

Robot Programming by Demonstration: Trajectory Learning Enhanced by sEMG-Based User Hand Stiffness Estimation

Luigi Biagiotti ¹, Member, IEEE, Roberto Meattini ², Member, IEEE, Davide Chiaravalli ¹, Gianluca Palli ¹, Senior Member, IEEE, and Claudio Melchiorri ¹, Fellow, IEEE

Abstract—Trajectory learning is one of the key components of robot Programming by Demonstration approaches, which in many cases, especially in industrial practice, aim at defining complex manipulation patterns. In order to enhance these methods, which are generally based on a physical interaction between the user and the robot, guided along the desired path, an additional input channel is considered in this article. The hand stiffness, that the operator continuously modulates during the demonstration, is estimated from the forearm surface electromyography and translated into a request for a higher or lower accuracy level. Then, a constrained optimization problem is built (and solved) in the framework of smoothing B-splines to obtain a minimum curvature trajectory approximating, in this manner, the taught path within the precision imposed by the user. Experimental tests in different applicative scenarios, involving both position and orientation, prove the benefits of the proposed approach in terms of the intuitiveness of the programming procedure for the human operator and characteristics of the final motion.

Index Terms—Constrained smoothing B-splines, human-robot interaction, programming by demonstration (PbD), surface electromyography (sEMG) signals.

I. INTRODUCTION

IN AN increasing number of applications, ranging from industrial to service robotics, the use of collaborative robots is becoming a crucial point and a key factor for competitiveness. The original paradigm related to robot usage has changed over the years from an initial scenario in which robots operated

Manuscript received 16 September 2022; revised 4 February 2023; accepted 20 February 2023. Date of publication 11 April 2023; date of current version 8 August 2023. This paper was recommended for publication by Associate Editor E. De Momi and Editor W. Burgard upon evaluation of the reviewers' comments. This work was supported in part by the European Union's Horizon 2020 research and innovation program under Grant 870133 as part of the RIA project REMODEL (Robotic tEchnologies for the Manipulation of cOmplex Deformable Linear objects) and in part by the project "FAR ATTREZZATURE 2021" of University of Modena and Reggio Emilia. (Corresponding author: Luigi Biagiotti.)

Luigi Biagiotti is with the Department of Engineering "Enzo Ferrari," University of Modena and Reggio Emilia, 41125 Modena, Italy (e-mail: luigi.biagiotti@unimore.it).

Roberto Meattini, Davide Chiaravalli, Gianluca Palli, and Claudio Melchiorri are with the Department of Electrical, Electronic and Information Engineering "Guglielmo Marconi," University of Bologna, 40136 Bologna, Italy (e-mail: roberto.meattini2@unibo.it; davide.chiaravalli2@unibo.it; gianluca.palli@unibo.it; claudio.melchiorri@unibo.it).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TRO.2023.3258669>.

Digital Object Identifier 10.1109/TRO.2023.3258669

autonomously in a separate cell to a scenario in which robots and humans work together and share the same workspace. Accordingly, methods for robot programming have changed as well. Traditional approaches for robot programming, based on offline programming and simulation and on the use of the teach pendant to define the sequence of desired configurations to be reached, have been supported, and in some cases replaced, by various methods that can be classified in a broad sense as programming by demonstrations (PbD) [1]. In PbD, the user directly trains the robot to perform the desired task with different forms of interaction and different levels of abstraction. These modalities of interacting with robots allow reducing the time needed for programming a task while minimizing users' errors [2]. Moreover, they also enable nonexpert users to program robots and are, therefore, one of the key technologies that can foster the use of robotic systems, especially in small- and medium-sized companies.

If the medium/long-term goal of PbD of robotic systems aims at reproducing the training processes that typically occur among humans [1], based, for instance, on skills transfer by means of practical demonstrations and/or vocal instructions, in industrial applications, PbD methods, such as lead-through programming or walk-through programming [3], [4], [5], which consist in manually guiding the robot, recording, and then reproducing its motion, are still the most widespread. Even if these approaches date back to the early 1980 s, when some painting robots could be programmed by manual guidance [6], it is in the last decade that, thanks to the availability on the market of collaborative robots specifically designed to safely interact with humans, programming via direct kinesthetic teaching has become a real option in many applications. However, the interaction between the user and the robot is often limited to a force exchange, and the robot task is a mere reproduction of the imposed trajectories.

A. Related Work

PbD can be categorized according to different features: the demonstration can be performed on a virtual or on a real environment, the user must hand guide the robot or simply perform the task with some kind of tracking system, and different input channels can be used (e.g., visual, kinesthetic, speech, etc.). Moreover, the task can be encoded in different ways: typical solutions are based on parametric curves, such as B-splines

or NURBS [7], or on different types of probabilistic methods, e.g., Gaussian mixture regression [8], [9], dynamical movement primitives (DMPs) [10], probabilistic movement primitives [11], kernelized movement primitives [12], or Gaussian process regression [13]. Despite the different features, a common characteristic of almost all the proposed approaches is the need for suppressing noise and unwanted/unnecessary motions that inevitably affect human demonstrations. To cope with these issues, the basic solution consists in recording multiple demonstrations, which are performed by an expert user, and then applying some kind of averaging or optimization method. In the context of probabilistic approaches, the differences among the input trajectories are translated into different variance levels of the stochastic processes used to model the task [13], [14], but alternative methods are also possible. In [15], variations between multiple demonstrations performed by a human operator are exploited to define the boundaries of a region free from obstacles. Then, a piecewise linear segment trajectory, fully contained in this admissible region, is built up to minimize the geometric path of the robot. In this approach, the human inconsistency in repeating the same desired trajectory is not treated as a noise to be canceled but is considered a measure of accuracy required by the task. A similar approach is proposed in [16], where several trajectories obtained by observing a human operator that replicates a given task are discretized in space and then condensed in a sequence of pairs composed by the average point and the standard deviation. On the basis of these pairs, a feasible space is constructed, and then, different trajectories for a robotic system are computed optimizing criteria such as the total length, the execution time, or the energy consumption. In [17], the motions performed by a force-controlled robot directly commanded by the human operator are used to define a safe volume, i.e., a volume free from obstacles, where the trajectory, which is needed for a given task, can be planned according to a given optimization criterium. In [18], NURBS curves are used for smoothing the trajectories learned via a user demonstration in a virtual environment. Spline curves are the tool used in [19] to design a smooth trajectory from the user demonstration. In this article, the reduction of the points of the input trajectory, to make the interpolation process less expensive from a computational point of view, has as a side effect a rejection of the noise affecting the motion.

As abovementioned, the generality of the PbD methods relies on multiple demonstrations of the same task, while only a few cases are based on a one-shot demonstration despite the clear advantages in terms of required effort and time for programming. In [20], the points of the recorded trajectory are modified by interpreting them as particles interacting among them and with the surrounding environment. In [21], the points acquired after the demonstration are used to define a continuously differentiable (polynomial) trajectory that approximates the piecewise linear path joining the points within a given tolerance, initially set by the user. The idea of modifying a given motion trajectory with a maximum deviation was initially proposed in [22]. In this case, the nominal motion is defined via multiple demonstrations, but then the user has the possibility of progressively refining the trajectory inside a tube, whose radius is determined automatically,

based on the variation of the training data. The PbD approach, that we propose in this article, is not dissimilar. It is based on a one-shot demonstration and a procedure for approximating the resulting trajectory in an optimal manner within a given tolerance. However, in the framework that we propose, this tolerance is not constant over the entire length of the path or computed automatically but can be specified at runtime by the operator himself. This possibility requires an additional input channel besides the standard position recording, according to a multimodal paradigm.

Especially, in the context of semistructured tasks, additional modalities of interaction can be exploited in support of the physical interaction of kinesthetic trajectory teaching. In this way, further guidance information can be transferred to the robot. The additional interaction modality should be naturally and unobtrusively adjustable by the operator to convey human intentions continuously and during the entire robot teaching execution.

In recent years, human–robot interaction has largely benefited from wearable devices for the measurements of biological signals. Physical interaction augmented by such kinds of biointerfaces allows lightweight, effective, and intuitive information exchange with minimal constraint/encumbrance on the human body. In [23], it has been demonstrated that human motor intentions can be detected with good accuracy from surface electromyography (sEMG). The online extraction of information from sEMG signals represents a valuable approach for the multimodal guidance of robotic devices. In particular, the human ability to simultaneously activate antagonistic muscles both in static postures and dynamic motions can be exploited. The central nervous system continuously exploits muscle cocontraction (i.e., coactivation of antagonistic muscles) to regulate joint mechanical impedance [24]. We experience every day that we modulate muscle cocontraction to stiffer the joints of our limbs during the execution of motor tasks that require higher accuracy, while we reduce their mechanical stiffness when low precision is required. This behavior has been experimentally investigated and confirmed by clinical trials in [25], where human joint stiffening induced by muscle cocontraction has been measured by sEMG signals during a pointing task of randomized targets. On the basis of these considerations, the idea developed in this article consists in augmenting the kinesthetic teaching of collaborative robots, with an sEMG-based measurement of the forearm muscle cocontraction, that can be used by the operator to continuously modulate the desired level of accuracy. In order to follow the natural behavior of the human motor control, low cocontraction levels are mapped onto large tolerance radii and *vice versa*. Furthermore, the operator is provided with a vibrotactile feedback about the actual level of cocontraction that simplifies this operation.

B. Contribution and Structure

This article presents a novel PbD algorithm that starting from a single kinesthetic demonstration provides an optimized motion trajectory subject to geometric constraints directly defined by

the user during the demonstration. In this manner, it is possible to cope at the same time with:

- 1) the compliance with the task constraints that are directly taught by the operator by varying the hand stiffness deduced from forearm muscles' sEMG;
- 2) the smoothness of the path, which by virtue of the adopted parameterization based on smoothing B-spline has the minimum curvature/acceleration.

Note that the regularity of the trajectory obtained as a result of a PbD approach is neglected in a large body of literature focusing on this topic. Accordingly, industrial applications where the execution time must be minimized cannot take advantage of these programming methods because of the high curvature that often characterizes the resulting trajectories, which may induce large accelerations and even vibratory phenomena. The trajectories considered in this article are the so-called smoothing B-splines [26], which are used in the industrial field, and in particular in the robotic one when it is necessary to define a trajectory as a tradeoff between the need of minimizing the approximation error with respect a given set of via points and the need of limiting the geometric curvature. Interesting enough, smoothing B-splines are also a standard tool for solving nonparametric regression problems of estimating a curve $g(t)$ given n observations affected by random noise, i.e., $y(t_i) = g(t_i) + \varepsilon_i$ for $i = 1, \dots, n$ [27]. In our application, the curve $g(t)$ is the ideal trajectory that the robot should track, and the noise ε_i is caused by the unwanted motions affecting the human during the demonstration phase. Therefore, the approach based on smoothing B-splines can be seen as an attempt of eliminating this noise. Furthermore, to cope with the bounds imposed by the user, the standard approach has been enhanced with constraints on the maximum deviation with respect to the original trajectory. To do this, a novel use of the human stiffness estimation has been exploited. As a matter of fact, the idea of using an estimation of the user impedance for teaching a robot is not new in general, since many examples can be found in the scientific literature where the impedance of the human operator is reflected to the robot. For instance, in [28], the user can select the stiffness of the commanded robot while teaching its position trajectory by acting on an input interface based on a linear spring-return potentiometer that maps a button position to the arm stiffness. Previously, the concept of teleimpedance was introduced in [29], where in a leader–follower teleoperation scheme, a reference command composed of both the desired motion trajectory and the impedance profile is sent from the human operator to the remotely operated robot. Note that the impedance profiles are deduced from sEMG measurements on the operator's arm. In [30], a framework for robot PbD based on a single demonstration provided by the human operator is proposed, in which the teleimpedance approach, for teaching the robot stiffness, is integrated with a generalization of the trajectories demonstrated by the user in order to let the robot adapt to different object and environment configurations. In [31], sEMG signals and kinesthetic control are exploited in a PbD task, where the positions imposed by the human operator and impedance gains deduced from the muscles activity are recorded and then applied offline to the robot. Similarly, in [32], the trajectory

demonstrated by a human operator together with sEMG signals is used to command both the position and the stiffness of a robot manipulator interacting with the environment. In these cases, the signals are encoded by using DMPs models, so that they can be easily adapted to new tasks.

Differently from the abovementioned examples, in the framework proposed in this article, we are not going to estimate the user stiffness at her/his arm's endpoint (i.e., Cartesian stiffness) with the objective of making the robot replicate the human stiffness profile. In our approach, we use sEMG to estimate the overall stiffness of the user's hand (fingers and wrist joints) from the activation of the predominant extrinsic hand's antagonistic muscles located in the forearm, to provide the user with an additional programming input. This can be used to specify at runtime—i.e., during the execution of kinesthetic teaching by physically guiding the robot—the geometric constraints for a B-spline trajectory optimization problem, which minimizes the noise and curvature of the trajectory taught by the user. Importantly, since the user should be able to voluntarily modulate the additional programming input during the operation, vibrotactile feedback is provided to inform him about the actual modulated level of hand stiffness. This approach allows us to better capture the human intentions and enhance the effectiveness of the PbD method in programming complex tasks, by maintaining the same level of intuitiveness. Experiments involving ten subjects in teaching trajectories of a seven-degree-of-freedom collaborative robot, while modulating sEMG-estimated hand stiffness, show that the proposed system can be successfully used for intuitive multimodal robot teaching of tasks in semistructured environments. Without pretraining or previous experience with sEMG/vibrotactile interfacing, subjects were required to teach trajectories with instructed target hand stiffness levels in a test scenario characterized by obstacle courses with narrow passages, variations of end-effector orientation, and specific workspace locations to be reached, with the underlying idea of reproducing some mock-up welding process teaching trajectory.

The rest of this article is organized as follows. In Section II, the general scheme for robot programming based on kinesthetic teaching and sEMG signals is illustrated; then, the constrained optimization problem for the B-spline robot trajectory definition is formulated in Section III. In Section IV, the robotic setup used in the experiments is presented; in Section V, the effectiveness of the proposed PbD approach is demonstrated both in terms of characteristics of the programmed motions and intuitiveness for operators. Finally, Section VII concludes this article.

II. TRAJECTORY LEARNING TECHNIQUE

The proposed programming framework aims at defining an optimal trajectory that approximates the geometric path demonstrated by the user, directly acting on the robot, within a tolerance defined at each time instant by modulating the cocontraction level of forearm muscles. A block scheme representation of this approach is illustrated in Fig. 1. The user steers the robot along the desired trajectory by applying a wrench $\mathbf{F}_h(t)$ at its end-effector. It is possible to make the end-effector able to follow and adapt to the force exerted by the operator in many ways,

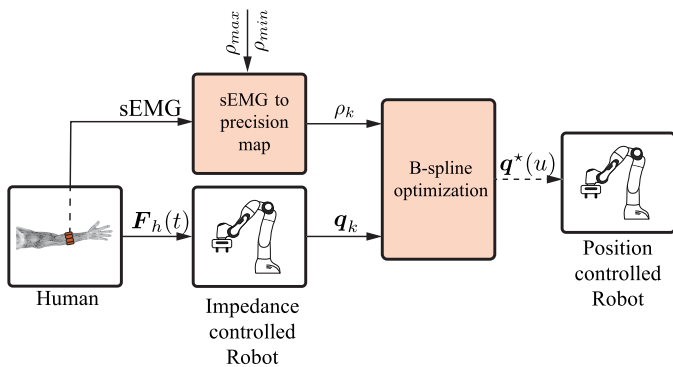


Fig. 1. Block scheme representation of the novel framework for trajectory learning via kinesthetic teaching and sEMG-based accuracy modulation.

for instance, by imposing that the dynamical behavior of the robot matches the dynamics of the mass–damper system [33]. A detailed description of the controller implemented in the experimental robotic setup is reported in Section IV-C.

While the user performs the position programming task, and the position \mathbf{q}_k of the robot is recorded with a given sampling period T_s , the sEMG signals that activate the muscles of the forearm are also acquired and elaborated to obtain the profile of the desired precision ρ_k at each sampling instant. This basic element of the proposed strategy is described in Section II-A. Finally, the pairs (\mathbf{q}_k, ρ_k) , $k = 0, \dots, n$, are the input of an optimization process, illustrated in Section III, that provides the trajectory $\mathbf{q}^*(u)$ to be applied to the robot. It is worth noticing that:

- 1) the point \mathbf{q}_k , and consequently the trajectory $\mathbf{q}^*(u)$, can be considered both in the robot joint space and in the robot workspace; the difference consists in the meaning of ρ_k , which in the former case is a maximum deviation with respect to joint variable values, while in the latter is a maximum distance from the geometric path demonstrated by the user;
- 2) when defined in the workspace, function $\mathbf{q}^*(\cdot)$ only specifies the geometric path of the motion; in general, the variable u can be defined independently to specify time law along the curve.

A. Desired Precision Estimation by sEMG Analysis

As already mentioned, one of the key elements of the proposed scheme is the mapping between the sEMG signals of the human operator and the desired level of accuracy. This mapping is based on a two-stage process: first, the stiffness of the operator's hand is estimated based on a simple consideration of the physiological phenomena governing the regulation of the human stiffness, and then, this value is converted in a distance via linear scaling.

Various implementations of human stiffness estimations from sEMG have been presented in the literature. For instance, in [34] and [35], sEMG-based human stiffness estimation methods are developed and used for supervisory control and bilateral teleoperation of robotic systems. Other approaches for sEMG-based stiffness estimation involve detailed Hill-based muscle

modeling techniques [36] (see, e.g., [37]), where an assistive control scheme control for a knee exoskeleton device has been developed. These methods mainly differ on the desired level of accuracy required by the specific application. As can be easily understood, as much as a higher accuracy is requested, the higher will be the complexity of the calibration on different users, the difficulty of parameter identification, and the time necessary to obtain them. In the application proposed in this article, a very accurate stiffness estimation is not necessary, because the objective is to obtain a rough measurement of the overall hand stiffness *continuously modulated* by the operator, and therefore, a simplified model of the stiffness regulation can be adopted. Hand stiffness is the overall stiffness seen by the object grasped by the human hand (in our case, the robot end-effector) and is the result of the cumulative stiffness of each finger, which, in turn, depends on the stiffness of each joint. It is well known that the joints of the fingers are actuated by the muscles located in the forearm through a net of tendons. Moreover, studies on the human muscle geometry and joint mechanics [38], [39] have demonstrated that the total human hand stiffness is a function of hand muscle activations. Since the mechanical impedance of the human hand can be seen as a consequence of the overlapping of several muscle activations working in an antagonistic configuration [40], it is possible to consider the hand stiffness as a summation of the activations of all the muscles involved in the grasping task [39]. According to these considerations, the hand stiffness estimation can be deduced from the sEMG signals acquired from the operator's forearm. This approach has been demonstrated to be suitable in several telerobotics works [41], [42], [43], [44]. In detail, we exploit the fact that the n sEMG signals taken from the forearm, as described in Section IV-A (in our work, $n = 8$), are located on the $n/2$ couples of predominant hand antagonistic extrinsic muscles [45], according to the well-known concept of the human antagonistic actuation model [39], [41]. We can, therefore, consider that such sEMG activity captures the information related to the activation of the predominant hand muscular antagonistic actions. Formally, considering n sEMG channels e_1, e_2, \dots, e_n measured by a bracelet placed around the forearm, as described in Section IV-A, the estimation of the stiffness σ_k of the operator's hand is computed as

$$\sigma_k = \sigma(kT_s) = \sum_{i=1}^n e_i(kT_s) / e_{\max} \quad (1)$$

where

$$e_{\max} = \max \left\{ \sum_{i=1}^n \text{MVC}(e_i) \right\} \quad (2)$$

is the maximum value of the sum of the maximum voluntary contractions (MVCs) [46] recorded from each sEMG channel during a calibration operation, in which the human operator is asked to stiffen at maximum level her/his hand for three times while grasping the robot handle end-effector (see Fig. 2), T_s is the sampling time, and $k = 0, 1, 2, \dots$. According to its definition, the hand stiffness estimation σ_k is an index normalized in the range $[0, 1]$. From σ_k , the desired precision ρ_k can be readily

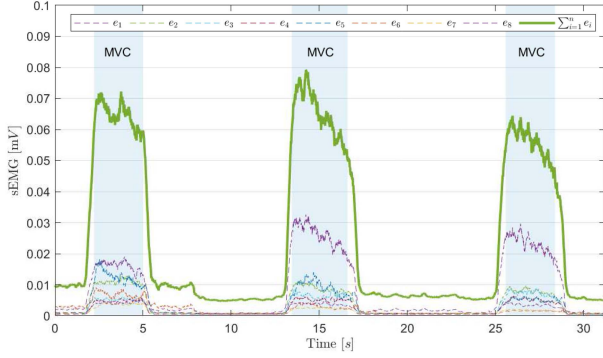


Fig. 2. Example of an sEMG calibration procedure [see (2)]. The blue-colored zones highlight the execution of the MVC considered in (2).

obtained via a linear scaling, i.e.,

$$\rho_k = \rho_{\min} + (\rho_{\max} - \rho_{\min}) \sigma_k \quad (3)$$

where ρ_{\min} and ρ_{\max} are arbitrary parameters denoting, respectively, the minimum and the maximum level of accuracy in order to match the requirements of a specific robotic task of interest.

III. TRAJECTORY PLANNING

A. Trajectory Definition

The learned trajectory is defined by using the so-called B-spline curves. They represent a different mathematical formulation of standard spline parameterization based on piecewise polynomial functions. A B-spline of degree p is a parametric curve $\mathbf{s} : [u_{\min}, u_{\max}] \rightarrow \mathbb{R}^d$ defined as convex combinations of *control points* $\mathbf{p}_j \in \mathbb{R}^d$ weighted by *B-spline basis functions* of degree p , $B_j^p(u)$:

$$\mathbf{s}(u) = \sum_{j=0}^N \mathbf{p}_j B_j^p(u), \quad u_{\min} \leq u \leq u_{\max}. \quad (4)$$

More details about the definition and computation of basis functions $B_j^p(u)$ are provided in Appendix A.

The vectorial coefficients \mathbf{p}_j , $j = 0, \dots, N$, determine the shape of the curve and are generally computed by imposing approximation/interpolation conditions on a given set of data points. In the proposed application based on the direct control of the robot manipulator by the human operator via physical interaction, these points are the samples \mathbf{q}_j , $j = 0, \dots, n$, of the recorded trajectory. Note that \mathbf{q}_j , and accordingly \mathbf{p}_j , are vectors whose dimension d depends on the specific task. For instance, in an application where only the position of the robot end-effector must be programmed and the orientation is kept constant, \mathbf{q}_j and \mathbf{p}_j will be 3-D vectors, while if also the orientation is considered, the two vectors will include a minimal representation of the orientation, like, e.g., the Euler angles, and they will be, therefore, defined in a 6-D space.

Since the via points \mathbf{q}_j are derived from the trajectory of the robot $\mathbf{q}(t)$ obtained via kinesthetic teaching, an exact interpolation is certainly unsuitable because of the unwanted movements that affect the user. In the consideration of the modest

precision required by the task, which is comparable to that of a human operator, the idea is to design a trajectory path that lies in the neighborhood of the given via points with the lowest possible curvature and, therefore, the lowest acceleration. To this purpose, smoothing B-splines are the ideal tool [26], being the functions $\mathbf{s}(u)$ that minimize the quantity

$$J := \sum_{j=0}^n w_j \|\mathbf{s}(u_j^*) - \mathbf{q}_j\|^2 + \lambda \int_{u_{\min}}^{u_{\max}} \left\| \frac{d^2 \mathbf{s}(\tau)}{d\tau^2} \right\|^2 d\tau \quad (5)$$

where $\lambda \geq 0$ is a parameter that can be freely chosen to control their smoothness, while $w_j > 0$ is a parameter used to selectively weight the contribution of the squared error at a particular point \mathbf{q}_j . Parameter u_j^* denotes the time instant at which the approximation occurs.

If $\lambda = 0$, the problem becomes a standard least-squares error fitting of the given trajectory points with a B-spline. In case $n = N$, the spline exactly interpolates all the points \mathbf{q}_j , while if $n > N$, the obtained B-spline only approximates these points in a quadratic sense. However, in the latter case, the reduction of the number of control points has a significant impact on the computational load required by the optimization problem and is, therefore, desirable. For this reason, we assume that $n \gg N$, considering, for instance, a fixed ratio between the number of samples \mathbf{q}_j of the input trajectory and the number of control points \mathbf{p}_j .

Interestingly enough, by rewriting in a matrix form the standard expression of the B-spline (4) as

$$\mathbf{s}(u) = (\mathbf{B}(u) \mathbf{P})^T = \mathbf{P}^T \mathbf{B}^T(u) \quad (6)$$

where

$$\mathbf{B}(u) = [B_0^p(u), \dots, B_N^p(u)] \quad (7)$$

$$\mathbf{P} = [\mathbf{p}_0, \dots, \mathbf{p}_N]^T \quad (8)$$

the cost function (5) can be written as

$$J = J_1 + \lambda J_2 \text{ with}$$

$$J_1 = \text{tr} \left((\hat{\mathbf{Q}} - \mathbf{B}\mathbf{P})^T \mathbf{W} (\hat{\mathbf{Q}} - \mathbf{B}\mathbf{P}) \right)$$

$$J_2 = \text{tr} \left((\mathbf{C}\mathbf{P} + \hat{\mathbf{P}})^T \mathbf{A} (\mathbf{C}\mathbf{P} + \hat{\mathbf{P}}) \right) \quad (9)$$

where $\text{tr}(\cdot)$ denotes the trace of a matrix, $\mathbf{W} = \text{diag}(w_j) > 0$, $j = 0, \dots, n$, is the weight matrix, and $\hat{\mathbf{Q}}$, $\hat{\mathbf{P}}$, \mathbf{B} , \mathbf{C} , and \mathbf{A} are matrices of proper dimensions, which depend on known quantities such as \mathbf{q}_j , u_j^* , and u_j . The solution of the optimal problem

$$\min_{\mathbf{P}} J(\mathbf{P}) \quad (10)$$

is unique and can be computed analytically.

For the mathematical details, including the expression of the matrices that appear in (9) and the expression of the optimal solution, see Appendix B.

B. Constraint Definition

One of the main drawbacks of the optimization problem (10) is the critical role played by the parameter λ , which must be

selected by the user. More in general, as highlighted in [47], in many problems related to robot navigation/trajectory planning based on splines, it is difficult to balance the tradeoff between the smoothness of the curve and the desired shape. For this reason, smoothing splines with prescribed tolerance have been already proposed in the literature [26]. According to this approach, the parameter λ is selected with the purpose of bounding the maximum approximation error below a given threshold ρ , i.e.,

$$\max_j \|\mathbf{s}(u_j^*) - \mathbf{q}_j\| \leq \rho.$$

However, this type of bound may lead to overconservative solutions, since the problem is global, i.e., involving the entire curve. With the proposed approach, the maximum distance between the points of the input curve and the corresponding point of the B-spline trajectory is limited in a local sense, i.e., for each point, and with a value of the threshold ρ , which may be different for each of them, i.e., ρ_j , $j = 1, \dots, n-1$ (being the first and last point exactly interpolated). If the goal is quite clear, the ways of implementing it can be different in terms of complexity and implications. A possible solution is based on the pointwise selection of weight coefficients w_j , as in [48]. However, this approach is generally based on heuristics and does not assure the exact compliance with the bounds. For this reason, the setup of a constrained optimization problem is preferable.

Three alternative types of constraints on the control points \mathbf{p}_j have been identified, each one with pros and cons.

1) *Polyhedron*: For each point \mathbf{q}_j , it is assumed that

$$|(\mathbf{s}(u_j^*) - \mathbf{q}_j)_k| \leq \rho_{j,k}, \quad k = 1, \dots, d \quad (11)$$

where subscript k denotes the k th component of the vector $(\mathbf{s}(u_j^*) - \mathbf{q}_j)$. Note that for each component, it is possible to select a different bound $\rho_{j,k}$. By considering (6), condition (11) be rewritten as

$$\mathbf{q}_j - \boldsymbol{\rho}_j \leq \mathbf{P}^T \mathbf{B}^T(u_j^*) \leq \mathbf{q}_j + \boldsymbol{\rho}_j \quad (12)$$

where $\boldsymbol{\rho}_j = [\rho_{j,1}, \dots, \rho_{j,d}]^T$. With these linear constraints, the optimization problem (10) becomes a convex quadratic programming problem, for which efficient solutions exist [49].

2) *Euclidean ball*: For each point \mathbf{q}_j , it must be assured that

$$\|\mathbf{s}(u_j^*) - \mathbf{q}_j\| \leq \rho_j \Leftrightarrow \|\mathbf{P}^T \mathbf{B}^T(u_j^*) - \mathbf{q}_j\| \leq \rho_j \quad (13)$$

where ρ_j is a scalar. This is clearly a nonlinear constraint, but the convexity, and therefore the possibility of finding a solution of the optimization problem, is ensured in any case. In addition, only a parameter is sufficient to bound the distance between each point and the approximating spline, even if the meaning in the case of trajectories, that include also the orientation, is questionable. In this case, multiple constraints, one for the position and one for the orientation, are preferable.

3) *Cylinder*: The error between the B-spline trajectory and the points \mathbf{q}_j is decomposed by considering the tangent direction to the motion and the orthogonal plane to this vector (a similar approach can be found, e.g., in [50]).

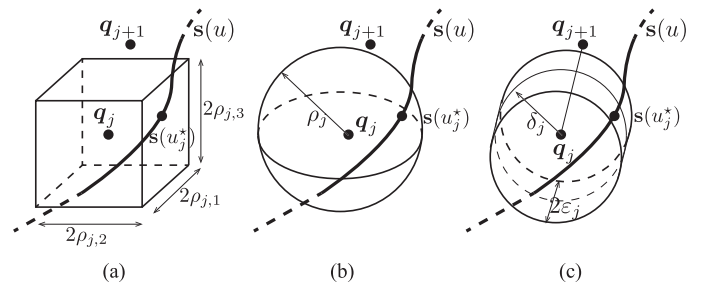


Fig. 3. Different types of convex sets, defined in the 3-D space to constraints the approximation error of a B-spline with respect to a given point \mathbf{q}_j . (a) Polyhedron. (b) Ball. (c) Cylinder.

Obviously, this type of constraint can be applied only to position trajectories in the 3-D space. Let us assume that

$$\mathbf{t}_j = \frac{\mathbf{q}_{j+1} - \mathbf{q}_j}{\|\mathbf{q}_{j+1} - \mathbf{q}_j\|}$$

and that \mathbf{n}_j is the vector of maximum norm (properly normalized) among $\mathbf{t}_j \times \mathbf{i}$, $\mathbf{t}_j \times \mathbf{j}$, and $\mathbf{t}_j \times \mathbf{k}$ being \mathbf{i} , \mathbf{j} , and \mathbf{k} the unit vectors defining the base frame of the application, and where \times denotes the cross product between vectors. Finally, assume that $\mathbf{b}_j = \mathbf{t}_j \times \mathbf{n}_j$. With these three vectors, it is possible to define the rotation matrix $\mathbf{R}_j = [\mathbf{t}_j \ \mathbf{n}_j \ \mathbf{b}_j]^T$ that brings the approximation error in a frame, whose x -axis is aligned with the tangent direction while y - and z -axis are within the plane orthogonal to this direction, i.e.,

$$\mathbf{e}_j = \mathbf{R}_j(\mathbf{P}^T \mathbf{B}^T(u_j^*) - \mathbf{q}_j)$$

and then it is possible to impose that

$$-\varepsilon_j \leq e_{j,1} \leq \varepsilon_j \quad (14)$$

and

$$\sqrt{e_{j,2}^2 + e_{j,3}^2} \leq \delta_j \quad (15)$$

with $\mathbf{e}_j = [e_{j,1} \ e_{j,2} \ e_{j,3}]^T$. The definition of this feasible set is much more complex than balls and polyhedra, but the convexity is still preserved. Moreover, even if limited to the 3-D position trajectory, its meaning is very clear.

In Fig. 3, a pictorial representation of the three different sets for a given point \mathbf{q}_j is reported. Note that the constraints only affect the geometry of the curve, while the way in which the B-spline is tracked and the related quantities, such as velocity, acceleration, etc., are not taken into account. The choice of a particular type of constraint has a strong influence on the computation time required to solve the constrained nonlinear optimization problem. In particular, it is possible to prove that ball- and polyhedron-type constraints lead to similar computation times, while the more complex constraint based on cylinders leads to a duration that is approximately double. Moreover, the different types of constraints affect differently the approximation error and the smoothness of the final trajectory, but the variation is rather marginal, and from this point of view, the three methods are almost equivalent.

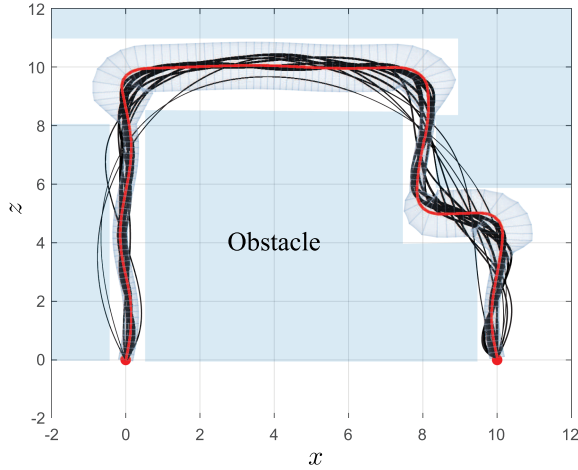


Fig. 4. Pick-and-place task with collision avoidance: approximated B-splines obtained for different values of N (black lines) and input trajectory (red line).

C. Constrained Trajectory Optimization

Because of the convexity of the objective function and of the admissible sets, the solution of (10) subject to (12), (13), or (14) and (15) can be efficiently found with state-of-the-art numerically optimization methods. However, since usually $n \gg N$, and accordingly the number of constraints $(n - 1)$ is much larger than the number of free parameters p_j , it may happen that the intersection among all the admissible sets, which is still a convex set, does not contain any solution. For this reason, a two-stage optimization problem has been devised.

In the first step, it is checked whether a number $N + 1$ of control points guarantees that the intersection of all the convex sets deriving by the constraints is not empty. This can be done by computing the (unconstrained) approximating B-spline, by means of (23) with $\lambda = 0$, and then by verifying if all the constraints are met. In this case, the constrained optimization problem can be finally solved (in the second step of the procedure). If the approximating B-spline is not compliant with all the constraints, then it is necessary to increase N . To illustrate this procedure, we consider a simple pick-and-place task that also requires obstacle avoidance, which is shown in Fig. 4. It is assumed that obstacles remain stationary, and an initial trajectory (red line) is given together with some bounds that define a feasible space where the optimized trajectory can lie. The approximating B-spline trajectories for increasing values of N (varying from 3 to 14) are shown in the figure with back lines, whose thickness is proportional to N (thin lines correspond to B-spline with few control points and *vice versa*). It is possible to notice that as N grows, B-spline curves become closer and closer to the original trajectory (which is composed of 145 points). This is confirmed by trend of the cumulative approximating error

$$J_1 = \sum_{j=0}^n \|s(u_j^*) - \mathbf{q}_j\|^2 \quad (16)$$

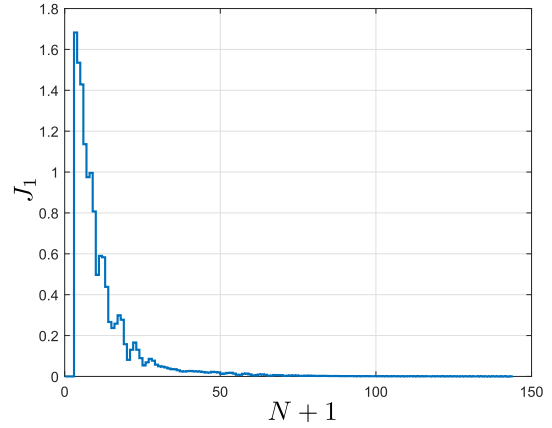


Fig. 5. Cumulative approximating error $J_1 = \sum_{j=0}^n \|s(u_j^*) - \mathbf{q}_j\|^2$ produced by B-spline curves with respect to the original trajectory reported in Fig. 4, as a function of the number of control points $N + 1$, ranging from 3 to 145, i.e., the number of samples of the initial trajectory.

as a function of the number $N + 1$ of control points defining the B-spline, which is shown in Fig. 5. It can be noticed that this error decreases as N grows, even if the trend is not strictly monotonic, and many local minima are present. As a matter of fact, the error tends to zero as the number of control points $N + 1$ approaches the number of given via points $n + 1$, since in this case an exact interpolation is obtained. Note that not only the global error but also the local error at a specific via point tends to zero, as N goes to n . On the other hand, as remarked in [51], an error function such as (16) is not convex with respect to the knot location. By considering a fixed knot spacing, the variation of the number of knots also modifies their position and, therefore, causes the local minima observed in Fig. 5.

In any case, by progressively increasing N , the bounds on the error are certainly met. Therefore, when the initial value of N does not assure any feasible solution, or if parameter N is not an input of the problem but can be freely chosen, it is possible to find the value that minimizes the dimension of the trajectory optimization problem by starting from the given value, or from a small value (e.g., 3), of N and then iteratively increasing this value until the approximating trajectory is compliant with all the constraints. Since the analytic solution for determining the approximating B-spline is available (see (23) in Appendix B), this procedure can be performed very efficiently and in a very small amount of time.

Once that the value of N has been determined, the constrained optimization problem based on the cost function (10) with $\lambda \neq 0$ can be finally solved by using off-the-shelf convex optimization solvers and exploiting the solution found in the first step as an initial guess. Note that the value of N found is only a sufficient condition for the existence of the solution, but, providing a feasible solution used as the initial condition for the constrained optimization, it allows us to speed up the numerical computation process. Algorithm 1 summarizes the overall procedure.

By considering the pick-and-place example previously mentioned, the minimum value of N that guarantees the compliance of the approximating B-spline trajectory with all the constraints

Algorithm 1. Algorithms for the Determination of the Constrained Smoothing B-Spline.

input : parameter λ^* , weights w_j , via-points \mathbf{q}_j , thresholds ρ_j , $j = 0, \dots, n$, number of control points $N^* + 1$, constraint type: (12), (13), or (14)-(15);

output: vector \mathbf{P} of the optimal constrained trajectory;

$N \leftarrow N^*$;

$\lambda \leftarrow 0$;

Solve the unconstrained optimization problem (20) with (23);

while constraints are not satisfied **do**

$N \leftarrow N + 1$;

 Solve the unconstrained optimization problem (20) with (23);

end

Numerically solve the constrained optimization problem (20)

+ (12) or (13) or (14)-(15)

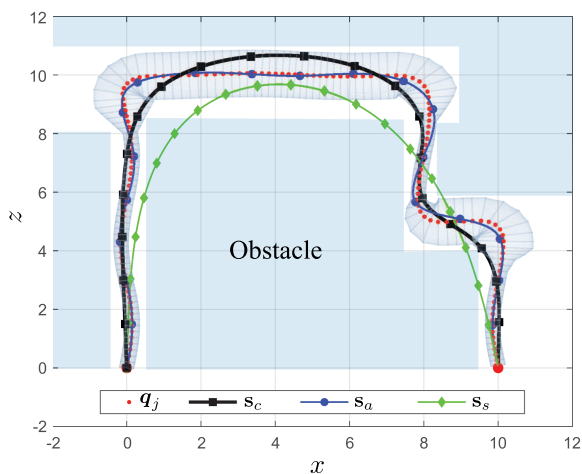


Fig. 6. Pick-and-place task with collision avoidance: geometric path of input trajectory \mathbf{q}_j , constrained smoothing B-spline \mathbf{s}_c , approximated B-spline \mathbf{s}_a , and (unconstrained) smoothing B-spline \mathbf{s}_s .

is 14. The resulting trajectory, obtained with $\lambda = 0.01$, is shown in Fig. 6. In the same figure, the constrained B-spline is compared with the approximating B-spline that satisfies all the constraints (initial solution) and with the unconstrained smoothing B-spline defined with the same value of λ . It is clear that the unconstrained trajectory is the smoothest one, but it collides with the obstacles being not included in the feasible region. On the other hand, the approximating B-spline is very close to the input trajectory but reproduces also unwanted phenomena, such as involuntary movements that may be superimposed to the ideal trajectory. The constrained smoothing B-spline represents the optimal tradeoff as confirmed also by Fig. 7, where the first and second derivatives with respect to u (which are proportional to velocity and acceleration, respectively, once the motion law $u = u(t)$ has been defined) are shown. In particular, the constrained B-spline curve exhibits intermediate values between the unconstrained smoothing B-spline and the approximating B-spline, with a substantial reduction with respect to the input trajectory. Finally, it is worth noticing that the devised optimization procedure only affects the geometry of the trajectory, while the way in which the trajectory is tracked, i.e., the motion law, is not taken into

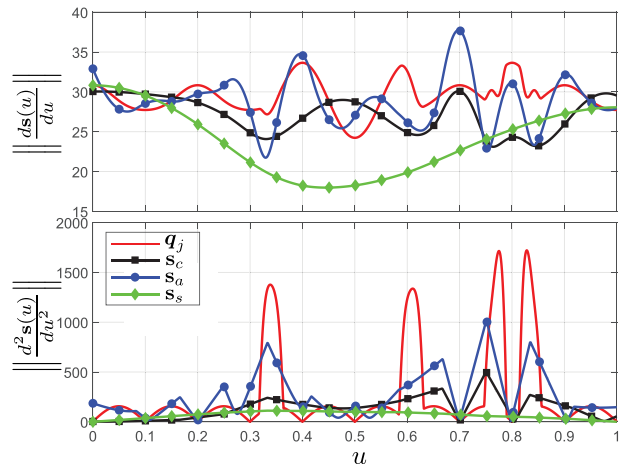


Fig. 7. Pick-and-place task with collision avoidance: norm of the first and second derivatives of input trajectory \mathbf{q}_j , constrained smoothing B-spline \mathbf{s}_c , approximated B-spline \mathbf{s}_a , and (unconstrained) smoothing B-spline \mathbf{s}_s .

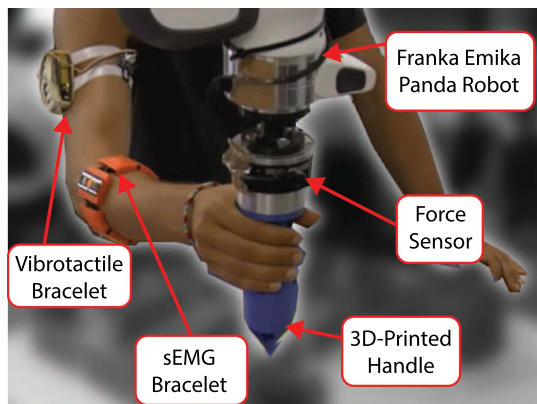


Fig. 8. sEMG-enhanced robot programming setup of the present work.

account, because many different approaches can be adopted according to the needs. For instance, it is possible to reproduce the input trajectory with the same timing or follow the geometric path at a constant velocity or in minimum time.

IV. EXPERIMENTAL SETUP

To validate the proposed approach, the experimental setup shown in Fig. 8 has been used. It is composed of a robot manipulator equipped with a force/torque sensor and an sEMG sensing device with vibrotactile feedback. In the following, a detailed description of the hardware is given.

A. sEMG Sensing

For the sensing of the sEMG activity, eight sEMG channels are acquired from the forearm of the operator. Specifically, a gForcePro wearable armband (by OYMotion) has been used for signal acquisition. Since the predominant muscles producing flexion and extension actions in the hand are the *Flexor Digitorum Superficialis* and *Extensor Digitorum Communis*, the armband has been placed on the forearm in the proximity of such

muscles—that is, close to the proximal part of the forearm [45]. The sEMG provided by the armband is a raw signal at the sampling frequency of 1 kHz, sent to a nearby PC through a Bluetooth interface. After the acquisition, each raw sEMG channel is processed by a chain [41] composed of a 50-Hz notch filter for powerline interference cancellation, a 20-Hz high-pass filter for powerline noise reduction, and the root-mean-square filter computed over 200-ms running window.

B. Vibrotactile Feedback

The operator has been equipped with a wearable bracelet, based on a Grove vibration motor by Seeed Technology Co., that produces an adjustable vibrational skin stimulation. The intensity of such stimulation can be modulated by controlling the motor vibration magnitude, using the electronic control board Seeeduino V4.2, in a range of integer values from 100 to 255. In order to provide the operator with a vibrational feedback indicating the hand stiffness level in real time, σ_k has been mapped in the input range of the bracelet. Note that this technological element is not strictly necessary for the proposed application, but is an aid that allows the user to better perceive the applied level of stiffness.

C. Robotic Manipulator

The experiments have been carried out by using the collaborative robot Panda (by Franka Emika GmbH, Munich, Germany) equipped with a 3-D printed handle and a six-axis force sensor (ATI Industrial Automation, Apex, NC, USA), placed between the handle and the robot flange. A kinesthetic teaching interface between the operator and the Panda robot has been implemented by imposing a partial compensation of the robot dynamics plus a standard proportional–derivative (PD) control along the motion directions that are constrained during the PbD tasks [33]. By using this mechanism, in the first experiment, the orientation has been kept constant. Let us consider the dynamic model of the robot manipulator, expressed in the operational space [52]

$$\Lambda(\theta)\ddot{x} + \Gamma(\theta, \dot{\theta})\dot{x} + \eta(\theta) = F_c + F_h$$

where $\theta \in \mathbb{R}^n$ is the vector of the joint coordinates, $x = (p^T, \varphi^T)^T$ is the vector describing the end-effector position and orientation (φ is a minimal representation of the orientation, like, e.g., Euler angles), $\Lambda(\theta) = (JM^{-1}(q)J^T)^{-1} \in \mathbb{R}^{6 \times 6}$ is the workspace inertia matrix, $\Gamma(\theta, \dot{\theta}) = J^{-T}C(\theta, \dot{\theta})J^{-1} - \Lambda(\theta)\dot{J}J^{-1}$, and $\eta(\theta) = J^{-T}g(\theta)$ are the wrenches of centrifugal and gravitational effects, respectively; $M(\theta)$, $C(\theta, \dot{\theta})$, and $g(\theta)$ are the corresponding quantities defined in the joint space. $F_c = J^{-T}\tau_c$ is the end-effector wrench corresponding to the control (joints) torques τ_c and F_h is the wrench applied by the user to end-effector. The control action

$$F_c = \hat{\Gamma}(\theta, \dot{\theta})\dot{x} + \hat{\eta}(\theta) + F_{PD} + \alpha\hat{F}_h \quad (17)$$

is characterized by three terms:

- 1) a compensation of the Coriolis and gravity terms based on a proper estimation of the functions $\Gamma(\theta, \dot{\theta})$ and $\eta(\theta)$;

- 2) a PD controller

$$F_{PD} = K_p(x_{\text{ref}} - x) + K_d(\dot{x}_{\text{ref}} - \dot{x})$$

where K_p and K_d are diagonal matrices with nonnegative elements; in particular, the gains of K_p equal to zero correspond to the directions that are directly commanded by the user (in this case, the corresponding components of x_{ref} and \dot{x}_{ref} are null);

- 3) a feedforward term proportional to the wrench \hat{F}_h exerted by the user and measured by the six-axis force sensor; a low-pass filter is applied for rejecting high-frequency vibrations elasticity induced by robot's kinematic and actuation chains.

By assuming $\hat{F}_h \approx F_h$, the dynamics of the robot equipped with this control along the workspace directions commanded by the user becomes

$$\frac{\Lambda(\theta)}{1 + \alpha}\ddot{x} + \frac{K_d}{1 + \alpha}\dot{x} + \frac{\rho(\theta, \dot{\theta})}{1 + \alpha} = F_h$$

where $\rho(\theta, \dot{\theta})$ takes into account the mismatch between the actual Coriolis and gravity terms and the corresponding estimations and also the unmodeled dynamics like, e.g., frictional forces. Note that, even if the used controller does not modify the workspace inertia of the robot, the control action based on the external force measurement has the effect of scaling the inertia seen by the user acting on the end-effector, especially for a large value of gain α . Moreover, this term contributes to mitigating the consequences of the undesired dynamics represented by $\rho(\theta, \dot{\theta})$, making the programming task by kinesthetic teaching easier.

V. EXPERIMENTAL RESULTS

In order to demonstrate the effectiveness of the proposed approach, in terms of both the intuitiveness of the programming method for the operator and the features of the final trajectory for the robotic application, various experiments have been carried out under different conditions. In particular, since the human operator plays a key role in the programming task, ten participants have been involved in the experiments to evaluate the proposed approach enhanced with sEMG measurements (age: 30 ± 5). In the following, the subjects are denoted as S1, S2, . . . , S10. The experiment has been conducted in accordance with the Declaration of Helsinki. An informed consent form has been signed by all the subjects, in which a detailed explanation of the experimental protocol has been provided.

A. Experimental Session 1: Obstacle Course With Welding Points

In this first experimental session, the subjects have been asked to lead the robot along a specific path characterized by narrow passages. In particular, as shown in Fig. 9, the robot must initially track a winding path, and then, the end-effector must be placed in specific points avoiding all the obstacles located in the workspace. As shown in Fig. 9(b), the path has been marked with different colors, according to a code that denotes the ideal level of stiffness/precision that the subjects should apply while moving the robot end-effector: maximum level (namely “highest

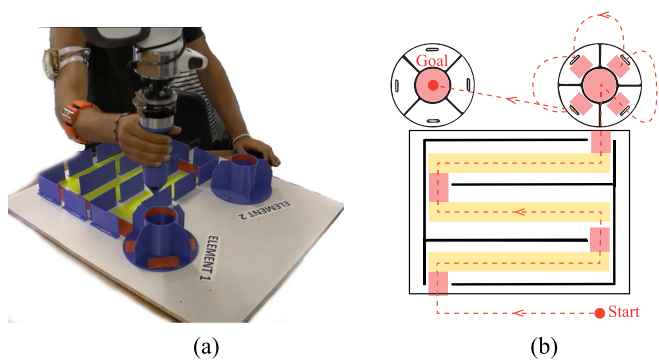


Fig. 9. (a) Setup used in the first experiment and (b) task that the user is requested to demonstrate.

accuracy level”) in the red-coded zones, medium level (namely “intermediate accuracy level”) in the yellow-coded zones, and minimum level in the remaining zones (namely “minimum accuracy level”). First, the calibration has been carried out, and thereafter, the subjects have briefly familiarized themselves with the system and started the experimental task. Note that in this experiment, the orientation of the end-effector has been kept constant and the users only act on the robot position.

In Fig. 10, the procedure for the robot motion definition is exemplified, by considering the PbD task performed by one of the subjects (S1) considered in the experiment.

- 1) The user teaches the robot by moving its end-effector along the prescribed path and modulates the stiffness of its hand at the same time, in order to specify the required accuracy. The color of the points along the trajectory denotes the stiffness level, ranging from 0 to 1. A simple inspection of Fig. 10(a) shows that the stiffness is augmented where a small variance with respect to the imposed trajectory is necessary, while lower values are applied where the final trajectory can differ much more from the input motion.
- 2) The stiffness information is translated into an admissible region about the trajectory applied by the human operator. In particular, the bound on the variation is deduced by exploiting (3), where the minimum and maximum levels are selected on the basis of the specific application (in the experiment, $\rho_{\min} = 0.005$ m and $\rho_{\max} = 0.025$ m).
- 3) The smoothing B-spline is finally computed by assuming $\lambda = 10^{-5}$ and cylindrical constraints.

By comparing the initial motion in Fig. 10(a) and the final trajectory in Fig. 10(b), the positive effect of the proposed approach is evident: the segments that are approximately straight are further straightened and sharp corners are made smoother; in addition, unwanted fast motions of the user are canceled. These observations are confirmed by the motion profiles along the Cartesian axes reported in Fig. 11.

In this figure, the position velocity and acceleration of the initial trajectory and of the optimal B-spline are compared. Both the motion profiles are a function of the variable $u \in [0, 1]$, obtained by normalizing the time of the demonstration by its duration. The use of the same parameterization also for the B-spline curve allows reproducing the original motion, including, for instance,

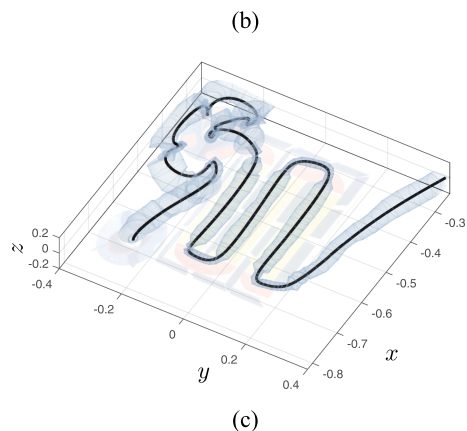
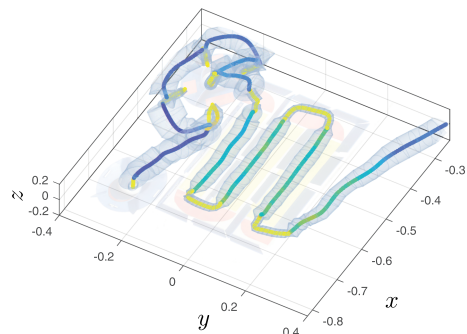
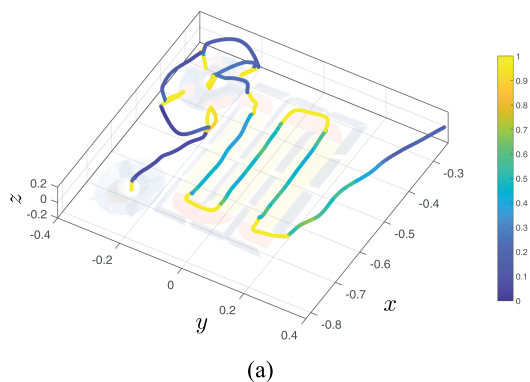


Fig. 10. Sequence of the operations for the definition of the trajectory in the first experimental test. (a) Cartesian path and stiffness level imposed by the human operator. (b) Feasible region about the input trajectory. (c) Constrained smoothing B-spline.

possible stops of the end-effector at some locations. The derivatives of the demonstrated trajectory are obtained via numerical differentiation (this explains the noise superimposed to the first and above all to the second derivative), while the derivatives of the final trajectory are available in a closed form. For a specific insight into the optimized trajectory’s smoothness, velocity, and acceleration characteristics, considering both single-subject and aggregated data, see Section VI-B. In order to evaluate also the intuitiveness of the proposed approach, the capability of the users of modulating the hand stiffness while operating the robot has been tested. As illustrated in Fig. 12, showing the results of the stiffness modulation for a single subject, the user is able to adequately modulate the stiffness in accordance with the requested levels during the kinesthetic teaching, that is to match

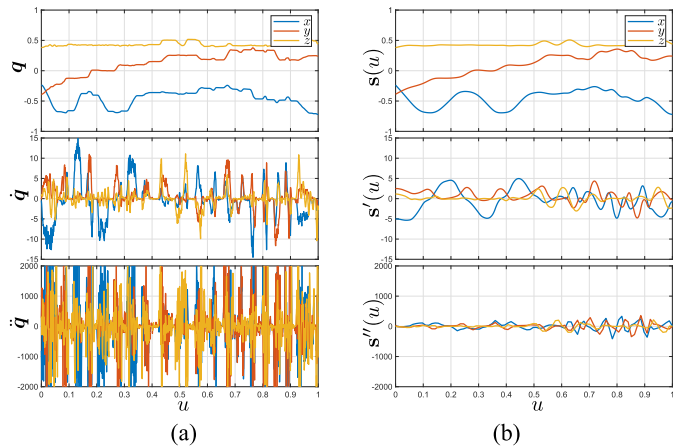


Fig. 11. Motion profiles (position, velocity, and acceleration) along the Cartesian axes obtained in the first experimental test. (a) Trajectory performed by the robot under kinesthetic teaching. (b) Optimal trajectory found with the proposed approach.

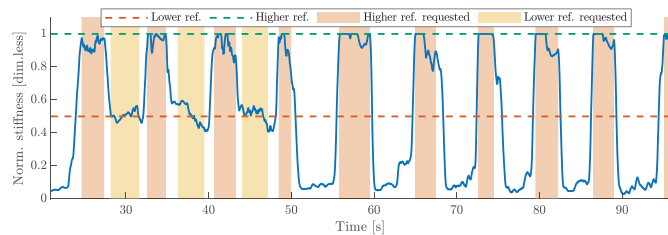


Fig. 12. Stiffness modulation for the subject S1 during the obstacle course with welding points session.

with good approximation the maximum, medium, and lower levels requested in the different zones of experimental test bed. Similar performances are achieved by all the subjects involved in the experiment, and further analyses of the aggregated data on the hand stiffness modulation capabilities are reported in Section VI-A.

Finally, the experiment ends with the position-controlled robot executing the trajectories taught by the subjects and optimized by the constrained smoothing B-spline. The effect of the optimization process is very clear, if the tracking errors $\tilde{\theta}_i$ in the joint space and the corresponding control torques τ_i , $i = 1, \dots, 7$, are considered. From the profiles reported in Fig. 13, a generalized reduction of both the tracking error and the related control action for all the joints, due to the improved smoothness of the optimized trajectory, is evident. A comparison between the peak absolute values (PAVs) of the trajectory demonstrated by the subject S1 and the related optimal path, reported in Table I, shows that the tracking errors are reduced by $[-57.34\%, -86.30\%]$, while the commanded torques by $[-7.69\%, -72.11\%]$.

B. Experimental Session 2: Tilted Welding Points

In this second experiment, based again on the mock-up of a welding process, the users are asked to place the end-effector in

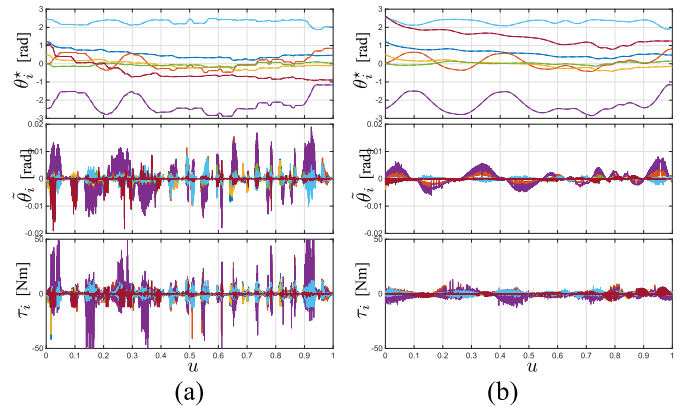


Fig. 13. Reference joint trajectories θ_i^* , joint tracking errors $\tilde{\theta}_i = \theta_i^* - \theta_i$, where θ_i is the actual robot position, and control torques τ_i , $i = 1, \dots, 7$, corresponding to the trajectory demonstrated by subject S1 (a) and to the related optimal constrained smoothing B-spline (b). The actual trajectory duration is 57.45 s.

TABLE I
PAV OF JOINT TRACKING ERRORS $\tilde{\theta}_i$ AND CONTROL TORQUES τ_i , $i = 1, \dots, 7$, FOR THE ORIGINAL AND THE OPTIMIZED TRAJECTORY OBTAINED IN EXPERIMENT #1 FOR SUBJECT S1

i	PAV $\{\tilde{\theta}_i\}$			PAV $\{\tau_i\}$		
	Original	Optimized	$\Delta\%$	Original	Optimized	$\Delta\%$
1	0.0101	0.0039	-61.65	41.8444	12.9185	-69.12
2	0.0160	0.0039	-75.38	43.7600	12.2010	-72.11
3	0.0100	0.0038	-61.56	37.1008	13.1934	-64.43
4	0.0190	0.0081	-57.34	59.9982	21.7350	-63.77
5	0.0055	0.0013	-76.11	11.7998	5.1917	-56.00
6	0.0101	0.0022	-77.73	12.0000	11.0771	-7.69
7	0.0191	0.0026	-86.30	12.0000	9.0137	-24.88

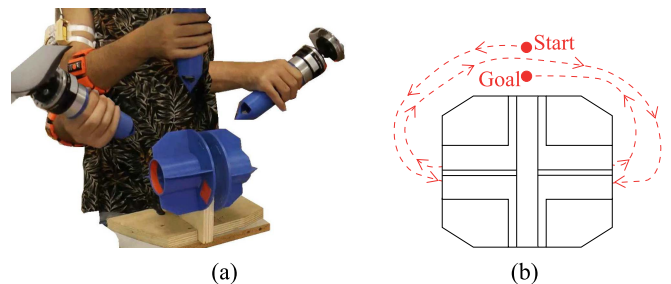


Fig. 14. (a) Setup used in the second experiment. (b) Desired task.

specific locations according to a task that, besides a specific motion, also requires a change in the orientation. More specifically, as can be observed in Fig. 14, from the starting point, the robot end-effector has to be tilted while moved down-left in order to be inserted in the first required location and then carried back to the initial position to repeat the same task on the left side with opposite inclination. The operator must impose a precise robot configuration on the two trajectory ends, while during the rest of the motion, the variation with respect to the “ideal” trajectory can be large. For this reason, the user must impose a high level of stiffness in the neighborhood of the mock-up welding points, while maintaining a low level elsewhere. This behavior can be easily achieved, as highlighted in Fig. 15, showing the

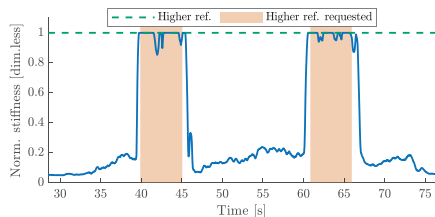


Fig. 15. Stiffness modulation for subject S1 during the tilted welding points session.

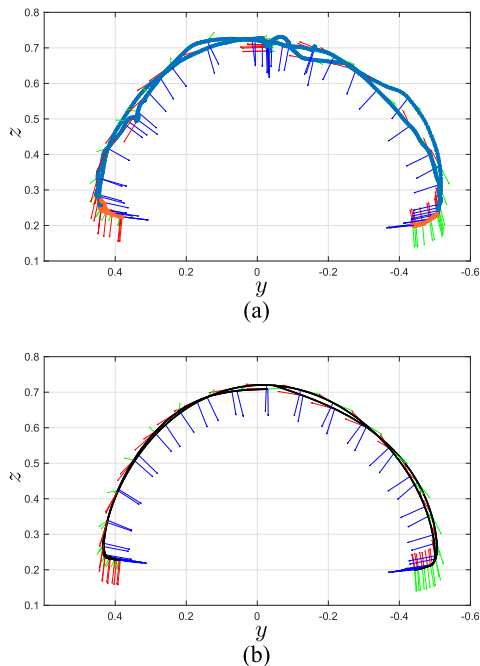


Fig. 16. Second experiment. (a) Position/orientation trajectory taught by the user (the color of the points denotes the stiffness level in $[0, 1]$). (b) Optimized smoothing B-spline. The arrows with RGB colors are, respectively, the x , y , and z axes of the local reference frame.

stiffness modulation for a single subject during the test. The same behavior is confirmed for all the subjects of the experiment, and we refer to Section VI-A for a dedicated analysis of the related aggregated results.

Since in this application both position and orientation are involved, the optimization problem for defining the robot trajectory has been split into two parts.

- 1) For the position, the same bounds considered in the previous test have been assumed (i.e., $\rho_{\min} = 0.005$ m and $\rho_{\max} = 0.025$ m), and the optimization problem has been based on cylindrical constraints.
- 2) The optimized orientation, parameterized with roll–pitch–yaw angles, has been obtained by considering smoothing B-splines with polyhedral constraints, where the bounds ρ for all the angles have been obtained from the normalized stiffness by setting $\rho_{\min} = 0.1^\circ$ and $\rho_{\max} = 1^\circ$. Therefore, $\rho \in [0.1^\circ, 1^\circ] = [0.0017, 0.0175]$ rad.

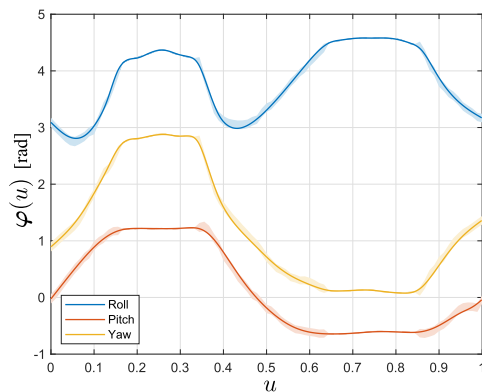


Fig. 17. Roll–pitch–yaw angles defining the orientation of the robot end-effector as a function of the independent variable $u \in [0, 1]$.

In Fig. 16, the trajectory trained by subject S1 and the final trajectory are compared, and the benefits of the proposed method in regularizing the trajectory, eliminating unwanted motions of the operator within the limits imposed by the application, are clear. The fact that the two-way trajectories are almost coincident is worth noting. To better highlight the advantages of the procedure not only for the position but also for the orientation, in Fig. 17, the roll, pitch, and yaw angles of the original motion (with the given bounds) and of the final one are reported as a function of the normalized variable u , showing that, where allowed by the application, this approach tends to eliminate all the oscillations that affect the considered signals.

VI. DISCUSSION

In the following, observations related to the experimental findings reported in the previous section are discussed. By interpreting the obtained results with respect to the experimental task carried out, hardware/software choices, and future perspectives, specific aspects are analyzed and contextualized for a wider insight into the study of the proposed framework.

A. Aggregated Analysis of the Hand Stiffness Modulation

In order to analyze the aggregated results in relation to the hand stiffness modulation within Experimental Session 1 (as described in Section V-A), the data of all subjects have been grouped on the basis of the requested reference level; see the box plots reported in Fig. 18. From the aggregated results, it can be seen that the different stiffness levels are matched by all the subjects, with acceptable deviations (a maximum error of 0.22 and 0.096 is reported for the higher and lower reference bands) and, importantly, reporting for a statistically significant difference among all the requested reference levels. For the statistical analysis, we have set the statistical significance to $p < 0.001$ and conducted a one-way repeated measures analysis of variance (ANOVA) [53]. The Shapiro–Wilk test for normality and Mauchly’s test for sphericity has been checked, concluding that the ANOVA assumptions have been not violated [53], [54]. The ANOVA results reveal a statistically significant difference among all levels, $F(2, 6) = 63.15$. The ability of the subjects

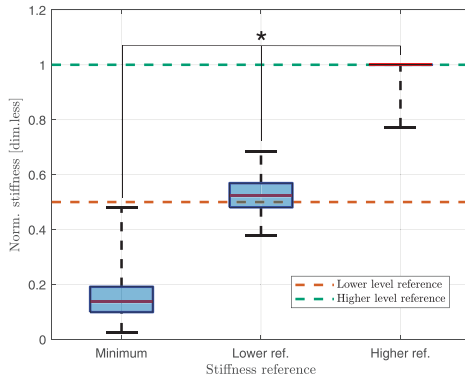


Fig. 18. Box plot for the obstacle course session. Symbol “*” indicates statistical significance.

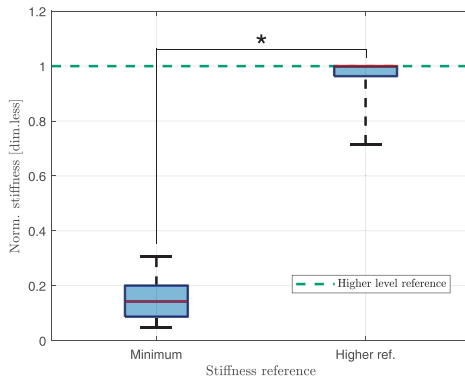


Fig. 19. Box plot for the tilted welding points session. Symbol “*” indicates statistical significance.

in modulating the stiffness during the robot teaching task, as requested by the instructed levels, is therefore supported by statistical evidence.

The capability of the subjects to correctly modulate the hand stiffness during the programming task is also confirmed in relation to Experimental Session 2 (as described in Section V-A). The aggregated data are reported in the box plots of Fig. 19, which group the results on the basis of the requested lower and higher reference levels. Similarly to Experimental Session 1, a statistical analysis based on ANOVA has been conducted, setting the statistical significance to $p < 0.001$. The Shapiro–Wilk test and Mauchly’s test report that the ANOVA assumptions are not violated. The ANOVA reveals a statistically significant difference between the requested stiffness modulation levels, $F(1, 9) = 24.62$, and $p < 0.001$. All the subjects are, therefore, statistically able to successfully modulate the stiffness (a maximum error of 0.3 is reported for the higher reference bands) according to requested reference levels in the presence of different orientations of the end-effector during the sEMG-enhanced kinesthetic teaching task.

According to the analyzed aggregated results, the proposed approach is, therefore, capable to provide the user with an

TABLE II
PEAK AND MEAN VALUES OF THE NORM OF VELOCITY AND ACCELERATION DURING EXPERIMENT #1 FOR SUBJECT S1

	Velocity		Acceleration	
	Peak	Mean	Peak	Mean
Original Trj	15.6483	3.9882	6.2120e+03	1.1664e+03
Optimized Trj	5.7571	3.1987	435.4710	138.1530

additional programming input channel, which could be *voluntarily* modulated to program a specific parameter of a constrained B-spline optimization of the trajectory. Particularly, this highlights a clear improvement with respect to state-of-the-art works in which the sEMG information is exploited to *observe* a biomechanical behavior of the user, as in teleimpedance-based approaches. Indeed, in the proposed approach, the programming modality estimated from sEMG sensors is not “indirectly” emerging from human motions; instead, it is arbitrarily exploited as a programming input in accordance with the presence of a real-time vibrotactile feedback informing the user about the current value of the modulated hand stiffness level. Note that the presence of a real-time “biofeedback” represents a novelty with respect to state-of-the-art approaches. Another aspect of novelty also regards the usage of sEMG to estimate the level of stiffness of the human hand instead of the stiffness at the arm endpoint: this allowed us to provide the users with a programming input that could be intuitively and voluntarily modulated even during the performance of a robot kinesthetic teaching with both arms.

B. Smoothness Characteristics of the Optimized Trajectory

With reference to the results reported in Section V-A for Experimental Session 1, in Table II, the peak and mean values of the norm of velocity and acceleration of the trajectory demonstrated by subject S1 and the corresponding constrained smoothing B-spline are reported. By comparing the numerical values related to S1, it comes out that peak values of both velocity and acceleration are considerably lowered by the optimization process (-63.20% and -92.98% , respectively). The average velocity is reduced by -19.79% . In this case, since the duration of both the trajectory (original and optimized) is the same, this decrease can be explained by a reduction of the total traveled distance: in fact, the procedure that makes the trajectory smoother tends to reduce its length, by avoiding unwanted oscillations. Similar conclusions can be drawn by observing the data aggregated for all the subjects in Fig. 20. The peak and mean values of velocity and acceleration are reduced by similar amounts. In addition, it is possible to observe that the optimization process based on constrained smoothing B-splines causes a general reduction of the width of the values’ distribution about the median element. This means that the influence of the single subject on (these features of) the final trajectory tends to become negligible.

In addition, the capability of the proposed procedure to increase the smoothness of the original trajectory at the price of larger approximation errors is confirmed by an analysis of the cost function (5), or equivalently of (9), and in particular of the two contributions J_1 and J_2 . Term J_1 takes into account the overall amount of the squared errors of the optimal curve

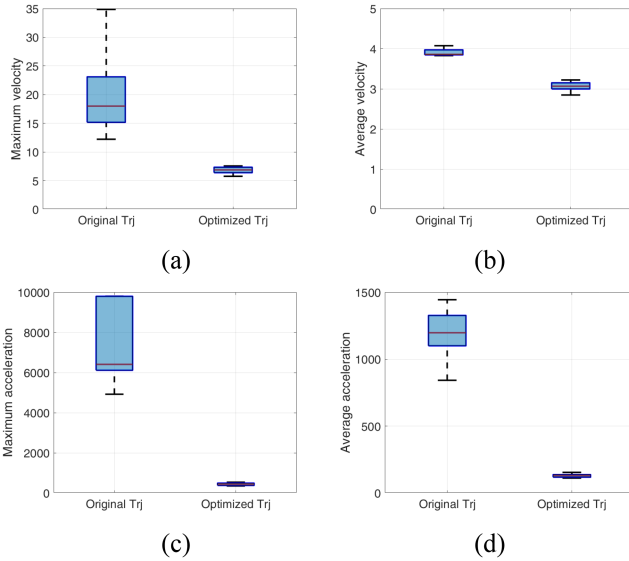


Fig. 20. (a)–(d) Box plots with peak and mean values of the norm of velocity and acceleration for all the subjects involved in experiment #1.

TABLE III
NUMERICAL VALUES OF THE TERMS $J_1 := \sum_{j=0}^n w_j \|s(u_j^*) - \mathbf{q}_j\|^2$ AND $J_2 := \int_{u_{\min}}^{u_{\max}} \left\| \frac{d^2 s(\tau)}{d\tau^2} \right\|^2 d\tau$ COMPOSING THE COST FUNCTION FOR THE DEFINITION OF SMOOTHING B-SPLINES OBTAINED BY OPTIMIZING THE TRAJECTORY DEMONSTRATED BY SUBJECT S1 IN EXPERIMENT #1 WITH DIFFERENT PARAMETERS' VALUES

Trajectory type	J_1	J_2
Approximated B-spline ($\lambda = 0$)	0.0154	6.7831e+05
Smoothing B-spline ($\lambda = 10^{-5}$)	0.1606	1.8483e+04
Constrained smoothing B-spline ($\lambda = 10^{-5}, \rho = \rho^*$)	0.0480	1.2120e+05
Constrained smoothing B-spline ($\lambda = 10^{-5}, \rho = \rho^*/2$)	0.0231	3.0095e+05
Constrained smoothing B-spline ($\lambda = 10^{-5}, \rho = 2\rho^*$)	0.0650	6.8525e+04

with respect to the original trajectory, while J_2 is a measure of the smoothness of the optimal curve based on the integral of the acceleration. The data, reported in Table III and related to the experiment with subject S1, reveal that the constrained smoothing B-spline has an intermediate behavior between a simple approximating spline, which takes into account only the approximation error by assuming $\lambda = 0$ and accordingly minimizes J_1 , and a standard smoothing B-spline without any additional constraint. Interestingly enough, by reducing the size of the bound ρ with respect to the initial value denoted by ρ^* , the approximation error J_1 decreases, while the term J_2 becomes larger. This means that, if the admissible region becomes smaller, the optimal trajectory tends toward an approximating curve. On the other hand, by relaxing the constraint, e.g., by considering $\rho = 2\rho^*$, the approximation error grows, but the smoothness of the curve is improved, being J_2 smaller. In this case, the constrained curve gets closer to an unconstrained smoothing B-spline. It is possible to conclude that with the introduction of the constraints in the optimization process, it is possible to transform a subjective choice of the parameters, i.e., λ , whose final effect on the overall trajectory cannot be predicted *a priori*, into an objective parameter selection, since the constraints (ρ_{\min}

TABLE IV
QUALITATIVE EVALUATION OF VIBROTACTILE FEEDBACK

Outcome Type	Questionnaire	Average Score (\pm std dev.)
Perceived ease of use (PE)	<i>It was easy to interpret the vibrotactile feedback to the aim of hand stiffness modulation.</i>	6.5 \pm 0.52
Perceived usefulness (PU)	<i>I think the vibrotactile feedback is useful to voluntarily modulate the hand stiffness level during kinesthetic teaching.</i>	6.2 \pm 0.63
Comfort (C)	<i>It was comfortable to feel the vibrotactile feedback on the upper arm's skin.</i>	6 \pm 0.52

Scores: likert scale from 1 (entirely disagree) to 7 (entirely agree).

and ρ_{\max}) depend on the geometry of the problem, while it is sufficient to assume λ small enough.

Therefore, the proposed approach allows the operators to program trajectories that are smoother with respect to the ones demonstrated by means of kinesthetic teaching only, with also the advantage of having the possibility to specify spatial precision constraints. Importantly, at the same time, the resulting trajectory is completely determined in geometrical terms and, therefore, applicable by definition to any timing law. This is a clear advantage with respect, for example, to state-of-the-art methods for task parameterization, based, e.g., on DMPs, that require the knowledge of additional motion information, such as velocity and acceleration, which are typically affected by noise when computed from position information.

C. Subjective Vibrotactile Feedback Evaluation

In accordance with the architecture described in Section IV, the subjects performed the experimental sessions described in the previous section by exploiting a vibrotactile feedback conveyed on the upper arm's skin by a wearable bracelet, with a range of vibration intensity proportional to the modulated hand stiffness level. In order to qualitatively evaluate its usage, we provided the subjects with a questionnaire to subjectively assess the influence of the vibrotactile feedback during the experimental session executions. In particular, the questionnaire required the subjects to rate three statements related to the perceived ease of use (PE), perceived usefulness (PU), and comfort (C) in exploiting the vibrotactile feedback, on the basis of a Likert scale from 1 (entirely disagree) to 7 (entirely agree), as reported in Table IV. As can be observed by the questionnaire results reported in the table, all the average scores computed over the subjects were greater than or equal to 6, showing a qualitative positive evaluation of the interpretability and usefulness of the vibrotactile feedback for the voluntary modulation of the hand stiffness level during the experimental tasks.

D. Hand Stiffness Level and Trajectory Sharpness

Sections V-A and V-B report how the subjects involved in the experiments were successfully able to modulate their hand

stiffness, according to the levels instructed by the task protocol during the physical teaching of end-effector robot trajectories. However, the characteristics of the trajectories that have been taught by the subjects with respect to the three different levels of stiffness (minimum, lower, and higher) are worth analyzing. In particular, we want to have an insight into the relationship between the sharpness of the human motion/demonstrated trajectories and different modulations of the stiffness, since this may have an impact on the actual capability of performing desired robot teaching as well as on how the robot is perceived by the user during the physical interaction, and then, we also want to check the effect obtained by the application of the constrained B-spline trajectory optimization. To this aim, we use jerk measures to quantify the level of trajectory sharpness, since it has been shown that—in a comparison scenario—the lower the trajectory’s jerk amplitude, the more smooth and natural the performed motion (minimal jerk of the human motion [55]). Conversely, higher jerk amplitudes correspond to motions/trajectories with higher sharpness. Specifically, we consider the entire sample of jerk values shown by each subject over the robot teaching tasks. On this basis, we report for the mean absolute value (MAV) of the trajectory jerk, which is simply the average value of all the rectified jerk values, and for the mean higher absolute value (MHAV) of the trajectory jerk, which is the average value over the rectified jerk values that are greater than 1.5 times the interquartile range (i.e., the mean value of the higher outlier values w.r.t. the jerk values distribution itself) for each subject and grouped according to the different levels of stiffness modulation during the experiment executions. According to their definitions, the MAV of the jerk is related to the average trajectory sharpness, whereas the MHAV is related to less frequent sharper behaviors that are not appreciable looking only at the total average jerk. Fig. 21(a) and (b) reports the box plot of the jerk MAV and MHAV of the demonstrated trajectory for the subjects, grouped by the different hand stiffness levels performed during the teaching task. Since the Shapiro–Wilk test reports that the considered data groups violate the assumption of normality, the Kruskal–Wallis test was performed, revealing that both the MAV and MHAV jerk in Fig. 21(a) and (b) were affected by the level of stiffness, $H(2) = 13.65$, $p < 0.01$. Thereafter, a Tukey test was performed for pairwise comparison, reporting statistically significant smaller values for both the MAV and MHAV jerk in the case of the higher stiffness level with respect to the minimum and lower stiffness levels. We can, therefore, conclude that during the teaching of trajectory paths with higher stiffness, the subject performed less sharp trajectories with statistical evidence. This is reasonably related to the fact that, when the subjects most stiffened their hand, they were less prone to execute motions characterized by rapid oscillations and abrupt variations with respect to the case with lower stiffness levels. Therefore, higher stiffness may be interpreted as an injection of higher damping in the motion generation process of the user during the teaching phase. Although this aspect did not affect the correct execution of the tasks, as described in Sections V-A and V-B, we will conduct future investigations to further study and characterize the role of this behavior. On the other

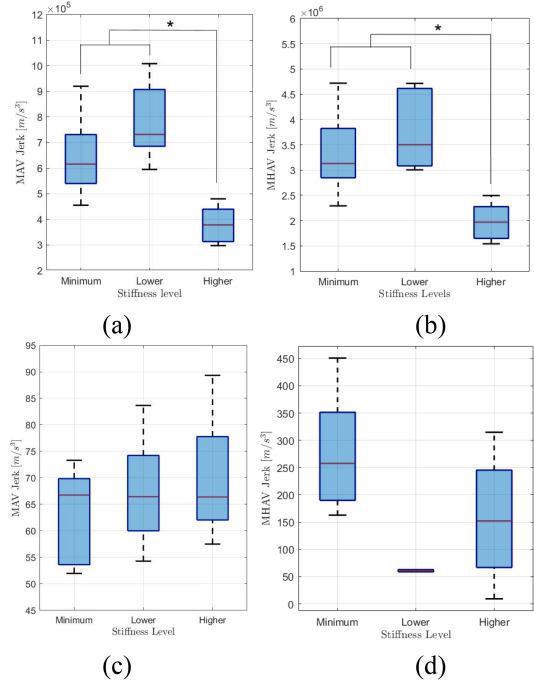


Fig. 21. Box plot of the trajectory MAV and MHAV jerk (a),(b) demonstrated by the subjects and (c),(d) after the B-spline optimization, grouped by the different hand stiffness levels modulated during the teaching task. Symbol “*” indicates statistical significance.

hand, looking at Fig. 21(c) and (d), it is possible to observe the MAV and MHAV jerk after the application of the B-spline trajectory optimization. As expected, it is appreciable how the jerk amplitude was tremendously reduced by five to six orders of magnitude, with no statistically significant difference among the trajectory sections related to the different levels of stiffness, as reported by the Kruskal–Wallis test based analysis ($H(2) = 0.8$, $p = 0.671$) performed in an analogous manner as for the data of Fig. 21(a) and (b). An aspect of particular interest observable by looking at Fig. 21(c) and (d) is related to the fact that the B-spline optimization almost completely cancels the presence of MHAV jerk values for the case of the lower stiffness level, since, in this case, straight paths were taught by the user to the robot end-effector trajectory (see Fig. 9). This result has been allowed by the possibility of selecting smoothing parameters α particularly small because the constrained optimization problem guarantees, in any case, the compliance with the limits on the maximum deviation with respect to the original trajectory, without the need for a difficult tradeoff between smoothness of the curve and interpolation error.

E. Future Work and Perspectives

The results obtained so far are based on two pillars, which represent also the two main novelties of the article:

- 1) smoothing B-splines enhanced with a bound on the maximum deviation with respect to the demonstrated trajectory;

- 2) sEMG signals related to the modulation of the hand stiffness of the operator for the continuous specification of this bound.

The former contribution has been completely developed with the proposed two-step optimization procedure that guarantees the definition of a minimum complexity trajectory, i.e., characterized by the minimum number of parameters (in the specific case, the control points of the B-spline function) compliant with the bound on the maximum deviation from the demonstrated path. The latter aspect will be the subject of further research. In particular, the use of sEMG signals as an independent communication channel between the user and the robotic system will be addressed to improve PbD tasks. Regarding the proposed application, the mapping between the muscle activation and the desired level of precision will be improved, by studying more in depth the role of vibrotactile feedback. In addition, novel PbD methods, based on multimodal interaction, namely force and stiffness, between the user and the robot will be developed.

1) *Mapping Between Muscles Activation and Desired Precision*: Future studies will include the consideration of different types of mapping relations between muscle activation and desired precision constraints that the user wants to instruct to the system. In particular, since from biomechanics, it is known that the muscle activation characteristics are nonlinear [38], a very interesting perspective regards the investigation of a nonlinear mapping also between muscle activation and precision-level modulation in terms of intuitiveness for the users. Furthermore, since in the present work we have implemented a linear mapping that actually enabled the users to modulate the hand stiffness level in accordance with the levels requested by the tasks/experimenters, we can observe how the presence of the vibrotactile feedback played an important role in allowing the users “compensating” for the nonlinear muscle activation characteristics by means of their cognitive control exploiting the provided vibrotactile feedback. We are, therefore, planning to carry out a future dedicated study in order to determine the most intuitive mapping relation between muscle activation and precision level modulation. Such future investigation should include properly designed experimental protocols in which a group of subjects will test different combinations of: 1) linear mappings; 2) nonlinear mappings; and 3) presence/absence of vibrotactile feedback, during kinesthetic teaching.

2) *Vibrotactile Feedback Influence*: Future experiments will also be considered to investigate the influence of the vibrotactile feedback on the modulation of the hand stiffness level and related precision constraint of the trajectory. More specifically, we plan to carry out tests of the capability of the users in modulating the hand stiffness level/inferred trajectory precision during kinesthetic teaching tasks with and without the presence of vibrotactile feedback. In addition, we will also further analyze the influence of the vibrotactile feedback by comparing vibration profiles different from the one used in the present work (i.e., modulation of the intensity of a continuous vibration), such as the modulation of the frequency of intermittent vibrations, and more structured profiles combining asynchronous “vibrations bursts” when certain meaningful values of the hand stiffness level are matched by the users. Finally, we also plan to test

different body locations for the placement of the vibrotactile device, such as in the proximity of the wrist or ankle.

3) *Novel PbD Methods Based on Stiffness Modulation*: Stiffness modulation can be exploited not only for specifying the desired precision of the trajectory to be planned, but, being an additional input channel, it can also be used for other different forms of interaction between the robot and the user. Future research will focus on the possibility for the human to physically interact with the (already) planned trajectory and locally reshape it according to the needs by modifying the stiffness level of the hand.

VII. CONCLUSION

The proposed multimodal teaching approach, based on kinesthetic interaction and information extracted from forearm sEMG, can be effectively used to enhance robot PbD procedures. The estimation of the desired accuracy level from the hand stiffness of the user that guides the robot along the desired path allows obtaining a smooth motion trajectory with a one-shot procedure that combines intuitiveness for the user, due to the natural tendency of humans to modulate their stiffness according to the requested precision, and optimality of the trajectory in terms of minimum curvature/acceleration within the bounds imposed by the tasks. Based on a single demonstration, this method leads to a significant reduction of the time and effort required to program robotic tasks, with an important impact on industrial applications, at the price of using a (low-cost) wearable device for recording sEMG signals.

APPENDIX A

B-SPLINE DEFINITION AND MAIN PROPERTIES

A B-spline of degree p is a parametric curve $\mathbf{s} : [u_{\min}, u_{\max}] \rightarrow \mathbb{R}^d$ defined as convex combinations of *control points* $\mathbf{p}_j \in \mathbb{R}^d$ weighted by *B-spline basis functions*¹ of degree p , $B_j^p(u)$

$$\mathbf{s}(u) = \sum_{j=0}^N \mathbf{p}_j B_j^p(u), \quad u_{\min} \leq u \leq u_{\max}.$$

Let $\mathbf{u} = [u_0, \dots, u_m]$ be a vector of real numbers (called *knots*), with $u_j \leq u_{j+1}$. The j th B-spline basis function of degree p is defined, according to the Cox–de Boor recursive formula [56], as

$$B_j^p(u) = \frac{u - u_j}{u_{j+p} - u_j} B_j^{p-1}(u) + \frac{u_{j+p+1} - u}{u_{j+p+1} - u_{j+1}} B_{j+1}^{p-1}(u) \quad (18)$$

with

$$B_j^0(u) = \begin{cases} 1, & \text{if } u_j \leq u < u_{j+1} \\ 0, & \text{otherwise} \end{cases}.$$

Note that $B_j^p(u) \geq 0 \forall u$, and, in particular, $B_j^p(u) = 0$ everywhere except in the interval $[u_j, u_{j+p+1})$. The number of control points, $N + 1$, the number of the knot, $m + 1$, and the degree of the B-spline are related by $m = N + p + 1$.

¹The basis functions that appear in the definition of a B-spline are usually called *basis B-splines*.

Since the focus of this article is on the definition of the geometric path that the robot must track to accomplish a given task, the knots, that implicitly determine how this path is covered, are considered with a uniform distribution, i.e.,

$$\mathbf{u} = \underbrace{[u_{\min}, \dots, u_{\min}, \dots, u_j, \dots, u_{\max}, \dots, u_{\max}]}_{p+1}$$

where the internal knots u_j are computed according to

$$u_j = u_{\min} + j \frac{u_{\max} - u_{\min}}{N - p + 1}, \quad j = 1, \dots, N - p.$$

Therefore, fixing the degree p of the B-spline (in robotics applications, cubic B-splines, i.e., $p = 3$, are usually considered because of the continuity of velocities and accelerations) and the support $[u_{\min}, u_{\max}]$ where the curve is defined (typically $= [0, 1]$, but other choices are also possible, the B-spline curve is completely defined by the sequence of control points \mathbf{p}_j , $j = 0, \dots, N$.

APPENDIX B SMOOTHING B-SPLINE DERIVATION

Smoothing B-splines are standard B-spline functions $\mathbf{s}(u)$ whose control points \mathbf{p}_j are determined by minimizing the cost function

$$J := \sum_{j=0}^n w_j \|\mathbf{s}(u_j^*) - \mathbf{q}_j\|^2 + \lambda \int_{u_{\min}}^{u_{\max}} \left\| \frac{d^2 \mathbf{s}(\tau)}{d\tau^2} \right\|^2 d\tau \quad (19)$$

where $\lambda \geq 0$ is a free parameter, and which, therefore, represent a tradeoff between the (squared) approximation error with respect to the given via points \mathbf{q}_j , $j = 0, \dots, n$, and the smoothness of the curve, which is measured by taking into account its curvature/second derivative. Generally, the number $N + 1$ of control points defining the spline is equal to the number $n + 1$ of data points to be approximated. However, when the number of data points is very large, it may be convenient for computational reasons to choose $n > N$ and, in particular, $n \gg N$. By recalling that points \mathbf{q}_j , $j = 0, \dots, n$, are obtained by sampling the demonstrated trajectory with a constant period T_s , one can assume a fixed ratio between n and N , by considering a new control point every m via points. In this case, the values of u , at which the approximation of points \mathbf{q}_j occurs, are obtained by normalizing the actual sampling instant jT_s with respect to the total duration nT_s of the trajectory, i.e.,

$$u_j^* = \frac{j}{n}, \quad j = 0, \dots, n. \quad (20)$$

By exploiting (6) and the relationship between the derivatives of a B-spline and its control points (for more details, see [26]),

the cost function (19) can be written in a matrix form as

$$\begin{aligned} J &= J_1 + \lambda J_2 \text{ with} \\ J_1 &= \text{tr} \left((\hat{\mathbf{Q}} - \mathbf{B}\mathbf{P})^T \mathbf{W} (\hat{\mathbf{Q}} - \mathbf{B}\mathbf{P}) \right) \\ J_2 &= \text{tr} \left((\mathbf{C}\mathbf{P} + \hat{\mathbf{P}})^T \mathbf{A} (\mathbf{C}\mathbf{P} + \hat{\mathbf{P}}) \right) \end{aligned} \quad (21)$$

where $\text{tr}(\cdot)$ denotes the trace of a matrix, $\mathbf{W} = \text{diag}(w_j) > 0$, $j = 0, \dots, n$, is the weight matrix, and $\mathbf{P} = [\mathbf{p}_1, \dots, \mathbf{p}_{N-1}]^T$ is the matrix with the internal control points² to be determined. Note that the first and the last via points are supposed to be perfectly interpolated, i.e.,

$$\mathbf{s}(u_{\min}) = \mathbf{q}_0 \text{ and } \mathbf{s}(u_{\max}) = \mathbf{q}_n$$

and, accordingly,

$$\mathbf{p}_0 = \mathbf{q}_0 \text{ and } \mathbf{p}_N = \mathbf{q}_n.$$

The analytic expression of the other matrices appearing in the cost function (19) is reported below. The matrix $\mathbf{B} \in \mathbb{R}^{(n-1) \times (N-1)}$ is built up with the vector of the basis functions at “time instants” u_j^* , $j = 1, \dots, n - 1$:

$$\mathbf{B} = \begin{bmatrix} B_1^p(u_1^*) & B_2^p(u_1^*) & \dots & B_{N-1}^p(u_1^*) \\ B_1^p(u_2^*) & B_2^p(u_2^*) & \dots & B_{N-1}^p(u_2^*) \\ \vdots & \vdots & \ddots & \vdots \\ B_1^p(u_{n-1}^*) & B_2^p(u_{n-1}^*) & \dots & B_{N-1}^p(u_{n-1}^*) \end{bmatrix}.$$

For cubic B-spline ($p = 3$), the expressions of matrices $\mathbf{A} \in \mathbb{R}^{(N-1) \times (N-1)}$ and $\mathbf{C} \in \mathbb{R}^{(N-1) \times (N-1)}$ are unnumbered equation shown at the bottom of this page.

$$\text{and } \mathbf{C} = \begin{bmatrix} c_{0,2} & c_{0,3} & 0 & \dots & 0 \\ c_{1,1} & c_{1,2} & c_{1,3} & 0 & \vdots \\ 0 & & & \ddots & 0 \\ \vdots & & 0 & c_{N-3,1} & c_{N-3,2} & c_{N-3,3} \\ 0 & \dots & 0 & c_{N-2,1} & c_{N-2,2} \end{bmatrix}$$

where

$$\begin{cases} c_{j,1} = \frac{6}{u_{j+1,j-1} u_{j+1,j-1}} \\ c_{j,2} = -\frac{6}{u_{j+1,j-1}} \left(\frac{1}{u_{j+1,j-1}} + \frac{1}{u_{j+2,j-1}} \right) \\ c_{j,3} = \frac{6}{u_{j+1,j-1} u_{j+2,j-1}} \end{cases}$$

²For the sake of simplicity, we use the same name of the matrix containing all the control points, which has been defined in (8). Note that the difference between the two matrices only consists of the first and last control points, which are already known.

$$\mathbf{A} = \begin{bmatrix} 2u_{1,0} & u_{1,0} & 0 & \dots & 0 \\ u_{1,0} & 2(u_{1,0} + u_{2,1}) & u_{2,1} & 0 & \vdots \\ \vdots & & \ddots & & 0 \\ 0 & & & u_{N-2,N-3} & 2(u_{N-2,N-3} + u_{N-1,N-2}) & u_{N-1,N-2} \\ 0 & \dots & 0 & u_{N-1,N-2} & 2u_{N-1,N-2} \end{bmatrix}$$

and $u_{i,j} = u_i - u_j$ with

$$u_j = \begin{cases} u_{\min}, & \text{if } j < 0 \\ u_{\min} + j \frac{u_{\max} - u_{\min}}{N-2}, & \text{if } j = 0, \dots, N-2 \\ u_{\max}, & \text{if } j > N-2 \end{cases}.$$

The vector $\hat{\mathbf{Q}}$ is composed of a (slightly) modified version of the recorded input trajectory, i.e.,

$$\hat{\mathbf{Q}} = [\hat{\mathbf{q}}_1, \dots, \hat{\mathbf{q}}_{n-1}]^T$$

where

$$\hat{\mathbf{q}}_j = \mathbf{q}_j - B_0^p(u_j^*)\mathbf{q}_0 - B_n^p(u_j^*)\mathbf{q}_n$$

with $\mathbf{q}_j, j = 0, \dots, n$, being the samples of the original trajectory, and finally

$$\hat{\mathbf{P}} = [c_{0,2}\mathbf{q}_0, \underbrace{\mathbf{0}, \dots, \mathbf{0}}_{N-4}, c_{N-2,3}\mathbf{q}_n]^T$$

where $\mathbf{0}$ are column vectors of proper length, whose elements are all zero.

The procedure that leads to the definition of these matrices, even if only in the particular case $N = n$, can be found in [26].

Remark 1: Matrix \mathbf{W} is symmetric and positive definite by construction. A simple inspection reveals that tridiagonal matrix \mathbf{A} is symmetric. Its positive definiteness can be easily inferred by applying the *Gershgorin circle theorem* (see [57, Th. 1.2]). Both matrices can, therefore, be rewritten as $\mathbf{\Gamma}^T \mathbf{\Gamma} = \mathbf{W}$ and $\mathbf{\Omega}^T \mathbf{\Omega} = \mathbf{A}$, respectively. Finally, it can be shown that, because of the choice of u_j and u_j^* , matrices \mathbf{B} (see [58, Th. 5.18]) and \mathbf{C} are full rank.

Remark 2: The trace function that appears in (21) is necessary to deal with the vectorial form of each control point \mathbf{p}_j that leads to a matrix form of the unknown \mathbf{P} . The cost function (21) can be rewritten as

$$J = \|\mathbf{\Gamma}(\hat{\mathbf{Q}} - \mathbf{B}\mathbf{P})\|_F^2 + \lambda \|\mathbf{\Omega}(\mathbf{C}\mathbf{P} + \hat{\mathbf{P}})\|_F^2 \quad (22)$$

where $\|\mathbf{x}\|_F = \sqrt{\text{tr}(\mathbf{x}^T \mathbf{x})}$ denotes the *Frobenius norm* of matrix \mathbf{X} [59], while the meaning of the other symbols is the same as in (21). By rearranging the components of control points \mathbf{p}_j in \mathbf{P} to obtain a column vector of length $(N-1)d$, and modifying the matrices that appear in (21) accordingly, the objective function becomes a combination of the square of two standard Euclidean norms at the price of a less compact notation and an increase of the problem dimension by a factor of d .

Since matrices \mathbf{B} and \mathbf{C} are full rank, the two functions $J_1(\mathbf{P})$ and $J_2(\mathbf{P})$ are both strictly convex. Thus, the whole objective function (21), which is a positive combination of these functions, is strictly convex too. For this reason, the optimal set of the problem $\min_{\mathbf{P}} J(\mathbf{P})$ contains at most one point. By differentiating (21) with respect to \mathbf{P} and equating the result to zero, the analytical solution of this minimization problem can be derived in an analytical form

$$\mathbf{P}^* = (\mathbf{B}^T \mathbf{W} \mathbf{B} + \lambda \mathbf{C}^T \mathbf{A}^T \mathbf{C})^{-1} (\mathbf{B}^T \mathbf{W} \hat{\mathbf{Q}} - \lambda \mathbf{C}^T \mathbf{A}^T \hat{\mathbf{P}}). \quad (23)$$

Since matrix $\mathbf{B}^T \mathbf{W} \mathbf{B} + \lambda \mathbf{C}^T \mathbf{A}^T \mathbf{C}$ is strictly positive, its inverse is well defined, and the unique solution \mathbf{P}^* of the minimization problem can be found.

REFERENCES

- [1] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, *Robot Programming by Demonstration*. Berlin, Germany: Springer, 2008, pp. 1371–1394.
- [2] V. Villani, F. Pini, F. Leali, C. Secchi, and C. Fantuzzi, “Survey on human-robot interaction for robot programming in industrial applications,” *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 66–71, 2018.
- [3] *Industrial Robots and Robot System Safety (Section IV, Chapter 4)*, OSHA Technical Manual Std. TED 01-00-015, Occupational Safety and Health Administration, Jan. 9, 2021.
- [4] L. Bascetta, G. Ferretti, G. Magnani, and P. Rocco, “Walk-through programming for robotic manipulators based on admittance control,” *Robotica*, vol. 31, no. 7, pp. 1143–1153, 2013.
- [5] F. Ferraguti, C. T. Landi, C. Secchi, C. Fantuzzi, M. Nolli, and M. Pesamosca, “Walk-through programming for industrial applications,” *Procedia Manuf.*, vol. 11, pp. 31–38, 2017.
- [6] H. Martin, K. Nilsson, and P. J. Norberto, *Industrial Robotics*. Berlin, Germany: Springer, 2008, pp. 963–986.
- [7] L. Piegl and W. Tiller, *The NURBS Book*, 2nd ed. New York, NY, USA: Springer, 1996.
- [8] S. Calinon, “A tutorial on task-parameterized movement learning and retrieval,” *Intell. Service Robot.*, vol. 9, no. 1, pp. 1–29, 2015.
- [9] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel, “Dynamic movement primitives in robotics: A tutorial survey,” 2021, *arXiv:2102.03861*.
- [10] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, “Dynamical movement primitives: Learning attractor models for motor behaviors,” *Neural Comput.*, vol. 25, no. 2, pp. 328–373, 2013.
- [11] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, “Probabilistic movement primitives,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 2616–2624.
- [12] Y. Huang, L. Rozo, J. Silvério, and D. G. Caldwell, “Kernelized movement primitives,” *Int. J. Robot. Res.*, vol. 38, no. 7, pp. 833–852, 2019.
- [13] M. Schneider and W. Ertel, “Robot learning by demonstration with local Gaussian process regression,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 255–260.
- [14] S. Calinon, F. Guenter, and A. Billard, “On learning, representing, and generalizing a task in a humanoid robot,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 2, pp. 286–298, Apr. 2007.
- [15] N. Delson and H. West, “Robot programming by human demonstration: The use of human inconsistency in improving 3D robot trajectories,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 1994, vol. 2, pp. 1248–1255.
- [16] M. Ostanin, D. Popov, and A. Klimchik, “Programming by demonstration using two-step optimization for industrial robot,” *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 72–77, 2018.
- [17] Y. Maeda, T. Ushioda, and S. Makita, “Easy robot programming for industrial manipulators by manual volume sweeping,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 2234–2239.
- [18] J. Aleotti and S. Caselli, “Robust trajectory learning and approximation for robot programming by demonstration,” *Robot. Auton. Syst.*, vol. 54, no. 5, pp. 409–413, 2006.
- [19] D. L. Marino and M. Manic, “Fast trajectory simplification algorithm for natural user interfaces in robot programming by demonstration,” in *Proc. IEEE 25th Int. Symp. Ind. Electron.*, 2016, pp. 905–911.
- [20] C. Groth and D. Henrich, “One-shot robot programming by demonstration using an online oriented particles simulation,” in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2014, pp. 154–160.
- [21] D. Müller, C. Veil, M. Seidel, and O. Sawodny, “One-shot kinesthetic programming by demonstration for soft collaborative robots,” *Mechatronics*, vol. 70, 2020, Art. no. 102418.
- [22] D. Lee and C. Ott, “Incremental kinesthetic teaching of motion primitives using the motion refinement tube,” *Auton. Robots*, vol. 31, nos. 2/3, pp. 115–131, 2011.
- [23] S. Fani et al., “Simplifying telerobotics: Wearability and teleimpedance improves human-robot interactions in teleoperation,” *IEEE Robot. Autom. Mag.*, vol. 25, no. 1, pp. 77–88, Mar. 2018.
- [24] J. Soechting, “Effect of target size on spatial and temporal characteristics of a pointing movement in man,” *Exp. Brain Res.*, vol. 54, no. 1, pp. 121–132, 1984.
- [25] P. L. Gribble, L. I. Mullin, N. Cothros, and A. Mattar, “Role of co-contraction in arm movement accuracy,” *J. Neurophys.*, vol. 89, no. 5, pp. 2396–2405, 2003.
- [26] L. Biagiotti and C. Melchiorri, *Trajectory Planning for Automatic Machines and Robots*, 1st ed. Berlin, Germany: Springer, 2008.
- [27] B. W. Silverman, “Spline smoothing: The equivalent variable kernel method,” *Ann. Statist.*, vol. 12, no. 3, pp. 898–916, 1984.

- [28] L. Peternel, T. Petrič, and J. Babič, “Human-in-the-loop approach for teaching robot assembly tasks using impedance control interface,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 1497–1502.
- [29] A. Ajoudani, N. Tsagarakis, and A. Bicchi, “Tele-impedance: Teleoperation with impedance regulation using a body-machine interface,” *Int. J. Robot. Res.*, vol. 31, no. 13, pp. 1642–1656, 2012.
- [30] G. Lentini, G. Grioli, M. G. Catalano, and A. Bicchi, “Robot programming without coding,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 7576–7582.
- [31] R. Wu, H. Zhang, T. Peng, L. Fu, and J. Zhao, “Human-robot interaction and demonstration learning mode based on electromyogram signal and variable impedance control,” *Math. Probl. Eng.*, vol. 2018, pp. 1–11, 2018.
- [32] C. Yang, C. Zeng, C. Fang, W. He, and Z. Li, “A DMPs-based framework for robot learning and generalization of humanlike variable impedance skills,” *IEEE/ASME Trans. Mechatron.*, vol. 23, no. 3, pp. 1193–1203, Jun. 2018.
- [33] F. Ficuciello, A. Romano, L. Villani, and B. Siciliano, “Cartesian impedance control of redundant manipulators for human-robot co-manipulation,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 2120–2125.
- [34] A. Ajoudani, C. Fang, N. Tsagarakis, and A. Bicchi, “Reduced-complexity representation of the human arm active endpoint stiffness for supervisory control of remote manipulation,” *Int. J. Robot. Res.*, vol. 37, no. 1, pp. 155–167, 2018.
- [35] M. Laghi, A. Ajoudani, M. G. Catalano, and A. Bicchi, “Unifying bilateral teleoperation and tele-impedance for enhanced user experience,” *Int. J. Robot. Res.*, vol. 39, no. 4, pp. 514–539, 2020.
- [36] J. M. Winters, “Hill-based muscle models: A systems engineering perspective,” in *Multiple Muscle Systems*. Berlin, Germany: Springer, 1990, pp. 69–93.
- [37] N. Karavas, A. Ajoudani, N. Tsagarakis, J. Saglia, A. Bicchi, and D. Caldwell, “Tele-impedance based assistive control for a compliant knee exoskeleton,” *Robot. Auton. Syst.*, vol. 73, pp. 78–90, 2015.
- [38] D. A. Winter, *Biomechanics and Motor Control of Human Movement*. Hoboken, NJ, USA: Wiley, 2009.
- [39] E. Burdet, D. W. Franklin, and T. E. Milner, *Human Robotics: Neuromechanics and Motor Control*. Cambridge, MA, USA: MIT Press, 2013.
- [40] D. W. Franklin, R. Osu, E. Burdet, M. Kawato, and T. E. Milner, “Adaptation to stable and unstable dynamics achieved by combined impedance control and inverse dynamics model,” *J. Neurophysiol.*, vol. 90, no. 5, pp. 3270–3282, 2003.
- [41] R. Meattini, S. Benatti, U. Scarcia, D. De Gregorio, L. Benini, and C. Melchiorri, “An sEMG-based human-robot interface for robotic hands using machine learning and synergies,” *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 8, no. 7, pp. 1149–1158, Jul. 2018.
- [42] R. Meattini, S. Benatti, U. Scarcia, L. Benini, and C. Melchiorri, “Experimental evaluation of a sEMG-based human-robot interface for human-like grasping tasks,” in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2015, pp. 1030–1035.
- [43] M. Laghi, A. Ajoudani, M. Catalano, and A. Bicchi, “Tele-impedance with force feedback under communication time delay,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 2564–2571.
- [44] M. Laghi et al., “Shared-autonomy control for intuitive bimanual tele-manipulation,” in *Proc. IEEE-RAS 18th Int. Conf. Humanoid Robots*, 2018, pp. 1–9.
- [45] A. O. Perotto, *Anatomical Guide for the Electromyographer: The Limbs and Trunk*. Springfield, IL, USA: Charles C. Thomas, 2011.
- [46] B. Peacock, T. Westers, S. Walsh, and K. Nicholson, “Feedback and maximum voluntary contraction,” *Ergonomics*, vol. 24, no. 3, pp. 223–228, 1981.
- [47] A. Ravankar, A. Ravankar, Y. Kobayashi, Y. Hoshino, and C.-C. Peng, “Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges,” *Sensors*, vol. 18, no. 9, 2018, Art. no. 3170.
- [48] X. Li, X. Gao, W. Zhang, and L. Hao, “Smooth and collision-free trajectory generation in cluttered environments using cubic B-spline form,” *Mech. Mach. Theory*, vol. 169, 2022, Art. no. 104606.
- [49] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, MA, USA: Cambridge Univ. Press, 2004.
- [50] G. Rousseau, C. Stoica Maniu, S. Tebbani, M. Babel, and N. Martin, “Minimum-time B-spline trajectories with corridor constraints. application to cinematographic quadrotor flight plans,” *Control Eng. Pract.*, vol. 89, pp. 190–203, 2019.
- [51] B. Demeulenaere, G. Pipeleers, J. D. Caigny, J. Swevers, J. D. Schutter, and L. Vandenberghe, “Optimal splines for rigid motion systems: A convex programming framework,” *J. Mech. Des.*, vol. 131, no. 10, 2009, Art. no. 101004.
- [52] O. Khatib, “A unified approach for motion and force control of robot manipulators: The operational space formulation,” *IEEE J. Robot. Autom.*, vol. 3, no. 1, pp. 43–53, Feb. 1987.
- [53] E. R. Girden, *ANOVA: Repeated Measures*. Newbury Park, CA, USA: Sage, 1992.
- [54] A. Field, J. Miles, and Z. Field, *Discovering Statistics Using R*. Newbury Park, CA, USA: Sage, 2012.
- [55] S. Balasubramanian, A. Melendez-Calderon, A. Roby-Brami, and E. Burdet, “On the analysis of movement smoothness,” *J. Neuroeng. Rehabil.*, vol. 12, no. 1, pp. 1–11, 2015.
- [56] C. De Boor, *A Practical Guide to Splines* (ser. Applied Mathematical Sciences), revised ed. Berlin, Germany: Springer, 2001.
- [57] M. Andelic and C. M. da Fonseca, “Sufficient conditions for positive definiteness of tridiagonal matrices revisited,” *Positivity*, vol. 15, no. 1, pp. 155–159, 2010.
- [58] T. Lyche and K. Mørken, “Spline methods draft,” Dept. Informat., Center Math. Appl., Univ. Oslo, Oslo, Norway, 2008, pp. 3–8.
- [59] W. Ford, *Numerical Linear Algebra With Applications: Using Matlab*. Burlington, MA, USA: Elsevier, 2014.



Luigi Biagiotti (Member, IEEE) received the M.Sc. degree in electronics engineering and the Ph.D. degree in automation engineering from the University of Bologna, Bologna, Italy, in 2000 and 2003, respectively.

He is currently an Associate Professor with the Department of Engineering “Enzo Ferrari,” University of Modena and Reggio Emilia, Modena, Italy. He is the author or coauthor of more than 50 scientific papers presented at conferences or published in journals and two books on motion planning and automatic control.

His research interests include modeling and control of physical systems, trajectory planning and optimization, control of robotic systems, and human-robot interaction.



Roberto Meattini (Member, IEEE) received the master’s degree in automation engineering and the Ph.D. degree in biomedical, electrical, and system engineering from the University of Bologna, Bologna, Italy, in 2014 and 2018, respectively.

From 2017 to 2018, he was a Visiting Scientist with the Robotics and Mechatronics Institute, German Aerospace Center, Weßling, Germany. He is currently a Junior Assistant Professor with the Department of Electrical, Electronic and Information Engineering, University of Bologna. He is a coauthor of about 30

scientific papers presented at international conferences or published in journals. His research interests include control of robot hands, human-in-the-loop robot control, advanced human-robot interfaces for robotic grasping and prosthetics, and human intent detection for telerobotics and assistive and collaborative robotics.



Davide Chiaravalli received the M.Sc. and Ph.D. degrees in automation engineering from the University of Bologna, Bologna, Italy, in 2015 and 2018, respectively.

He is currently a Research Fellow with the University of Bologna. His research interests include robotic manipulation, haptic control, human-robot interaction, and mobile robotics.



Gianluca Palli (Senior Member, IEEE) received the Laurea and Ph.D. degrees in automation engineering from the University of Bologna, Bologna, Italy, in 2003 and 2007, respectively.

He is currently a Full Professor with the University of Bologna. He has authored or coauthored more than 150 scientific articles presented at conferences or published in journals. His research interests include design and control of manipulation devices, human-robot physical interaction, mobile manipulation, manipulation of deformable objects, and soft robotics.



Claudio Melchiorri (Fellow, IEEE) received the bachelor's and Ph.D. degrees in electronic engineering from the University of Bologna, Bologna, Italy, in 1985 and 1990, respectively.

In 1985, he started his research activity with the Department of Electrical, Electronic and Information Engineering (DEI), University of Bologna, where he became a Research Associate in 1990. Since 2001, he has been a Full Professor in Robotics with DEI, University of Bologna. Since 2014, he has been a Member of the Academy of Sciences, Bologna Institute. He has participated to and coordinated several research projects in robotics, automatic control, and automation, both at the national and international levels. He is the author or coauthor of about 300 scientific papers presented at conferences or published in journals and six books on digital control, and he is a coeditor of seven books on robotics. His research interests include robotics (dexterous manipulation, telemanipulation, haptic systems, and advanced sensors), control theory (nonlinear control, passivity, and robust control), and motion control (electric motors, trajectory planning, and hardware/software real-time control systems).

Open Access funding provided by 'Università degli Studi di Modena e Reggio Emilia' within the CRUI CARE Agreement