text{if } \|\boldsymbol{x}_i - \boldsymbol{x}_i^\star\| \geq R_{\mathcal{I}} \\ \varphi_{\mathcal{I}} \to \infty & \text{if } \|\boldsymbol{x}_i - \boldsymbol{x}_i^\star\| \to 0^+ \end{array}$$

Furthermore, $\varphi_{\mathcal{I}}$ is strictly monotonic in $[0, R_{\mathcal{I}}]$ (see Fig. 4).

The action of (14) is twofold. If $j = i$, then it applies to $\boldsymbol{x}_i$ a velocity $\dot{\boldsymbol{p}}_{\boldsymbol{x}_i}$ that is directed away from $\boldsymbol{x}_i^\star(\boldsymbol{x}, s) \in \Omega_i(\boldsymbol{x})$. If $j \neq i$, then it applies to $\boldsymbol{x}_j$ a velocity $\dot{\boldsymbol{p}}_{\boldsymbol{x}_j}$ such that it moves $\boldsymbol{x}_i^\star(\boldsymbol{x}, s) \in \Omega_i(\boldsymbol{x})$ away from $\boldsymbol{x}_i$. In both cases, the intensity of the velocity goes to infinity as $\|\boldsymbol{x}_i - \boldsymbol{x}_i^\star\| \to 0$.

PoIs: $\boldsymbol{u}_{a,\mathcal{R}}$ is chosen as

$$\boldsymbol{u}_{a,\mathcal{R}} = -\sum_{\boldsymbol{r} \in \mathcal{R}} \left(\frac{\partial \boldsymbol{\gamma}(\boldsymbol{x}(t), s)}{\partial \boldsymbol{x}(t)}^\dagger \underbrace{\frac{\partial \varphi_{\mathcal{R}}(\|\boldsymbol{\gamma}(\boldsymbol{x}(t), s) - \boldsymbol{r}\|)}{\partial \boldsymbol{\gamma}(\boldsymbol{x}(t), s)}}_{\dot{\boldsymbol{p}}_{\boldsymbol{r}}(s)} \right) \bigg|_{\bar{s}_{\boldsymbol{r}}} \quad (15)$$

where $\bar{s}_{\boldsymbol{r}}$ indicates the closest point of $\boldsymbol{\gamma}_S(\boldsymbol{x})$ to $\boldsymbol{r}$ and $\varphi_{\mathcal{R}} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a smooth distance-based artificial potential function such that

$$\begin{array}{ll} \varphi_{\mathcal{R}} = 0 & \text{if } \|\boldsymbol{\gamma}(\boldsymbol{x}, s) - \boldsymbol{r}\| = 0 \\ \varphi_{\mathcal{R}} = U_{\mathcal{R}} > 0 & \text{if } \|\boldsymbol{\gamma}(\boldsymbol{x}, s) - \boldsymbol{r}\| \geq R_{\mathcal{R}} \end{array}$$

Furthermore, $\varphi_{\mathcal{R}}$ is strictly monotonic in $[0, R_{\mathcal{R}}]$ (see Fig. 4 for an example). Unlike $\varphi_{\mathcal{O}}$ and $\varphi_{\mathcal{I}}$, the action of $\varphi_{\mathcal{R}}$ is bounded because reaching the PoIs has a lower priority w.r.t. to the other objectives. However, the force feedback produced by the attractive action helps the operator to locate a PoI even if it is not reached. The structure of (15) is similar to that of $\boldsymbol{u}_{a,\mathcal{O}}$ (cf. (13)): the attractive velocity $\dot{\boldsymbol{p}}_{\boldsymbol{r}}(\bar{s})$ is applied to the point $\boldsymbol{\gamma}(\boldsymbol{x}, \bar{s}_{\boldsymbol{r}}) \in \boldsymbol{\gamma}_S(\boldsymbol{x})$ and then projected on the $\mathbb{R}^{2n}$ space of the control points using the pseudo-inverse $\frac{\partial \boldsymbol{\gamma}(\boldsymbol{x}, s)}{\partial \boldsymbol{x}}^\dagger$.

Despite the action of $\boldsymbol{u}_a$, the dependency of $\boldsymbol{u}_h$ from the exogenous human directive $\dot{\boldsymbol{x}}_h$ might produce path variations that are difficult to cope with for the robot. Moreover, since the time derivatives of $\dot{\boldsymbol{x}}_h$ are not available, then also the time derivatives of $\dot{\boldsymbol{x}}$ cannot be computed and therefore the signals $\frac{d^k \boldsymbol{\gamma}(\boldsymbol{x}(t), s(t))}{dt^k}$ with $k \geq 2$ cannot be used to control the robot[7].

To solve these problems we design $\boldsymbol{N}$ in (4) as

$$\boldsymbol{N} = \boldsymbol{I}_{2n} - \boldsymbol{J}^\dagger \boldsymbol{J} \quad (16)$$

where $\boldsymbol{I}_{2n} \in \mathbb{R}^{2n \times 2n}$ is the identity matrix and $\boldsymbol{J} \in \mathbb{R}^{2k \times 2n}$ with $k < n$ is

$$\boldsymbol{J}(\boldsymbol{x}, s) = \left(\frac{\partial \boldsymbol{\gamma}}{\partial \boldsymbol{x}}^T \quad \frac{\partial}{\partial \boldsymbol{x}} \left(\frac{\partial \boldsymbol{\gamma}}{\partial s} \right)^T \quad \cdots \quad \frac{\partial}{\partial \boldsymbol{x}} \left(\frac{\partial^k \boldsymbol{\gamma}}{\partial s^k} \right)^T \right)^T. \quad (17)$$

Jacobian $\boldsymbol{J}$ relates variations of $\boldsymbol{x}$ to changes of local geometric properties of the path in $s(t)$, such as the position of the point $\boldsymbol{\gamma}(\boldsymbol{x}(t), s(t))$, the tangent vector $\frac{\partial}{\partial s}\boldsymbol{\gamma}(\boldsymbol{x}(t), s(t))$, and so on. Matrix $\boldsymbol{N}$ defined in (16) is the well known orthogonal projection matrix [19] in the null-space of $\boldsymbol{J}$, i.e., it imposes that $\boldsymbol{J}\boldsymbol{N}(\boldsymbol{u}_h + \boldsymbol{u}_a) = \mathbb{0}_{2n}$. The intuitive interpretation to this design is that it imposes the invariance of the *local* geometric properties of the path at the current location of the robot regardless of the *global* changes brought by $\boldsymbol{u}_h$ and $\boldsymbol{u}_a$.

Formally, using the projection property of N, we can see that:

$$\frac{d\,\boldsymbol{\gamma}(\boldsymbol{x}(t), s(t))}{d\,t} = \underbrace{\frac{\partial\,\boldsymbol{\gamma}(\boldsymbol{x}, s)}{\partial\,\boldsymbol{x}}\dot{\boldsymbol{x}}}_{=\mathbb{0}_{2n}} + \frac{\partial\,\boldsymbol{\gamma}(\boldsymbol{x}, s)}{\partial\,s}\dot{s}$$

$$\frac{d^2\,\boldsymbol{\gamma}(\boldsymbol{x}(t), s(t))}{d\,t^2} = \underbrace{\frac{\partial}{\partial\,\boldsymbol{x}}\left(\frac{\partial\,\boldsymbol{\gamma}(\boldsymbol{x}, s)}{\partial\,s}\dot{s} \right)\dot{\boldsymbol{x}}}_{=\mathbb{0}_{2n}} + \sum_{j=1}^2 \frac{\partial}{\partial s^{(j-1)}}\left(\frac{\partial\,\boldsymbol{\gamma}(\boldsymbol{x}, s)}{\partial\,s}\dot{s} \right) s^{(j)}$$

$$(18)$$

It is straightforward to verify, by iterating with the chain rule, that also the higher order derivatives are independent from $\dot{\boldsymbol{x}}^k$ with $k \geq 1$. We omit writing these derivatives for the lack of space. In conclusion, $\boldsymbol{N}$ eliminates the effect of abrupt human commands and allows the computation of the signals $\frac{d^k\,\boldsymbol{\gamma}(\boldsymbol{x}(t), s(t))}{d\,t^k}$ with $k \geq 2$.

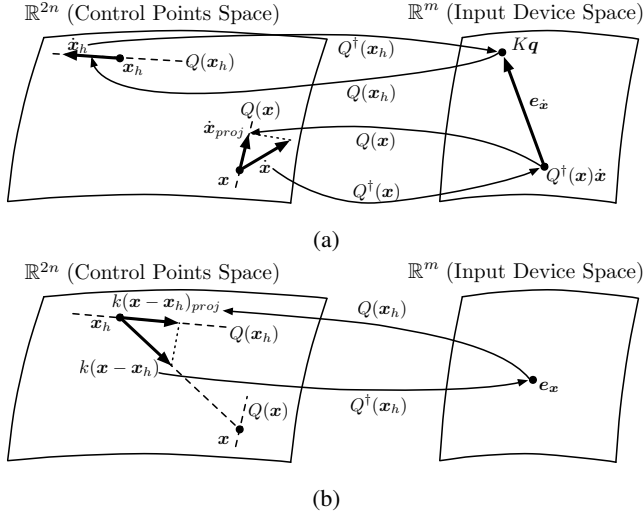[7]Under the assumption that a timing law $s(t)$ is provided with its time derivatives.

Fig. 5: Computation of the haptic feedback. a) First haptic cue, $e_{\dot{x}}$. b) Second haptic cue, e_x.

C. Haptic Feedback

We design τ as a function of two haptic cues, $e_{\dot{x}}$ and e_x.

First haptic cue: $e_{\dot{x}}$ is used to provide an information on how well the actual path $\gamma_S(x)$ is following the directives given by the operator. For example, if the human commands an expansion to $\gamma_S(x_h)$ then $e_{\dot{x}}$ should inform if the expansion rate of $\gamma_S(x)$ is different from the desired one. To compute $e_{\dot{x}}$, it is useful to start by making some observations with the help of Fig. 5a:

1) $\dot{x}_h$ is obtained by mapping the input device configuration Kq through the matrix $Q(x_h)$. Note that the map $Q(\cdot)$ depends on the application point and in general $Q(x_h) \neq Q(x)$ (see dashed lines in Fig. 5a).
2) $\dot{x}$ is generated by reactive actions and in general it does not correspond to any of the path modifications defined by the map $Q(\cdot)$, i.e., $\dot{x} \notin \Im(Q(x))$ where $\Im(\cdot)$ indicates the range space of a matrix.
3) Vector $\dot{x}_h - \dot{x}$ does not correspond in general to a path modification achievable through the map $Q(\cdot)$, i.e., $(\dot{x}_h - \dot{x}) \notin \Im(Q(x_h))$ and $(\dot{x}_h - \dot{x}) \notin \Im(Q(x))$.

In view of these considerations, a feedback proportional to $\dot{x}_h - \dot{x}$ would not give a good information on how well the operator's commands are executed. The approach that we adopt is to map $\dot{x}_h$ and $\dot{x}$ back onto the space of input device configurations $\mathbb{R}^m$ and only then compute the mismatch between the corresponding configurations. The haptic cue $e_{\dot{x}}$ is thus

$$e_{\dot{x}} = Q(x_h)^\dagger \dot{x}_h - Q(x)^\dagger \dot{x}$$
$$= Kq - Q(x)^\dagger \dot{x} \qquad (19)$$

where $Q(\cdot)^\dagger = (Q(\cdot)^T Q(\cdot))^{-1} Q(\cdot)^T$ and $Q(x)^\dagger \dot{x}$ is the mapping onto $\mathbb{R}^m$ of the orthogonal projection of $\dot{x}$ on $\Im(Q(x))$ (cf. [20]).

Second haptic cue: e_x is designed to provide a feedback that indicates how much $\gamma_S(x)$ differs in shape from $\gamma_S(x_h)$ and guides the operator in steering $\gamma_S(x_h)$ towards $\gamma_S(x)$. To achieve this result, we consider a velocity vector $k(x - x_h) \in \mathbb{R}^{2n}$, with $k > 0$, whose effect is to drive x_h towards x (see Fig. 5b). This fictitious velocity vector in general

does not correspond to a human input, i.e., $k(x - x_h) \notin \Im(Q(x_h))$, therefore we project it onto $\Im(Q(x_h))$ and then map it to a configuration of the input device. Hence, it is

$$e_x = kQ(x_h)^\dagger (x - x_h). \qquad (20)$$

Force feedback: τ is computed from $e_{\dot{x}}$ and e_x as

$$\tau = -B\dot{q} - K_M q - K^*(e_{\dot{x}} - e_x) \qquad (21)$$

where B is a positive definite damping matrix used to stabilize the device, K_M is a diagonal non-negative matrix used to provide a perception of the distance from the zero-commanded velocity[8], and K^* a diagonal positive definite matrix of gains. In order to guarantee stability despite the possible presence of non-modeled dynamics, communication delays and packet losses, we make use of the *passive set-position modulation* (PSPM) approach, a very general and flexible framework for guaranteeing stability (passivity) of the master side and of the closed-loop teleoperation system [21]. Let $\bar{z}[k]$ be the PSPM version of the following signal

$$z = Q(x)^\dagger \dot{x} - kQ(x_h)^\dagger (x - x_h), \qquad (22)$$

that is sampled and sent from the mobile robot to the haptic interface through the (possibly non-ideal) communication channel. Exploiting the PSPM action, the final *passive* implementation of τ in (21) then becomes

$$\tau = -B\dot{q} - K_M q - K^* (Kq - \bar{z}[k]). \qquad (23)$$

This is sufficient for guaranteeing stability (passivity) of the bilateral system assuming that the human operator behaves as a passive system (see [21] for more details).

IV. ALTERNATIVE PATHS IN PRESENCE OF OBSTACLES

The reactive approach used in $u_{a,\mathcal{O}}$ achieves collision avoidance but prevents the path from 'passing over an obstacle'. This well known limitation of reactive methods (see e.g. [13]) can degrade the operator's capability of steering the path. We tackle this problem by implementing a strategy that generates *alternative* paths in presence of obstacles and that is based on reactive deformations to allow real time implementation. The underlying idea is that, given an obstacle o and a collision free path $\gamma_S(x)$ between two points[9], we can find another collision free path $\gamma_S(x_o)$ with the same endpoints and *non-homotopic* to $\gamma_S(x)$ (i.e., it cannot be continuously morphed into $\gamma_S(x)$ without intersecting o). For each obstacle $o \in \mathcal{O}$, the computation of x_o is done in three steps:

Crossing: The algorithm starts when the reaction applied to $\gamma_S(x)$ by o is greater than a predefined threshold $\overline{F} > 0$, i.e.,

$$(Cond.\ \mathcal{C}1) \quad \begin{cases} \left\| \frac{\partial \varphi_\mathcal{O}(\|\gamma(x,\bar{s}) - o\|)}{\partial \gamma(x,\bar{s})} \right\| \geq \overline{F} \\ \text{s.t. } \bar{s} = \operatorname{argmin}_{s \in S} \|\gamma(x,s) - o\| \end{cases} \qquad (24)$$

---

[8]If this effect is not desired, one can alway disable it by taking $K_M = 0$.
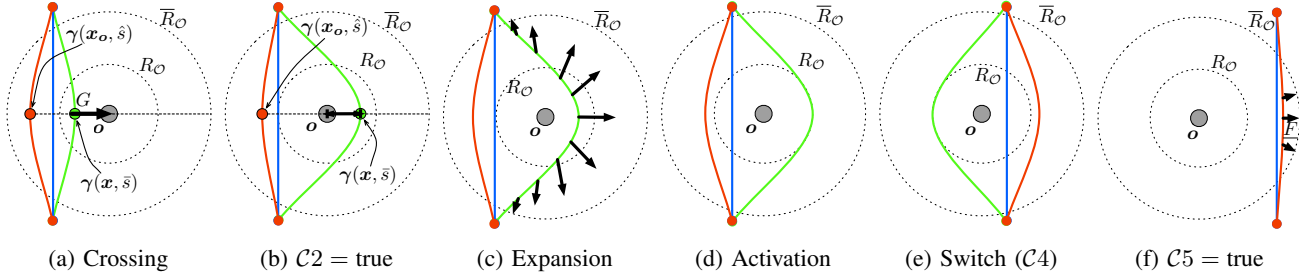[9]It can also be a portion of a path.

Fig. 6: Generation of an alternative path: $\gamma_S(x_o)$ (green line), $\gamma_S(x)$ (red line), $\gamma_S(x_h)$ (blue line), obstacle o (gray disc). From a) to f), $\gamma_S(x_h)$ is moving from left to right, passing over the obstacle.

When condition $\mathcal{C}1$ becomes true, let this be at time t_1, x_o is initialized and updated according to the following dynamic system

$$\dot{x}_o = \left.\frac{\partial \gamma(x, s)}{\partial x}\right|^{\dagger}_{(x_o, \hat{s})} G \frac{o - \gamma(x, \bar{s})}{\|o - \gamma(x, \bar{s})\|} \quad (25)$$

$$x_o(t_1) = x(t_1)$$

where $G > 0$ is a parameter and $\gamma(x_o, \hat{s})$ is the intersection between $\gamma_S(x_o)$ and the segment $o - \gamma(x, \bar{s})$. System (25) 'pulls' the point $\gamma(x_o, \hat{s}) \in \gamma_S(x_o)$ towards and beyond o, as illustrated in Fig. 6a.

Expansion: System (25) is active until $\gamma_S(x_o)$ becomes non-homotopic to $\gamma_S(x)$ w.r.t. o, i.e.,

$$(\text{Cond. } \mathcal{C}2) \quad \frac{(o - \gamma(x, \bar{s}))^T (\gamma(x_o, \hat{s}) - \gamma(x, \bar{s}))}{\|o - \gamma(x, \bar{s})\|^2} \geq 1 + F_c \quad (26)$$

where $F_c > 0$ is a user defined threshold. Condition (26) is illustrated in Fig. 6b. When (26) becomes true, since $\gamma_S(x_o)$ can be too close to the obstacle o, its evolution switches to

$$\dot{x}_o = -\int_S \left.\left(\frac{\partial \gamma(x,s)}{\partial x}^{\dagger} \frac{\partial \varphi_{\mathcal{E}}(\|\gamma(x,s) - o\|)}{\partial \gamma(x,s)}\right)\right|_{x_o(t)} ds \quad (27)$$

where $\varphi_{\mathcal{E}} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a smooth distance-based artificial potential function, that is strictly monotonic in $[0, \bar{R}_{\mathcal{O}}]$ and such that

$$\varphi_{\mathcal{E}} = 0 \quad \text{if } \|\gamma(x_o, s) - o\| \geq \bar{R}_{\mathcal{O}}$$
$$\varphi_{\mathcal{E}} \to U \quad \text{if } \|\gamma(x_o, s) - o\| \to 0^+$$

where $U > 0$ is a fixed parameter. An example for the design of $\varphi_{\mathcal{O}}^0$ is shown in Fig. 4. System (27) 'pushes' $\gamma_S(x_o)$ outside the ball of radius $R_{\mathcal{O}}$ and centered in o, as shown in Fig. 6c.

Activation: Once $\gamma_S(x_o)$ is collision free, i.e.,

$$(\text{Cond. } \mathcal{C}3) \quad \min_{s \in S} \|\gamma(x_o, s) - o\| > R_{\mathcal{O}} \quad (28)$$

the evolution of x_o switches to (4)[10] (see Fig. 6d). The alternative path $\gamma_S(x_o)$ becomes active (i.e., x and x_o are switched) only when closer to the desired $\gamma_S(x_h)$, i.e.,

$$(\text{Cond. } \mathcal{C}4) \quad \begin{cases} \|x_o - x_h\| < \|x - x_h\| \\ \|\gamma(x, s(t)) - \gamma(x_o, s(t))\| \simeq 0 \\ \quad\quad \vdots \\ \left\|\dfrac{d^k \gamma(x, s(t))}{d t^k} - \dfrac{d^k \gamma(x_o, s(t))}{d t^k}\right\| \simeq 0 \end{cases} \quad (29)$$

[10]Using x_o instead of x for the computation of N, u_h and u_a.

The conditions on the derivatives in $\mathcal{C}4$ are meant to prevent that the change from x to x_o causes discontinuities in the trajectory tracked by the robot, so that the initial requirement of a sufficiently smooth trajectory is still satisfied. The switch to the new path is depicted in Fig. 6e.

Finally, $\gamma_S(x_o)$ is deleted if the reaction applied to $\gamma_S(x)$ by o becomes smaller than a positive threshold $\underline{F} < \overline{F}$ (see Fig. 6f), i.e.,

$$(\text{Cond. } \mathcal{C}5) \quad \max_{s \in S} \left\|\frac{\partial \varphi_{\mathcal{O}}(\|\gamma(x, s) - o\|)}{\partial \gamma(x, s)}\right\| \leq \underline{F} \quad (30)$$

The algorithm presented here for a single obstacle has been used in practice with multiple obstacles. In particular we allowed the generation of a single alternative paths for each obstacle. Although not complete, this solution was very effective due to the rapidity of the algorithm.

V. EXPERIMENTS

The experimental setup consists, on the slave side, of a single quadrotor which internally implements the attitude and position control laws described in [10] and, on the master side, two force feedback devices (see Fig. 7). In particular, we used one device to command path translations (2 DoF) and the other to command path scalings (1 DoF) and rotations (1 DoF). The proposed framework is delegated to a PC running Matlab and which communicates via wireless ethernet to the low level flight control onboard the quadrotor. Finally, a motion capture system is used to measure the pose of the robot. Since the focus is on testing the action of the framework, the position of obstacles and PoIs is predefined. All the experiments are featured in the attached video[11].

Experiment 1: In this experiment we assess the contribution of the control term N in (4) by letting the operator command lateral translations (yellow arrow in Fig. 7b) to a straight reference path that is traveled by the quadrotor trying to keep a constant speed. The experiments is executed twice, first without using the control term N and recording the inputs provided by the operator, and then repeating it with the recorded inputs and activating N. Snapshots of the experiment in the two different conditions are presented in Figs. 8a and 8b. A simple visual inspection of the snapshots shows that:

1) without N, $\gamma_S(x)$ is identical to $\gamma_S(x_h)$ but the robot cannot track $\gamma(x(t), s(t))$ due to the human commands;

[11]The video is also available online at http://antoniofranchi.com/videos/trajgen_inthap.html

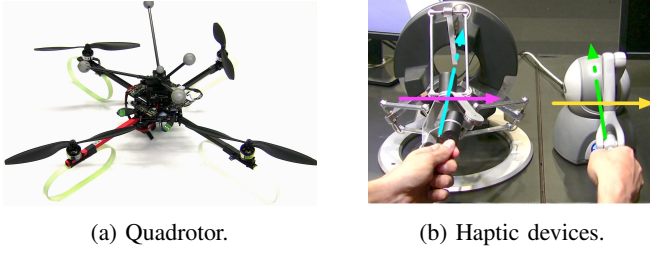(a) Quadrotor.

(b) Haptic devices.

Fig. 7: Experimental setup. a): the quadrotor is equipped with reflective markers that are used for the motion capture system. b): the device on the left (Omega.6) commands changes of scale (magenta direction) and rotations (cyan direction), the device on the right (PhantomOmni) commands 2D translations.
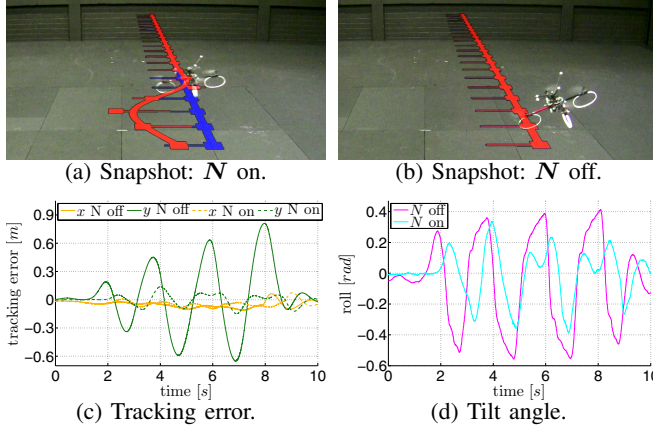

(a) Snapshot: N on.

(b) Snapshot: N off.

(c) Tracking error.

(d) Tilt angle.

Fig. 8: Experiment 1: Projection term N. In snapshots a) and b): overlay of the desired path $\gamma_S(x_h)$ in blue and of the actual path $\gamma_S(x)$ in red, with their corresponding control points x_h and x (blue and red squares) and their velocities $\dot{x}_h$ and $\dot{x}$ (blue and red thin lines).

2) with N, $\gamma_S(x)$ differs from $\gamma_S(x_h)$ and the robot can track $\gamma(x(t), s(t))$ quite precisely.

This analysis is confirmed by the plot of the tracking error of the trajectory $\gamma(x(t), s(t))$ in Fig. 8c. The experiment also shows that, when the projector N is used, the quadrotor has to tilt (roll) less (see Fig. 8d). This reduction in the tracking effort is due to the cancellation of the time derivatives of x that is produced by N, as described in Sec. III-B.

Experiment 2: This experiment showcases the action of the autonomous corrector when the path is steered by the operator in an environment populated by obstacles and a single PoI (a target placed on the floor). A downfacing camera mounted on the quadrotor confirms whether the PoI is actually reached. Snapshot 9a illustrates the initial configuration of the path and Snapshot 9a shows how the path has been teleoperated between the obstacles and the attractive action exerted by the PoI on the upper part of $\gamma_S(x)$. The commands given by the operator and the forces rendered on the input devices are depicted in Figs. 9c and 9d respectively. Once the PoI is within the range of action of φ_R (indicated by a black dashed line), the force feedback guides the operator in steering the path towards the target. Thanks to the automatic attractive force and to the operator's reaction to the force feedback the distance of $\gamma_S(x)$ from the PoI rapidly decreases once it is within the range R_R (see Fig. 9e). The video from the onboard camera confirms that the robot actually flies above the target (see Fig. 9f). At same
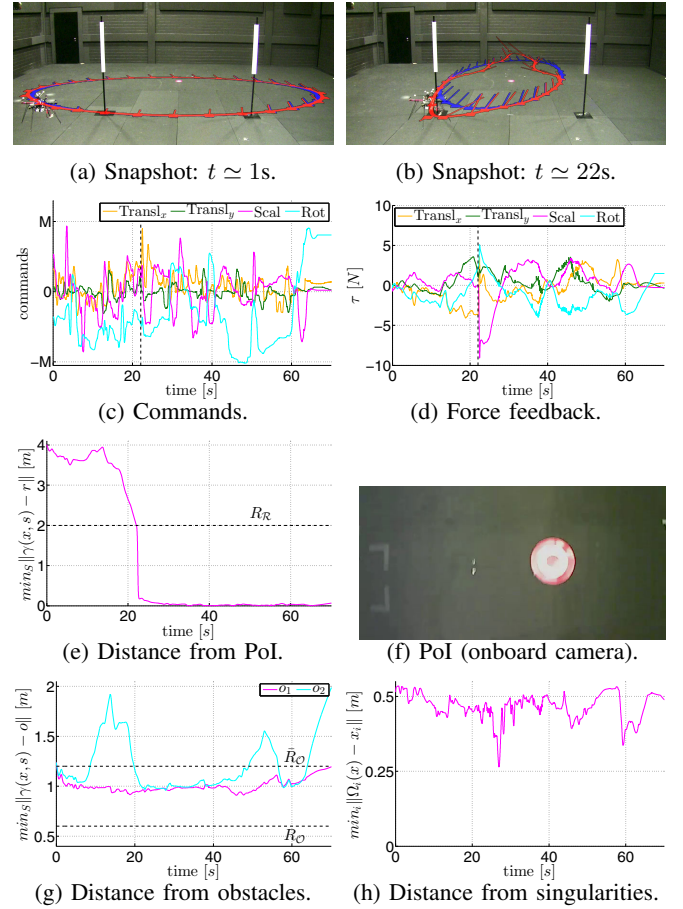

(a) Snapshot: $t \simeq 1$s.

(b) Snapshot: $t \simeq 22$s.

(c) Commands.

(d) Force feedback.

(e) Distance from PoI.

(f) PoI (onboard camera).

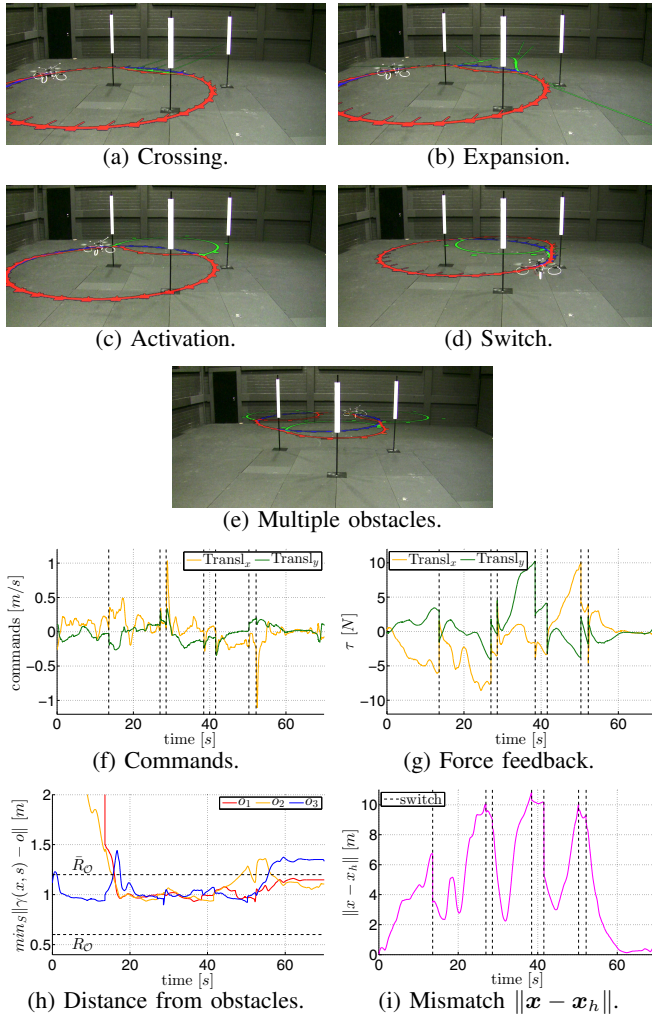(g) Distance from obstacles.

(h) Distance from singularities.

Fig. 9: Experiment 2. In snapshots a) and b): overlay of the desired path $\gamma_S(x_h)$ in blue and of the actual path $\gamma_S(x)$ in red, with their corresponding control points x_h and x (blue and red squares, respectively) and their velocities $\dot{x}_h$ and $\dot{x}$ (blue and red thin lines, respectively). In c) the limit M on the y axis is equal to 0.8 m/s for translations, 0.1 for the scaling rate and 7 deg/s for the rotation rate.

time, the distance of the path from the obstacles alway stays above the threshold R_O, as illustrated in Fig. 9g. Similarly, the minimum distance between the control points and the singular curves decreases when the path is forced between the obstacles but it never reaches zero (see Fig. 9h).

Experiment 3: In this last experiment the algorithm that generates alternative paths is activated to assist the operator who can only command translations. The environment is also populated by three obstacles and the path was chosen so that it is too big to pass between the obstacles. Snapshots 10a to 10d illustrate the various steps in the generation of an alternative paths. Snapshot 10e shows that the algorithm could generate simultaneously multiple alternative paths, one for each obstacle. The commands given by the operator and the forces rendered on the input devices are depicted in Figs. 10f and 10g respectively. Observe that path switches (black dashed lines) cause a jump in the forces due to the switch from x to x_o. The effect perceived is similar to stretching an elastic until it breaks. Similarly to Experiment 2, the autonomous corrector ensures that the distance between $\gamma_S(x)$ and the obstacles stays greater than R_O, even when multiple obstacles present at the same time (see Fig. 10h.) However, the use of the alternative paths reduces the mismatch $\|x - x_h\|$, as shown by Fig. 10i. Notice that

(a) Crossing.

(b) Expansion.

(c) Activation.

(d) Switch.

(e) Multiple obstacles.

(f) Commands.

(g) Force feedback.

(h) Distance from obstacles.

(i) Mismatch $\|\boldsymbol{x} - \boldsymbol{x}_h\|$.

Fig. 10: Experiment 3. In snapshots a)–e): overlay of the desired path $\boldsymbol{\gamma}_S(\boldsymbol{x}_h)$ in blue and of the actual path $\boldsymbol{\gamma}_S(\boldsymbol{x})$ in red, with their corresponding control points $\boldsymbol{x}_h$ and $\boldsymbol{x}$ (blue and red squares) and their velocities $\dot{\boldsymbol{x}}_h$ and $\dot{\boldsymbol{x}}$ (blue and red thin lines). The alternative paths $\boldsymbol{\gamma}_S(\boldsymbol{x}_o)$, control points $\boldsymbol{x}_o$ and their speed $\dot{\boldsymbol{x}}_o$ are drawn in green.

the error $\|\boldsymbol{x} - \boldsymbol{x}_h\|$ decreases after every path switch.

VI. CONCLUSIONS

This paper introduces a novel framework for semi-autonomous path corrections, which allows a human operator to modify the shape of the path traveled by a robot while an autonomous algorithm ensures obstacle avoidance, path regularity and assists the human in reaching meaningful locations. The algorithm is based on continuous reactive deformations and requires limited computational power so that is suitable for real time implementation. The framework also includes a haptic feedback algorithm in which the force is not explicitly given by the motion of the robot but rather depends on the teleoperated path. Experiments with a quadrotor UAV validated the framework. As a future extension of this work, we plan to formally study the stability of the closed-loop system. Furthermore, we plan to include the additional path constraints (e.g., minimum curvature) that arise for specific robots (e.g., fixed wing UAV). Lastly, we are working to use onboard sensing (e.g., cameras) to locate the obstacles.

REFERENCES

[1] R. R. Murphy, K. L. Dreger, S. Newsome, J. Rodocker, E. Steimle, T. Kimura, K. Makabe, , F. Matsuno, S. Tadokoro, and K. Kon, "Use of remotely operated marine vehicles at minamisanriku and rikuzentakata japan for disaster recovery," in *2011 IEEE Int. Symp. on Safety, Security and Rescue Robotics*, Kyoto, Japan, Nov. 2011, pp. 19–25.

[2] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035–2057, 2006.

[3] T. M. Lam, H. W. Boschloo, M. Mulder, and M. M. V. Paassen, "Artificial force field for haptic feedback in UAV teleoperation," *IEEE Trans. on Systems, Man, & Cybernetics. Part A: Systems & Humans*, vol. 39, no. 6, pp. 1316–1330, 2009.

[4] D. A. Abbink, M. Mulder, F. C. T. van der Helm, M. Mulder, and E. R. Boer, "Measuring neuromuscular control dynamics during car following with continuous haptic feedback," *IEEE Trans. on Systems, Man, & Cybernetics. Part B: Cybernetics*, vol. 41, no. 5, pp. 1239–1249, 2011.

[5] M. Riedel, A. Franchi, H. H. Bülthoff, P. Robuffo Giordano, and H. I. Son, "Experiments on intercontinental haptic control of multiple UAVs," in *12th Int. Conf. on Intelligent Autonomous Systems*, Jeju Island, Korea, Jun. 2012, pp. 227–238.

[6] G. Niemeyer, C. Preusche, and G. Hirzinger, "Telerobotics," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, pp. 741–757.

[7] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff, "A passivity-based decentralized strategy for generalized connectivity maintenance," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 299–323, 2013.

[8] A. Franchi, C. Secchi, M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano, "Shared control: Balancing autonomy and human assistance with a group of quadrotor UAVs," *IEEE Robotics & Automation Magazine, Special Issue on Aerial Robotics and the Quadrotor Platform*, vol. 19, no. 3, pp. 57–68, 2012.

[9] A. Franchi, C. Masone, V. Grabe, M. Ryll, H. H. Bülthoff, and P. Robuffo Giordano, "Modeling and control of UAV bearing-formations with bilateral high-level steering," *The International Journal of Robotics Research, Special Issue on 3D Exploration, Mapping, and Surveillance*, vol. 31, no. 12, pp. 1504–1525, 2012.

[10] D. J. Lee, A. Franchi, H. I. Son, H. H. Bülthoff, and P. Robuffo Giordano, "Semi-autonomous haptic teleoperation control architecture of multiple unmanned aerial vehicles," *IEEE/ASME Trans. on Mechatronics, Focused Section on Aerospace Mechatronics*, vol. 18, no. 4, pp. 1334–1345, 2013.

[11] C. Masone, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano, "Interactive planning of persistent trajectories for human-assisted navigation of mobile robots," in *2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vilamoura, Portugal, Oct. 2012, pp. 2641–2648.

[12] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in *1993 IEEE Int. Conf. on Robotics and Automation*, Atlanta, GA, May 1993, pp. 802–807.

[13] O. Brock and O. Khatib, "Elastic strips: A framework for motion generation in human environments," *The International Journal of Robotics Research*, vol. 21, no. 12, pp. 1031–1052, 2002.

[14] F. Lamiraux, D. Bonnafous, and O. Lefebvre, "Reactive path deformation for nonholonomic mobile robots," *IEEE Trans. on Robotics*, vol. 20, no. 6, pp. 967–977, 2004.

[15] T. Faulwasser, V. Hagenmeyer, and R. Findeisen, "Optimal exact path-following for constrained differentially flat systems," in *18th IFAC World Congress*, Milano, Italy, Aug. 2011, pp. 9875–9880.

[16] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of nonlinear systems: Introductory theory and examples," *International Journal of Control*, vol. 61, no. 6, pp. 1327–1361, 1995.

[17] A. Isidori, *Nonlinear Control Systems, 3rd edition*. Springer, 1995.

[18] L. Biagiotti and C. Melchiorri, *Trajectory Planning for Automatic Machines and Robots*. Springer, 2008.

[19] S. Chiaverini, G. Oriolo, and I. D. Walker, "Kinematically redundant manipulators," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, pp. 245–285.

[20] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. SIAM, 2001.

[21] D. J. Lee and K. Huang, "Passive-set-position-modulation framework for interactive robotic systems," *IEEE Trans. on Robotics*, vol. 26, no. 2, pp. 354–369, 2010.