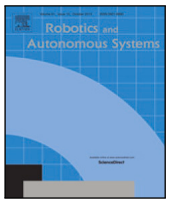
Robotics and Autonomous Systems 194 (2025) 105120

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot



Phase-independent Dynamic Movement Primitives with applications to human–robot co-manipulation and time optimal planning

Giovanni Braglia *, Davide Tebaldi, Luigi Biagiotti

University of Modena and Reggio Emilia, Department of Engineering Enzo Ferrari, Via Pietro Vivarelli 10, Modena, 41125, Italy

ARTICLE INFO

Keywords:
Physical human–robot interaction
Motion and path planning
Collaborative robotics
Co-manipulation
Programming by Demonstration
Dynamic Movement Primitives

ABSTRACT

Dynamic Movement Primitives (DMP) are an established and efficient method for encoding robotic tasks that require adaptation based on reference motions. Typically, the nominal trajectory is obtained through Programming by Demonstration (PbD), where the robot learns a task via kinesthetic guidance and reproduces it in terms of both geometric path and timing law. Modifying the duration of the execution in standard DMPs is achieved by adjusting a time constant in the model.

This paper introduces a novel approach to fully decouple the geometric information of a task from its temporal information using an algorithm called spatial sampling, which allows parameterizing the demonstrated curve by its arc-length. This motivates the use of the name Geometric DMP (GDMP) for the proposed DMP approach. The proposed spatial sampling algorithm guarantees the regularity of the demonstrated curve and ensures a consistent projection of the human force throughout the task in a human-in-the-loop scenario. GDMP exhibits phase independence, as its phase variable is no longer constrained to the demonstration's timing law, enabling a wide range of applications, including phase optimization problems and human-in-the-loop applications. Firstly, a minimum task duration optimization problem subject to velocity and acceleration constraints is formulated. The decoupling of path and speed in GDMP allows to achieve optimal time duration without violating the constraints. Secondly, GDMP is validated in a human-in-the-loop application, providing a theoretical passivity analysis and an experimental stability evaluation in co-manipulation tasks. Finally, GDMP is compared with other DMP architectures available in the literature, both for the phase optimization problem and experimentally with reference to an insertion task and a simulated welding task, showcasing the enhanced performance of GDMP with respect to other solutions.

1. Introduction

In recent years, the interest in robotics applications has been rising significantly [1,2]. This includes collaborative robotics applications, where robots work alongside humans rather than performing tasks independently. This scenario is typically referred to as Human–Robot Interaction (HRI), and a lot of research interest has been directed towards the development of novel algorithms and techniques to enhance the human–robot communication, coexistence and collaboration in various settings [3–6]. An important aspect of HRI applications is represented by their accessibility to non-expert users [7,8], where techniques such as Imitation Learning [9], or Programming by Demonstration (PbD) [8], come into play. These methods involve a demonstration phase and rely on encoding, planning and control strategies to correctly reproduce the desired task [10,11]. One of the most widespread encoding/planning techniques available in the literature are Dynamic Movement Primitives (DMP) [12], which were first introduced in [13] as a way to encode motor movement with convenient stability properties. DMP consist of a second-order system with the addition of a forcing term [14] and can be classified into rhythmic or discrete DMP, depending on whether the attractor system is a limit cycle or a point [12], respectively. Discrete DMP are typically formulated as follows [12,14]:

$$\tau \dot{z}(t) = \alpha[\beta(g - y(t)) - z(t)] + f(s(t)), \tag{1}$$

$$\tau \dot{y}(t) = z(t), \tag{2}$$

$$\tau \dot{s}(t) = -\delta s(t), \tag{3}$$

where Eqs. (1)–(2) are referred to as the Transformation System (TS), whereas (3) is referred to as the Canonical System (CS). The parameter τ is a time constant, α and β are system parameters chosen to have a critically damped TS (i.e. $\beta = \alpha/4$), g is the target goal and the forcing term $f(s(t))$ shapes the generated trajectory $y(t)$ through the

* Corresponding author.
E-mail addresses: giovanni.braglia@unimore.it (G. Braglia), davide.tebaldi@unimore.it (D. Tebaldi), luigi.biagiotti@unimore.it (L. Biagiotti).

https://doi.org/10.1016/j.robot.2025.105120
Received 30 December 2024; Received in revised form 6 May 2025; Accepted 27 June 2025
Available online 18 July 2025

TS dynamics. The TS can be rewritten as

$$\tau^2 \ddot{y}(t) + \alpha\tau \dot{y}(t) + \alpha\beta(y(t) - g) = f(s(t)), \tag{4}$$

representing a globally stable second-order system characterized by the unique equilibrium point $(y(t), \dot{y}(t)) = (g, 0)$ when $f(s(t)) = 0$. The CS (3) is a first-order system representing the evolution of the phase variable $s(t)$, with δ being a positive constant [14]. The forcing term $f(s(t))$ in (1) is typically expressed as the linear combination of radial basis functions [13]:

$$f(s(t)) = \sum_{i=1}^{N} \omega_i \phi_i(s(t)) \cdot s(t) \cdot (g - y_0), \tag{5}$$

where

$$\phi_i(s(t)) = \frac{b_i(s(t))}{\sum_{j=1}^{N} b_j(s(t))}, \quad b_i(s(t)) = \exp\left(\frac{-(s(t) - c_i)^2}{2h}\right). \tag{6}$$

Note that, since $s(t) \to 0$ when $t \to \infty$ from (3), $f(s(t)) \to 0$ from (5), therefore its impact in the TS (1)–(2) is limited in time. The constant $(g - y_0)$ in (5) is a modulation term enabling the TS (1)–(2) to be an orientation-preserving homeomorphism with respect to changes in the initial and goal positions y_0 and g [13,15]. The weight coefficients ω_i in (5) are calculated by Locally Weighted Regression (LWR) [16] as follows:

$$\omega_i = \underset{\omega_i}{\operatorname{argmin}} \sum_{t=0}^{T_f} \phi_i(s(t))[f_r(t) - \omega_i \xi(t)]^2, \tag{7}$$

where $\xi(t) = s(t)(g - y_0)$ and the nominal forcing term $f_r(t)$ is computed from (4) by using the reference position, velocity, and acceleration trajectories $y_r(t), \dot{y}_r(t)$ and $\ddot{y}_r(t)$:

$$f_r(t) = \tau^2 \ddot{y}_r(t) + \alpha\tau \dot{y}_r(t) + \alpha\beta(y_r(t) - g_r), \tag{8}$$

where $g_r = y_r(T_f)$. The parameter T_f is the total duration of the reference trajectory $y_r(t)$, and the time constant τ is commonly assumed to be unitary.

DMP represent a simple yet effective encoding technique, featuring useful properties such as stability, time scaling, spatial modulation, and compatibility with additional coupling terms [12,13,17]. Thanks to their compact and versatile formulation, DMP were successfully applied to various tasks, such as tennis swing [14], bi-manual manipulation [18], peg-in-hole [19], among others [4,20–22]. Despite the several interesting researches on DMP, we identified some research branches which, to the best of our knowledge, have not been fully investigated yet, as discussed in Section 3.

The remainder of this manuscript is structured as follows. An overview of the state of the art and of the issues when dealing with DMP is given in Section 2, whereas the contributions of this work are discussed in Section 3. The adopted DMP formulation is discussed in Section 4 together with the introduction of the proposed Spatial Sampling algorithm, representing the core of the proposed Geometric DMP (GDMP) formulation. Different applications of the GDMP are then proposed in the following sections. Specifically, the phase optimization problem is discussed in Section 5 together with the comparison with other DMP approaches, while the human-in-the-loop application is discussed in Section 6 together with the passivity analysis. The experimental stability evaluation in co-manipulation tasks is performed in Section 7, together with the experimental comparison of GDMP with other DMP approaches. The obtained results are discussed in Section 8, while the conclusions are given in Section 9.

2. Related works

The state trajectories of a dynamic physical system naturally evolve with time. However, if the considered dynamic model does not represent a physical system, the system trajectories may be desired to evolve according to a different timing law. For instance, scaling the original time by a suitable factor can enhance flexibility, offering advantageous properties like feedback linearizability [23,24].

In the specific case of DMP, the timing law is typically governed by the Canonical System (CS) in (3), where the variation of the phase variable $s(t)$ can be modulated by properly tuning the parameter τ, thus giving a time-scalable Transformation System (TS) in (1)–(2) and maintaining the topological equivalence of the output trajectories [13,19,22]. Choosing the CS in (3) to exhibit an exponential decay ensures that the contribute of $f(s(t))$ in (1) fades out when reaching the goal point g [12]. However, this solution inevitably leads to very high values of the weights in (7) when close to the final part of the reference trajectory [25], which can be undesirable when combining multiple DMP together [4,17,21]. To cope with this issue, other types of CS were adopted, such as piece-wise linear [25] or sigmoidal-like phase profiles [26].

The key aspect of having a timing law governed by the CS in (3) is that the evolution of the trajectory $y(t)$ in the TS (1)–(2) can be effectively slowed down or sped up by tuning the parameter τ, without introducing any geometric variation in the resulting trajectory $y(t)$ of the DMP [13,15]. In this case, an issue arises when the trajectory is required to stop during its reproduction, as this would imply $\tau \to \infty$. This problem was noted in [14,27], then further analyzed in [20]. In the latter reference, the scaling factor τ relies on the phase $s(t)$ and is expressed as a combination of basis functions, aiming at acquiring an optimized velocity profile for a bimanual kitchen task.

Many different formulations of the CS in (3) can be found in the literature, in order to adopt proper timing laws for the considered applications. In [28], a coupling term is used to modulate the task velocity based on the measured contact force with the environment. The task velocity can also be changed as in [27], where non-uniform velocity changes are allowed to move along the trajectory according to the forces and torques applied to the robot. In [29], Iterative Learning Control (ILC) is employed to learn an additional and optimal phase-dependent temporal scaling factor into the DMP equations, whereas an online optimization of the time duration of the task is performed in [22] in order to vary the task velocity, and thus the robot velocity, as a function of the distance from the target. Other works where the timing law is modified according to external perturbation on the planned trajectory can be found in [14,19,30–32].

A key application in trajectory planning is minimizing task duration. Shaping the timing law $s(t)$ along a predefined path for tracking is commonly known as time-optimal path parameterization (TOPP). In TOPP problems, the objective is to minimize the execution time T along a constrained curve, subject to first and second order constraints [33]. TOPP relies on the path-speed separation principle, where the constraint curve serves as a geometric constraint to which any speed profile can be applied. Robotic applications often impose constraints on the minimum allowable time duration T to respect the robot's kinematic limitations. In [34], this problem is transformed into a convex optimal control problem with a single variable. Alternatively, [35] solves the TOPP using factor graph variable elimination, achieving the global optimum for minimum time problems with quadratic objectives.

In the DMP literature, the minimum time problem has been indirectly addressed due to temporal scaling being a well-known property of DMPs. For example, scaling τ in the canonical system of (3) can reduce time duration, but it does not enforce kinematic constraints, such as velocity or acceleration limits. This limitation was acknowledged in [36,37], where the authors proposed a novel DMP formulation that scales τ offline and online to comply with the kinematic constraints. Similarly, in [38], the problem of respecting kinematic constraints was addressed by appropriately computing the weights ω_i in (10). While these approaches effectively scale robot velocity, they cannot be classified as TOPP methods because (i) position trajectories are used instead of geometric paths, and (ii) their objective is not to minimize time duration. To the best of our knowledge, the minimum time problem has not yet been directly addressed in the DMP literature.

Another important aspect when dealing with timing laws is the possibility of letting the trajectory move backward and forward between the initial and goal positions, y_0 and g respectively, while preserving the trajectory topology, i.e. while allowing $\dot{s}(t)$ to change sign [19,34]. This feature is called phase reversibility. Even though it offers many benefits in various scenarios [27], the classical DMP formulation in (1)–(3) does not fulfill the phase reversibility requirement [29]. In fact, the restriction of having a strictly monotonic $\dot{s}(t)$ implies that the new timing law neither allows the trajectory to be reproduced backward nor allows it to be stopped [24]. The problem of reversibility was considered in [39] and further investigated in [19,40], where it was shown that having a reversible phase can affect the DMP global stability property [27]. In [40], the authors formulate the design requirements for guaranteeing phase reversibility in DMP, which is a desirable property in some DMP formulation [19,22,38,41,42]. Firstly, a reversible DMP should maintain global stability properties as the original one [40], meaning that y_0 must be an attractor point when moving backward, whereas g must be an attractor point when moving forward. Secondly, the trajectory topology must be preserved when moving forward and backward. Finally, it would be desirable to adopt the same forcing term when progressing along the intended path, rather than switching between two different functions depending on the motion direction.

Whenever a new DMP formulation is proposed, it is also important to verify that global asymptotic stability is guaranteed. In [40], a candidate for such a system is given by the Logistic Differential Equation (LDE) $\tau\dot{y} = \alpha(y - y_0)(g - y)$. However, it can be demonstrated that the region of global stability of the LDE is restricted to $y \in (y_0, g)$ with $y_0 < g$ [43]. Consequently, two separate forcing terms are learned to switch along the desired curve whenever the sign of $\dot{s}$ changes [44].

When dealing with phase reversibility, a considerable step forward was provided in [19]. In this work, the authors present a new formulation of the original DMP satisfying all phase reversibility requirements by using a unique forcing term $f(s(t))$, based on a parameterization of the reference trajectory $y_r(t)$ and of its time-derivatives [34,45]. In this case, there is no need to limit the forcing term $f(s(t))$ by multiplying it by $s(t)$, as the terms $y_r(t), \dot{y}_r(t), \ddot{y}_r(t)$ are bounded by construction and the derivatives tend to zero when the system approaches the initial/goal positions [38]. Additionally, backward reproduction is achieved without impacting the DMP stability by simply reversing the sign of the phase velocity $\dot{s}(t)$ [22].

A potential application for reversible DMP is its use as a virtual fixture [46–48]. The latter represents a predefined path constraining the robot movement without a specified timing law. The phase variable $s(t)$ can be adjusted arbitrarily, allowing the system to evolve along the predefined paths. However, the DMP architecture imposes some limitations as the recorded reference trajectories are affected by the time dependence observed during the task demonstrations [8]. The same issue emerges when multiple demonstrations of the same task are recorded with different timing laws. In this case, techniques such as Dynamic Time Warping (DWT) [49] allow to align the different demonstrations and compute a unique reference motion, such that the related skill is extracted [7,10].

In this paper, we address the issues detailed in this section by proposing the concept of Geometric DMP (GDMP). This approach is based on the introduction of a Spatial Sampling algorithm, which enables a geometric interpretation of the phase variable $s(t)$. Unlike the standard case, this new phase variable is no longer constrained by the timing law of the demonstration and can be appropriately chosen based on the specific application. This characteristic justifies the alternative name we have given to this method: Phase-Independent DMP.

3. Methodology and contributions

Despite the extensive literature on DMP, to the best of our knowledge, there is still no unified framework that allows for the arbitrary selection of the Canonical System in (3). In this work, we aim at achieving this objective starting from the DMP formulation originally proposed in [19]. The new contributions of our work are: (i) The proposal of a new Spatial Sampling algorithm. This algorithm allows the conversion of the demonstrated trajectory, which is constantly time-sampled, into a new one which is constantly spatially-sampled. The demonstrated trajectory can also include parts with zero speed. The spatially-sampled trajectory no longer exhibits the time dependence of the original one, and is expressed as a function of its arc-length parameter. For this reason, it is a *regular curve*, i.e., a curve whose first derivative is always nonzero. (ii) The use of the spatial sampling algorithm allows to decouple the phase variable from the timing law during the demonstration phase. Therefore, the term "phase-independent" DMP is introduced, referring to the fact that the phase variable can now be freely chosen depending on the application and is no longer constrained by the Canonical System (3). Time scaling and reversibility are direct consequences of this feature. Since the forcing term $f(s(t))$ is computed solely based on the geometrical information contained in the demonstrated trajectory, the proposed methods is also called Geometric DMP (GDMP). The framework of GDMP is shown in Fig. 1. The Spatial Sampling algorithm is first applied to the demonstrated trajectory. Next, the forcing term $f(s(t))$ is computed, and employed in the transformation system to generate the output trajectory. (iii)Two significant applications of the GDMP are presented: (1) the offline computation of the phase variable that optimizes a given cost function, such as total execution time, subject to velocity and acceleration constraints; (2) a human–robot co-manipulation task, where the robot constrains the motion along the path encoded by the GDMP, while the human determines how the motion unfolds. The two aforementioned applications are highlighted on the right of Fig. 1, together with the first option consisting in reproducing the original demonstrated trajectory. Furthermore, as a feedback system with the human in the loop, a passivity analysis and an experimental evaluation of practical stability are provided. Both applications are compared with several approaches from the DMP literature, demonstrating the enhanced performance of GDMP.

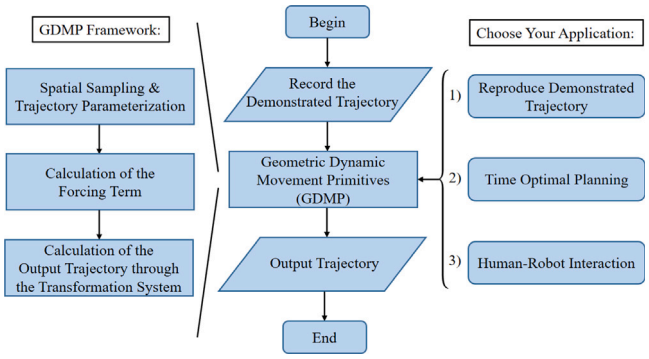


Fig. 1. Flowchart of the proposed GDMP framework.

4. Geometric DMP formulation

This section deals with the description of the proposed Geometric DMP, which employs the formulation in [19] as a starting point. The forcing term $f(s(t))$ in (4) is computed as follows:

$$f(s(t)) = \eta f^{\star}(s(t)), \quad \text{where} \quad \eta = \frac{g - y(s(0))}{g_r - y_r(0)} \tag{9}$$

is a scaling factor depending on the initial and goal positions $y_r(0)$ and $g_r = y_r(s_f)$ of the reference trajectory, as well as on the initial and desired goal positions $y(s(0))$ and g. The objective of the scaling factor η is: (1) to preserve the topology of the curve [15,22,41]; (2) to run the system passivity analysis as further discussed in Section 6.1.

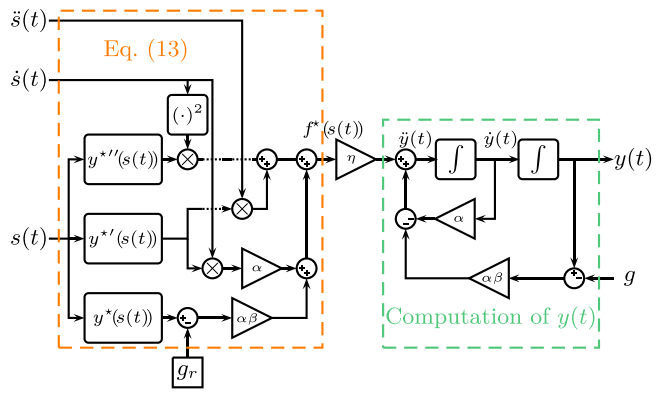


Fig. 2. Block-scheme representation of the TS (4) by incorporating the forcing term (13), which ensures the independence from the phase variable $s(t)$.

In the classical DMP, the forcing term $f^{\star}(s)$ is obtained as in (5), (6) and (7), where the nominal forcing term $f_r(t)$ is computed as a linear combination of the demonstrated position, velocity, and acceleration profiles as in (8). In our case, the forcing term $f^{\star}(s)$ in (9) is obtained using the linear combination (8) by replacing the demonstrated position, velocity, and acceleration profiles with the corresponding parameterizations $y^{\star}(s)$, $\dot{y}^{\star}(s)$ and $\ddot{y}^{\star}(s)$. Specifically, given the sequence of demonstrated positions $y_r(s)$, $s = 0, \dots, s_f$, the coefficients ω_i of the parametric function

$$y^{\star}(s) = \sum_{i=1}^{N} \omega_i \phi_i(s) \tag{10}$$

are determined by minimizing the cost function

$$J = \sum_{t=0}^{T_f} \left\| y_r(s) - \omega_i \phi_i(s) \right\|^2 . \tag{11}$$

The radial basis functions $\phi_i(\cdot)$ are defined in (6). In [19], the independent variable is the time of the demonstrated trajectory, i.e. $s \equiv t$. The function $y^{\star}(s)$ is then reparameterized by composing it with the desired timing law $s = s(t) : [0, T_f] \to [0, s_f]$. Note that $s(t)$ must be a monotonic C^1-class function such that $\dot{s}(0) = \dot{s}(T_f) = 0$ and $\ddot{s}(0) = \ddot{s}(T_f) = 0$. The derivatives $\dot{y}^{\star}(s(t))$ and $\ddot{y}^{\star}(s(t))$ of the parameterized trajectory $y^{\star}(s(t))$ can be deduced as follows:

$$\begin{aligned} \dot{y}^{\star}(s(t)) &= \frac{\partial y^{\star}(s(t))}{\partial s(t)} \frac{ds(t)}{dt} = y^{\star\prime}(s(t))\dot{s}(t), \\ \ddot{y}^{\star}(s(t)) &= \frac{d\dot{y}^{\star}(s(t))}{dt} = y^{\star\prime\prime}(s(t))\dot{s}^2(t) + y^{\star\prime}(s(t))\ddot{s}(t), \end{aligned} \tag{12}$$

where the prime and dot notations denote the differentiation with respect to the phase variable $s(t)$ and with respect to time, respectively. Therefore, with the proposed formulation, it is not necessary to know the derivatives of the reference trajectory, but only the position profile $y_r(t)$ is sufficient. Using (9), (10) and (12), the forcing term $f(s(t))$ in (4) can be written as:

$$\begin{aligned} f(s(t)) = \eta \Big[& y^{\star\prime\prime}(s(t))\dot{s}^2(t) + y^{\star\prime}(s(t))\ddot{s}(t) \\ & + \alpha y^{\star\prime}(s(t))\dot{s}(t) + \alpha\beta\big(y^{\star}(s(t)) - g_r\big) \Big]. \end{aligned} \tag{13}$$

By distributing the scaling factor η across all the terms on the right side of (13), $f(s(t))$ can be expressed as a linear combination of

$$y^{\star}_{\eta}(s(t)) = \eta\, y^{\star}(s(t)) \tag{14}$$

and its derivatives. The complete schematic view of the TS (1)–(2) with the forcing term (13) is reported in Fig. 2, and enables the following features [19,38]: time modulation by varying the evolution of the phase variable $s(t)$; phase reversibility allowing forward and backward reproduction of the trajectory, when $\dot{s}(t) > 0$ and $\dot{s}(t) < 0$, respectively; spatial scaling of the curve varying the initial position $y(s(0))$ and the goal g of the task.

The TS can then be generalized to the multidimensional case. If the considered application requires the robot position only to be considered, as in Section 5, Section 6 and Section 7, a Cartesian representation of the TS can be employed:

$$\ddot{\boldsymbol{y}}(t) + \boldsymbol{A}\dot{\boldsymbol{y}}(t) + \boldsymbol{AB}\big(\boldsymbol{y}(t) - \boldsymbol{g}\big) = \boldsymbol{E}\boldsymbol{f}^{\star}(s(t)), \tag{15}$$

where $\boldsymbol{A} = \mathrm{diag}\,(\alpha_x, \alpha_y, \alpha_z)$, $\boldsymbol{B} = \mathrm{diag}\,(\beta_x, \beta_y, \beta_z)$, $\boldsymbol{E} = \mathrm{diag}\,(\eta_x, \eta_y, \eta_z)$, and

$$\boldsymbol{y}(t) = \begin{bmatrix} y_x(t) \\ y_y(t) \\ y_z(t) \end{bmatrix}, \ \boldsymbol{g} = \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix}, \ \boldsymbol{f}^{\star}(s) = \begin{bmatrix} f^{\star}_x(s(t)) \\ f^{\star}_y(s(t)) \\ f^{\star}_z(s(t)) \end{bmatrix}. \tag{16}$$

Note that Eqs. (15)–(16)can also be generalized to account for the end-effector orientation by adopting a minimal representation, such as Euler angles or unit quaternions [13,41,50]. The resulting TS in Fig. 2 still maintains the characteristic properties of the original formulation, such as spatial adaptation with respect to different goal/initial position and with respect to the presence of obstacles by adding proper coupling terms [13]. Fig. 3 shows an example of application of such properties: Fig. 3a and Fig. 3b show the possibility of changing the goals and initial positions g and $y(s(0))$ in (9) with respect to the reference ones g and $y_r(0)$, while Fig. 3c shows how the introduction of coupling terms, in this case repulsive potential fields [38,51], allows to avoid obstacles along the path.

4.1. Regularity of the curve

The Canonical System (CS) in (3) can be set in order to ensure that the output of the DMP accurately reproduces the reference trajectory, including its time dependency [14]. This means that, if the user stops while recording the reference trajectory, i.e. $\frac{d\boldsymbol{y}_r(s)}{ds} = 0$, the corresponding parameterized trajectory will result in $\boldsymbol{y}^{\star\prime}(s) = 0$. In all the formulation of DMP proposed in the literature, the duration of this pausing phase can be shortened or extended by adjusting the parameter τ in (1)–(3), but cannot be removed [47]. This side effect persists also in the formulation of Fig. 2 proposed in [19], since $\dot{\boldsymbol{y}}_r(t) = 0$ implies $\dot{\boldsymbol{y}}^{\star}(s(t)) = \ddot{\boldsymbol{y}}^{\star}(s(t)) = 0$. This issue becomes particularly critical in applications where only the geometric aspects of the demonstrated path are required. For instance, in [19], the reference trajectory is captured in two distinct steps: one for the path and another for the velocity profiles. Similarly, in [27], a dynamic scaling factor is employed, modulated by the tangential component of an external force.

The aforementioned problem can be formally identified as the lack of regularity in the curve. By definition [52], a curve $\hat{\boldsymbol{y}}_r(s) : \mathbb{R} \to \mathbb{R}^3$ is regular iff:

$$\frac{d\hat{\boldsymbol{y}}_r(s)}{ds} \neq 0 \qquad \forall s \in \mathbb{R}. \tag{17}$$

From (17), it follows that a regular curve $\hat{\boldsymbol{y}}_r(s)$ and its tangential direction $\frac{d\hat{\boldsymbol{y}}_r(s)}{ds}$ are always well-defined at each point of the space.

4.2. Spatial sampling algorithm

To address the aforementioned issues, we propose re-sampling the recorded trajectory $\boldsymbol{y}_r(t)$ using the proposed "Spatial Sampling" algorithm as a preliminary step to the parameterization (10). The proposed algorithm generates a filtered trajectory $\boldsymbol{y}_{\Delta}(\boldsymbol{s}_k)$ which is a function of the arc-length parameter $\boldsymbol{s}_k$. Other approaches have been proposed in the literature to address the separation of the spatial and temporal components in the demonstrated trajectory, such as the arc-length (AL) DMP proposed in [53]. However, the ALDMP proposed in [53] cannot handle situations where the user stops while recording the demonstrated trajectory, namely it cannot handle situations where $\dot{\boldsymbol{y}}_r(t) = 0$ without introducing a trajectory segmentation. The proposed Spatial Sampling algorithm, detailed in Algorithm 1, works as follows:

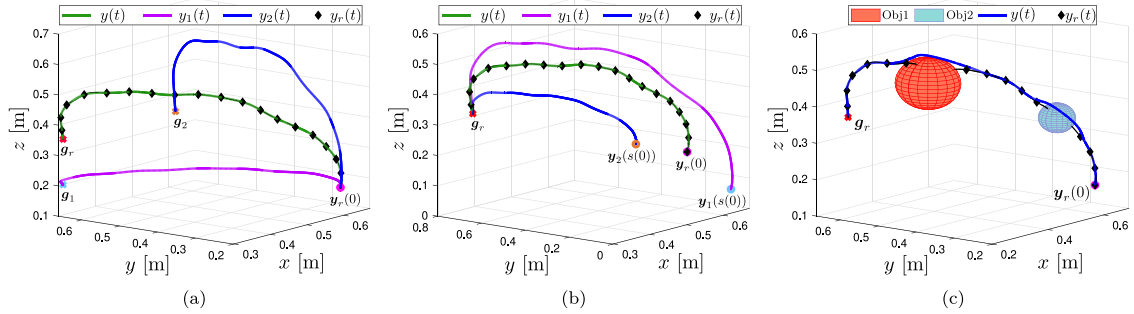


Fig. 3. Verification that the DMP formulation in Fig. 2 maintains the characteristic properties of the original one: spatial adaptation with respect to (a) different target goals, (b) different initial positions and (c) presence of obstacles.

(1) Starting from the sequence of samples obtained by recording the demonstrated trajectory $\boldsymbol{y}_r(t)$ with constant period T, $\boldsymbol{y}_{T,k} = \boldsymbol{y}_r(t_k) = \boldsymbol{y}_r(kT)$ for $k = 0, \dots, N$, a linearly interpolating, continuous-time function $\boldsymbol{y}_L(t)$ is built by applying a First-Order Hold (FOH) on it. If the sampling period T is small enough, then the following holds true:

$$\boldsymbol{y}_L(t) \approx \boldsymbol{y}_r(t). \tag{18}$$

(2) A new sequence $\boldsymbol{y}_{\Delta,k}$ is obtained by imposing that $\boldsymbol{y}_{\Delta,0} = \boldsymbol{y}_L(0)$ and, for $k > 0$, $\boldsymbol{y}_{\Delta,k} = \boldsymbol{y}_L(t_{\Delta_k})$, where t_{Δ_k} is the time value that guarantees

$$\|\boldsymbol{y}_{\Delta,k} - \boldsymbol{y}_{\Delta,k-1}\| = \Delta, \quad \text{for} \quad k = 1, \dots, M. \tag{19}$$

The parameter Δ defines the geometric distance between consecutive samples and can be freely chosen. If the reference trajectory $\boldsymbol{y}_r(t)$ also includes the orientation components, the enforcement of condition (19) is restricted to the position components, while the resulting time instants t_{Δ_k} will then be used to generate the new sequence $\boldsymbol{y}_{\Delta,k}$ including both position and orientation components. The condition (19) implies that the total distance between the first point $\boldsymbol{y}_{\Delta,0}$ and the generic kth point $\boldsymbol{y}_{\Delta,k}$, $k > 0$, is given by $k\Delta$. This distance approximates the length of the curve $\boldsymbol{y}_L(t)$ at the time instant t_{Δ_k}, with a precision that depends on the value of Δ. Therefore, for Δ small enough, the new sampling mechanism induces a mapping between the length $s_k = k\Delta$ and the position along the approximating linear curve $\boldsymbol{y}_L$, i.e.

$$\boldsymbol{y}_{\Delta,k} = \boldsymbol{y}_L(t_{\Delta_k}), \quad \text{with} \quad t_{\Delta_k} = \gamma^{-1}(s_k), \tag{20}$$

where $s(t) = \gamma(t)$ is the particular timing law imposed during the trajectory demonstration, describing how the robot moves along the imposed geometric path being variable s the arc-length parameterization of the curve. From (18), (19) and (20), the resulting trajectory parameterized with respect to the arc-length parameter $\hat{\boldsymbol{y}}_r(s) \approx \hat{\boldsymbol{y}}_L(s) = \boldsymbol{y}_L(\gamma^{-1}(s))$ can be obtained. The sequence $\boldsymbol{y}_{\Delta,k}$ can be viewed as the result of a sampling operation with a constant spatial period Δ, characterized by:

$$\left\| \frac{d\hat{\boldsymbol{y}}_r(s)}{ds} \right\|_{s=s_k} \approx \frac{\|\boldsymbol{y}_{\Delta,k+1} - \boldsymbol{y}_{\Delta,k}\|}{\|s_{k+1} - s_k\|} = \frac{\Delta}{\Delta} = 1. \tag{21}$$

Remark 1. The Spatial Sampling algorithm generates a filtered curve $\boldsymbol{y}_r(s)$ satisfying condition (21). From the latter, it follows that the filtered curve $\boldsymbol{y}_r(s)$ always satisfies the regularity condition (17), enabling the parameterization (12) to be always well-defined. Furthermore, condition (21) allows to achieve a consistent projection of the human force $\boldsymbol{F}_h$ throughout the task in human-in-the-loop applications, as further emphasized in the experimental results of Section 7.3.

The pseudo-code of the Spatial Sampling algorithm is shown in Algorithm 1[1], while the operation of the algorithm is graphically depicted in Fig. 4 for a one-dimensional case study. Initially, by sampling the reference trajectory $y_L(t_T)$ at a constant time interval T, the linear interpolation function $y_L(t)$ is derived in Fig. 4a. Note that the geometric distance of the samples $y_L(t_T)$ is clearly not constant. Subsequently, in Fig. 4b and Fig. 4c, the spatial sampling algorithm is applied to the curve $y_L(t)$ using two different spatial intervals Δ, where it can be noticed that smaller values of Δ result in a more accurate approximation of the original curve. This observation suggests considering the value of Δ as small as possible to maintain a meaningful representation of the demonstrated trajectory. However, since Δ represents the amplitude of the minimum movement that can be captured by the curve, in applications involving humans – typically characterized by unwanted motions, such as tremors, superimposed on the desired demonstration – the choice of Δ can be used to filter out these undesired contributions. This can be achieved by setting the value of Δ larger than the magnitude of the maximum disturbance affecting the demonstrated trajectory. Note that this threshold can be easily estimated by asking the user to keep the robot's end-effector in its initial location for a given amount of time and measuring the maximum apparent displacement.

In Fig. 4d, the samples of the timing law $s = \gamma(t)$ imposed during the demonstration of the trajectory and obtained with the same spatial period Δ of case (c) are also presented. These illustrate how the length along the geometric path changes with time throughout the demonstration. It is worth noting that, because the spatial sampling operation, the function $\gamma(t)$ remains strictly monotonically increasing (and thus invertible), despite the original curve being constant over certain intervals. On the contrary, the ALDMP method presented in [53]

Algorithm 1 Spatial Sampling

Input: $\boldsymbol{Y}_T \in \mathbb{R}^{(N\times d)}$, $\boldsymbol{t}_T \in \mathbb{R}^{(N\times 1)}$, Δ
Output: $\boldsymbol{Y}_\Delta \in \mathbb{R}^{(M\times d)}$, $\boldsymbol{s}_\Delta \in \mathbb{R}^{(M\times 1)}$, $\boldsymbol{t}_\Delta \in \mathbb{R}^{(M\times 1)}$
Initialization:
Initialize arrays: $\boldsymbol{Y}_{\Delta,0} = \boldsymbol{Y}_{T,0}$, $\boldsymbol{t}_{\Delta,0} = \boldsymbol{0}$, $\boldsymbol{s}_{\Delta,0} = 0$
Initialize current state: $\boldsymbol{y}_c = \boldsymbol{Y}_{T,0}$, $i = 0$
Compute $\boldsymbol{Y}_\Delta$, $\boldsymbol{t}_\Delta$, $\boldsymbol{s}_\Delta$:
while $i < N$ **do**
  if $||\boldsymbol{Y}_{T,i+1} - \boldsymbol{y}_c|| > \Delta$ **then** {}
    Compute $\boldsymbol{Y}_{\Delta,k}$ belonging to the segment $\overline{\boldsymbol{Y}_T(i)\boldsymbol{Y}_T(i+1)}$ and such that $||\boldsymbol{Y}_{\Delta,k} - \boldsymbol{y}_c|| = \Delta$
    Compute corresponding $\boldsymbol{t}_{\Delta,k}$ as in (20)
    Update arrays: $\boldsymbol{Y}_\Delta = [\boldsymbol{Y}_\Delta, \boldsymbol{Y}_{\Delta,k}]$, $\boldsymbol{s}_\Delta = [\boldsymbol{s}_\Delta, \boldsymbol{s}_{\Delta,end} + \Delta]$, $\boldsymbol{t}_\Delta = [\boldsymbol{t}_\Delta, \boldsymbol{t}_{\Delta,k}]$
    Update current state: $\boldsymbol{y}_c = \boldsymbol{Y}_{\Delta,end}$
  else
    $i = i + 1$
  end if
end while
return $\boldsymbol{Y}_\Delta$, $\boldsymbol{s}_\Delta$, $\boldsymbol{t}_\Delta$

[1] All codes can be found at the online repository: https://github.com/AutoLabModena/Geometric-Dynamic-Movement-Primitives.git

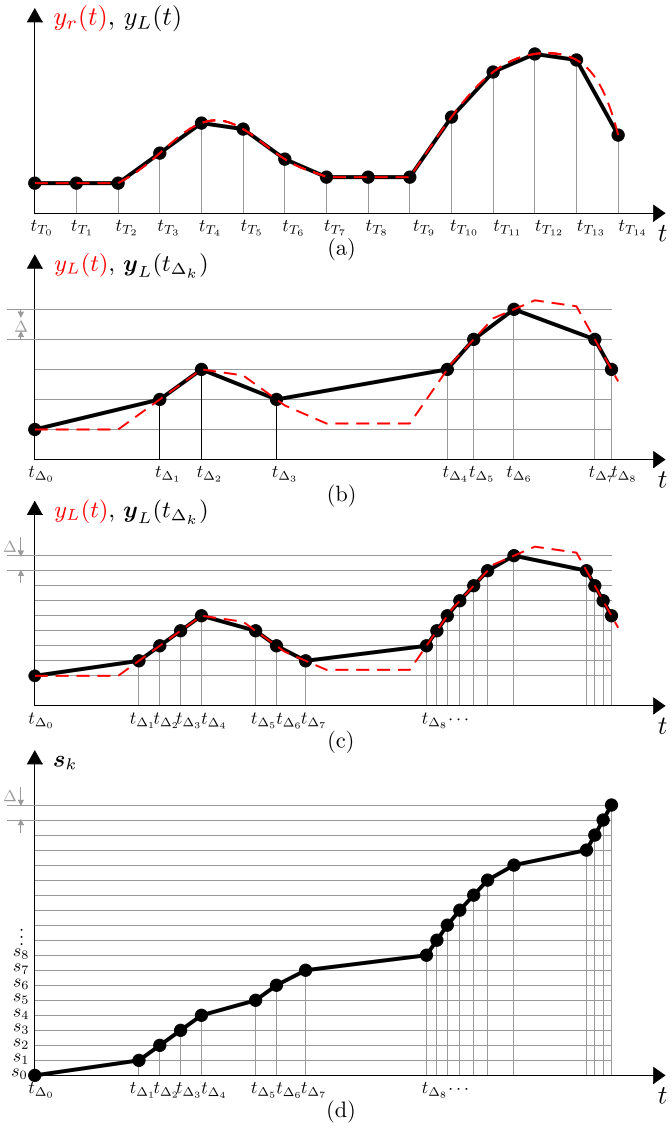


Fig. 4. Sketch displaying the operation of the Spatial Sampling algorithm in a one-dimensional scenario: reference trajectory $y_L(t_T)$ sampled with constant period T and interpolating linear curve $y_L(t)$ (a), sequence of spatially sampled points $\boldsymbol{y}_{\Delta,k} = \boldsymbol{y}_L(t_{\Delta_k})$ with $\Delta = \bar{\Delta}$ (b) and $\Delta = \bar{\Delta}/2$ (c), and samples of the function $s_k = \gamma(t_{\Delta_k})$ for $\Delta = \bar{\Delta}/2$ (d).

introduces the following operations to reparameterize the curve:

$$\{t_k, \boldsymbol{y}_{T,k}\}_{k=1}^{N}, \quad \Leftrightarrow \quad \{\sigma^{-1}(s_k), \boldsymbol{y}_{T,k}\}_{k=1}^{N}, \tag{22}$$

where $s_k = \sigma(t_k) = \int_0^{t_k} \|\dot{\boldsymbol{y}}_r(t)\| dt$. However, the equivalence in (22) is only true if $t_k = \sigma^{-1}(s_k)$ exists, that is if function $s = \sigma(t)$ is injective. Differently from $\gamma(t)$, function $s = \sigma(t)$ is injective iff $\dot{\boldsymbol{y}}_r(t) \neq 0$, a condition that cannot be guaranteed in the ALDMP method described in [53] when motion stops occur.

Remark 2. The diagram depicted in Fig. 2, where the parametric function $\boldsymbol{y}^\star(s)$ is computed based on the pair $(\boldsymbol{s}_k, \boldsymbol{y}_{\Delta,k})$, for $k = 0, \dots, M$, defines the concept of *Geometric* DMP (GDMP). From (12) and (21), it follows that this type of DMP solely derives from the geometric path of the demonstrated trajectory and can be linked to any phase variable $s(t)$. The guarantee of the curve regularity as well as the concept of having an independent phase variable – namely no longer constrained to a canonical system – represent therefore important novel aspects with respect to other DMP approaches. In fact, having a regular curve allows to overcome the problems related to motion discontinuities, as further discussed in the experimental results of Section 7.3. Furthermore, the concept of freely choosing the phase variable enables GDMP to be used for different applications, as further discussed in the remainder of this section.

From Remark 2, on the contrary with respect to other DMP approaches where the phase variable is still constrained to a canonical system, the phase variable $s(t)$ can be chosen based on the three distinct scenarios depicted in Fig. 5:

(1) Original: the demonstrated trajectory is reproduced using the original timing law $s(t) = \gamma(t)$.

(2) Offline Optimization: the phase variable $s(t)$ results from an optimization problem, which optimizes the motion along the geometric path to achieve a desired objective, as discussed in Section 5.

(3) Dynamical System: the phase variable $s(t)$ is defined by a dynamical system, possibly dependent on external inputs. This scenario includes co-manipulations tasks, as discussed in Section 6 and Section 7.

5. Phase optimization in geometric DMP

Many planning techniques focus on minimizing the execution time while satisfying task and kinematic constraints [34,38,45,54,55]. This optimization problem can be effectively addressed using the proposed GDMP framework, where the phase variable $s(t)$ is computed as the output of this dynamic system (i.e. a chain of two integrators):

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u,$$

where $\boldsymbol{x} = [s(t), \dot{s}(t), \ddot{s}(t)]^T$ and $u = \dddot{s}(t)$. The optimization problem can be formulated as follows [55]:

$$\min_{u(\cdot)} \quad T_f = \int_0^{T_f} 1 dt = \int_0^{\Delta M} \frac{1}{\dot{s}(t)} ds(t), \tag{23}$$

subject to:

$$s(0) = 0, \quad s(T_f) = \Delta M,$$
$$\dot{s}(0) = \dot{s}(T_f) = 0, \quad \dot{s} \geq 0,$$
$$\ddot{s}(0) = \ddot{s}(T_f) = 0.$$

Additional constraints can be imposed on $\boldsymbol{x}$ and u:

$$|\dot{s}(t)| \leq \dot{s}_{\text{MAX}}, \quad |\ddot{s}(t)| \leq \ddot{s}_{\text{MAX}}, \quad |\dddot{s}(t)| \leq \dddot{s}_{\text{MAX}}, \tag{24}$$

or on the robot velocity and acceleration in the task space:

$$\begin{aligned} \|\eta \dot{\boldsymbol{y}}^\star(s(t))\| &\leq \dot{\boldsymbol{y}}_{\text{MAX}} \quad \Leftrightarrow \quad \|\dot{\boldsymbol{y}}^\star(s(t))\| \leq \dot{\boldsymbol{y}}_{\text{MAX}}/\eta, \\ \|\eta \ddot{\boldsymbol{y}}^\star(s(t))\| &\leq \ddot{\boldsymbol{y}}_{\text{MAX}} \quad \Leftrightarrow \quad \|\ddot{\boldsymbol{y}}^\star(s(t))\| \leq \ddot{\boldsymbol{y}}_{\text{MAX}}/\eta. \end{aligned} \tag{25}$$

Alternatively, or in addition, constraints in the joint space can be considered as well, typically caused by the physical limitations of the actuation systems. Let $\boldsymbol{q}^\star(s(t)) \in \mathbb{R}^n$, with n the number of joints, be the joint trajectory profiles computed as follows:

$$\boldsymbol{q}^\star(s(t)) = \text{IK}\Big(\boldsymbol{y}_\eta^\star(s(t)), \boldsymbol{R}^\star(s(t))\Big),$$

where IK$(\cdot)$ is the Inverse Kinematics function of the robot. Function $\boldsymbol{y}_\eta^\star(s(t))$ is the demonstrated (position) trajectory defined in (14), while $\boldsymbol{R}^\star(s(t))$ is the desired orientation (in the simulation, $\boldsymbol{R}^\star(s(t))$ is kept constant but, in principle, it can also be varied during the demonstration). Velocity and acceleration constraints in the joint space can be expressed as:

$$|\dot{\boldsymbol{q}}^\star(s(t))| \leq \dot{\boldsymbol{q}}_{\text{MAX}}, \qquad |\ddot{\boldsymbol{q}}^\star(s(t))| \leq \ddot{\boldsymbol{q}}_{\text{MAX}}. \tag{26}$$

The practical definition of the constraints (26) is obtained by computing the joint configurations $\boldsymbol{q}_k$ from $(\eta \boldsymbol{y}_{\Delta,k}, \boldsymbol{R}_k)$, for $k = 0, \dots, M$, using the IK algorithm. Then, the parametric function $\boldsymbol{q}^\star(s)$ is deduced by applying the same algorithm used for $y^\star(s)$ in (10). Finally, the

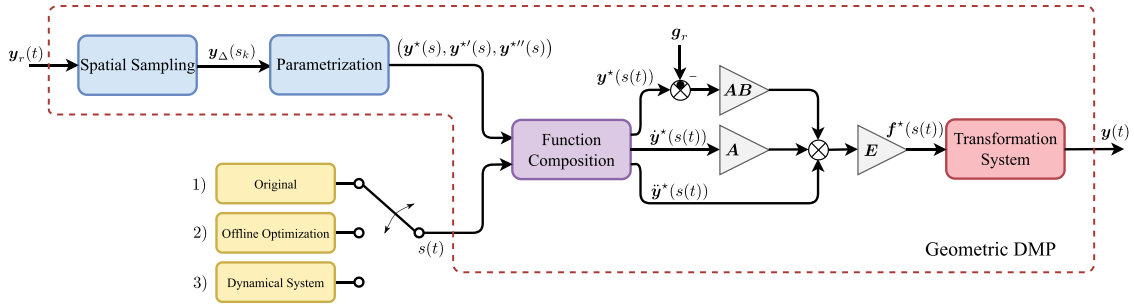


Fig. 5. Schematic of the proposed framework based on Geometric DMP.

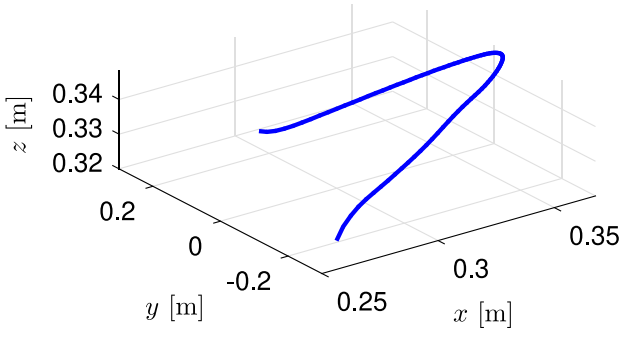


Fig. 6. Demonstrated curve used for optimization.

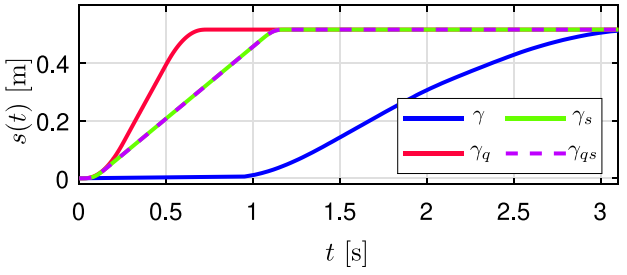


Fig. 7. Phase profiles obtained as a solution of the optimization problem (23) subject to constraints (24) and (26).

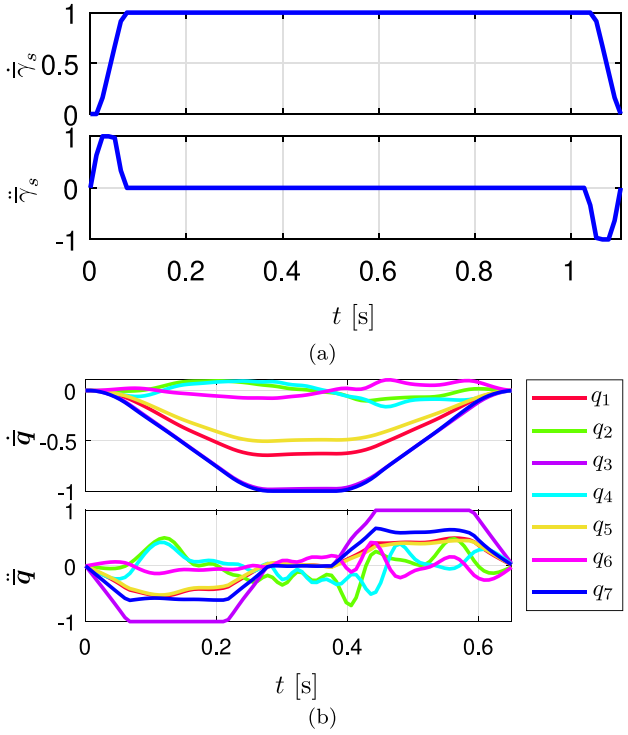


Fig. 8. Normalized results of the optimization problem (23) subject to (a) constraints (24) and (b) constraints (26).

constraints are defined as a function of the state $\boldsymbol{x} = [s(t), \dot{s}(t), \ddot{s}(t)]^T$ and of $\boldsymbol{q}^{\star\prime}(s)$, $\boldsymbol{q}^{\star\prime\prime}(s)$ which are computed analytically from $\boldsymbol{q}^\star(s)$. Therefore, the two inequalities become

$$\begin{array}{rcccl} -\dot{\boldsymbol{q}}_{\text{MAX}} & \leq & \boldsymbol{q}^{\star\prime}(s(t))\dot{s}(t) & \leq & \dot{\boldsymbol{q}}_{\text{MAX}}, \\ -\ddot{\boldsymbol{q}}_{\text{MAX}} & \leq & \boldsymbol{q}^{\star\prime\prime}(s(t))\dot{s}^2(t) + \boldsymbol{q}^{\star\prime}(s(t))\ddot{s}(t) & \leq & \ddot{\boldsymbol{q}}_{\text{MAX}}. \end{array}$$

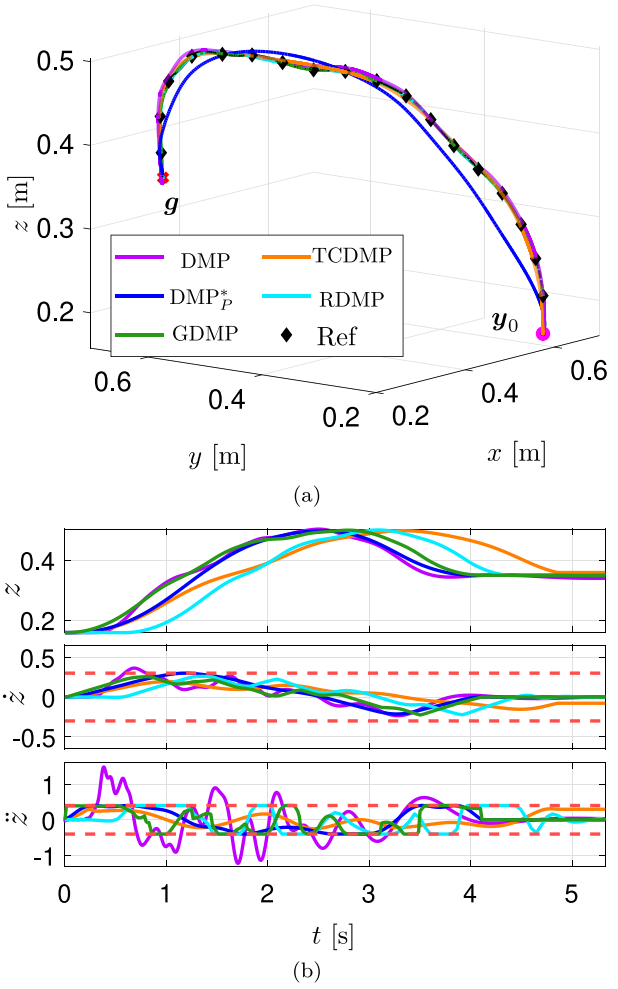
As a 7-DOF robot with a redundant kinematic structure, the inverse kinematics is not available analytically and must be computed numerically. This was implemented using the *generalizedInverseKinematics* package from MATLAB[1]; then, the considered optimization problem has been implemented using the CasADi optimal solver [56]. The reference curve $\boldsymbol{y}_\eta^\star(s)$ is depicted in Fig. 6, and the robot considered in the simulation is the same used in the experimental tests described in Section 7, i.e. a Franka Emika Panda robot.

Three different scenarios are considered: (a) the constraints (24), generating the timing law $s(t) = \gamma_s(t)$; (b) the constraints (26), generating the timing law $s(t) = \gamma_q(t)$; (c) both the constraints (24) and (26), generating the timing law $s(t) = \gamma_{qs}(t)$. The obtained results are compared in Fig. 7 with the original timing law $\gamma(t)$. The figure clearly shows that the execution time is significantly reduced compared to the demonstrated trajectory: $\gamma_s(t)$ and $\gamma_{qs}(t)$ result in an execution time $T_f = 1.1$ s, denoting a reduction of 64.5% with respect to $\gamma(t)$, while $\gamma_q(t)$ results in an execution time of $T_f = 0.6$s, denoting a reduction of 80.6% with respect to $\gamma(t)$. Indeed, one can conclude that the constraints (24) are more restrictive than (26).

The normalized profiles $\dot{\bar{\gamma}}_s = \dot{\gamma}_s(t)/\dot{s}_{\text{MAX}}$ and $\ddot{\bar{\gamma}}_s = \ddot{\gamma}_s(t)/\ddot{s}_{\text{MAX}}$ displayed in Fig. 8(a) confirm the optimality of the timing law $\gamma_s(t)$, since the boundary of at least one constraint in (24) is reached at every time instant. The same conclusion can be drawn for the timing law $\gamma_q(t)$ by referring to Fig. 8(b), presenting the normalized profiles $\dot{\bar{\boldsymbol{q}}} = \dot{\boldsymbol{q}}(\gamma_q(t))/\dot{\boldsymbol{q}}_{\text{MAX}}$ and $\ddot{\bar{\boldsymbol{q}}} = \ddot{\boldsymbol{q}}(\gamma_q(t))/\ddot{\boldsymbol{q}}_{\text{MAX}}$. In this case, the limiting factors are the velocity of joint #7 and the acceleration of joint #3.

5.1. Comparison with other DMPs

To demonstrate the effectiveness of the GDMP framework shown in Fig. 5, we applied it to solve the optimization problem (23), focusing on the task-space constraints described in (25). To provide a comprehensive evaluation, we compared its performance with several approaches from the literature, including the classical DMP [13], Temporal Coupling DMP (TCDMP) [37], the reversible DMP (RDMP) [19], and offline optimal DMP (DMP$^*_P$) [38]. These methods were applied to the same demonstrated trajectory, depicted in black in Fig. 9(a).



	Trajectory Reproduction	Constraints Satisfaction	Optimal Execution (min T)
DMP	✓	✗	✓
TCDMP	✓	✓	✗
RDMP	✓	✓	✗
DMP$^*_P$	✗	✓	✓
GDMP	✓	✓	✓

(c)

Fig. 9. Comparison of GDMP with other DMP solutions in addressing the minimum time optimization problem. (a) 3D path recordings; (b) z-axis' position, velocity and acceleration trajectories for each DMP model; (c) final evaluation table.

For what concerns GDMP, the trajectory was processed, and the optimization problem (23) was solved under the kinematic constraints (24) and (25). The optimization yielded an optimal task duration of $T_f^\star = 4.11s$. For consistency, the durations of the other methods were scaled to match $T_f^\star$, enabling a direct comparison of their adherence to the constraints and trajectory approximation.

The classical DMP achieves time scaling by modifying the temporal parameter τ, preserving the original curve shape, as shown in purple in Fig. 9(a). However, this approach does not inherently account for velocity or acceleration constraints, leading to violations of these constraints even when the desired task duration is achieved, see Fig. 9(b). In contrast, TCDMP leverages temporal coupling to enforce kinematic constraints by appropriately adjusting τ. While this ensures that the resulting trajectory stays within the constraint boundaries, it fails to meet the optimal duration $T^\star$, as the coupling mechanism indirectly affects task timing. Similarly, the RDMP structure allows addressing the optimization problem (23) via $s^\star(t)$ but fails to achieve $T_f^\star = 4.11s$ due to its reliance on time-parameterized demonstrations, which encode speed variations and pauses.

Finally, DMP$^*_P$ addresses these limitations by directly optimizing the weights $\boldsymbol{\omega}$ in (11), ensuring both compliance with the constraints and the desired task duration. However, strict kinematic constraints can lead to a compromise in the curve approximation, as the optimization prioritizes constraint satisfaction over precise interpolation of the demonstrated trajectory, which can be observed in Fig. 9(a).

The limitations of the existing approaches are summarized in the table of Fig. 9(c), where the terms "Trajectory Reproduction", "Constraints Satisfaction", and "Optimal Execution (min T)" indicate the ability to reproduce the demonstrated trajectory, satisfy kinematic constraints, and achieve the optimal duration $T^\star$, respectively. While each existing DMP solution has specific drawbacks, the proposed GDMP, as highlighted in green in Fig. 9(c), successfully fulfills all task requirements.

This success stems from the arc-length parameterization provided by the SS algorithm, which decouples the trajectory's path and velocity information. This decoupling allows the velocity profile to be computed independently, ensuring compliance with kinematic constraints while maintaining the desired trajectory shape and task duration. Consequently, GDMP offers a robust solution for tasks requiring high precision, constraint adherence, and time optimization.

6. Human-in-the-loop geometric DMP

The considered human–robot interaction framework employing GDMP is schematically depicted in Fig. 10. The phase variable $s(t)$ is determined as

$$m\,\ddot{s}(t) + b\,\dot{s}(t) = F_\tau(t) = \boldsymbol{y}_\eta^{\star\prime}(s(t))^T\,\boldsymbol{F}_h(t), \tag{27}$$

where $\boldsymbol{y}_\eta^\star(s) = \boldsymbol{E}\boldsymbol{y}^\star(s(t))$ is a generalization of (14) to the multidimensional case. Eq. (27) represents the dynamics of a mass m with damping coefficient b, influenced by the external force $\boldsymbol{F}_h(t)$ applied to the robot end-effector by a human operator. The presence of the scaling factor $\boldsymbol{E}$ in (27) brings the following two features:

(1) It enables a consistent human–robot interaction, as further discussed in the remainder of this section;

(2) It ensures the passivity of the human–robot interaction framework in Fig. 10, as proven in Section 6.1.

The force $F_\tau(t)$ in (27) denotes the projection of the human force vector $\boldsymbol{F}_h(t)$ onto the tangential direction of the reference trajectory $\boldsymbol{y}_\eta^\star(s) = \boldsymbol{E}\boldsymbol{y}^\star(s(t))$, where $\boldsymbol{y}^\star(s(t))$ is the parameterized trajectory as depicted in the GDMP schematic of Fig. 5. From (21), it follows that $\|\boldsymbol{y}_\eta^{\star\prime}(s(t))\| \neq 0$, meaning that the projection in (27) can always be successfully computed [11,57]. The fact that, thanks to the spatial sampling and the resulting arc-length parameterization, $\|\boldsymbol{y}_\eta^{\star\prime}(s(t))\|$ is not only non-zero but also constant,[2] makes the projection of the force along the tangent direction both constant and intuitive, avoiding modulation effects that can arise from variations in the magnitude of the tangent vector. Naturally, in the case of a time-parameterized trajectory—as in [27]—where the tangent vector may have variable magnitude, it is possible to normalize it by dividing it by its norm, i.e. $\boldsymbol{t} = \frac{d\boldsymbol{y}_r(t)}{dt} / \left\| \frac{d\boldsymbol{y}_r(t)}{dt} \right\|$. However, this expression becomes undefined when $\left\| \frac{d\boldsymbol{y}_r(t)}{dt} \right\| = 0$, a situation that can occur not only during unintended stops, but also at the initial and final points of the demonstrated trajectory. In [27], to prevent the tangent component of the reference velocity from approaching zero, a minimum threshold is enforced.

[2] Specifically, $\|\boldsymbol{y}_\eta^{\star\prime}(s(t))\| = |\eta|$.

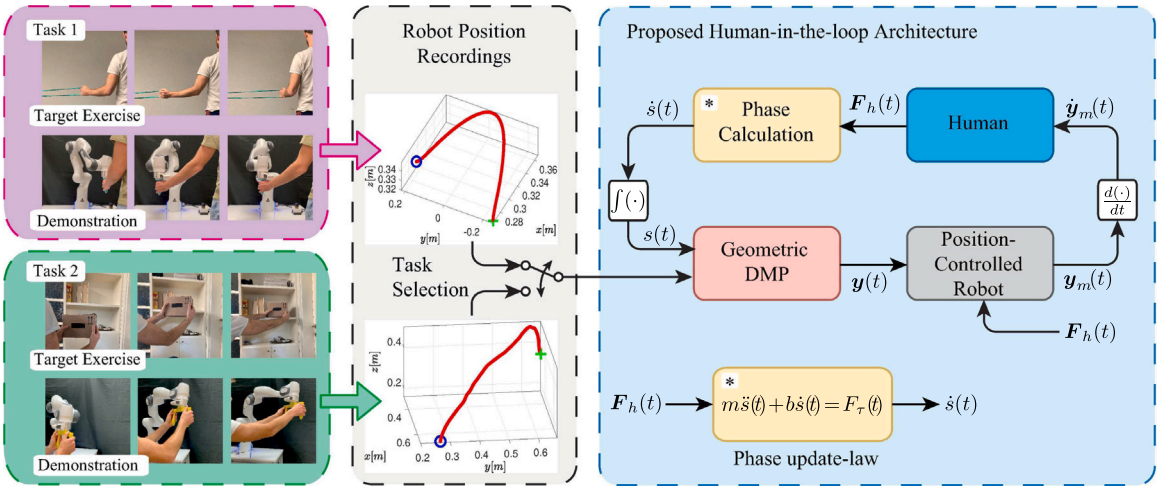


Fig. 10. Human–robot interaction through kinesthetic guidance along the trajectories encoded by GDMP.

With reference to the aforementioned feature (1), the inclusion of matrix $\boldsymbol{E}$ ensures that the tangential direction remains consistent with the current orientation of the new curve. This feature proves valuable as it guarantees an accurate projection of the force $\boldsymbol{F}_h(t)$ in (27) without the necessity of a new DMP parameterization [22,41].

As shown in the framework of Fig. 10, an additional term $\boldsymbol{F}_h(t)$ is inserted in Eqs. (13) and (15), leading to the following DMP formulation:

$$\begin{aligned}&\ddot{\boldsymbol{y}}(t)+\boldsymbol{A}\dot{\boldsymbol{y}}(t)+\boldsymbol{A}\boldsymbol{B}\boldsymbol{y}(t)-\boldsymbol{A}\boldsymbol{B}\boldsymbol{g}=\\ &\boldsymbol{E}\left[\ddot{\boldsymbol{y}}^{\star}(s(t))+\boldsymbol{A}\dot{\boldsymbol{y}}^{\star}(s(t))+\boldsymbol{A}\boldsymbol{B}\boldsymbol{y}^{\star}(s(t))-\boldsymbol{A}\boldsymbol{B}\boldsymbol{g}_r\right]+\boldsymbol{F}_h(t),\end{aligned} \tag{28}$$

where the additional term $\boldsymbol{F}_h(t)$ ensures the passivity of the human–robot interaction framework of Fig. 10, as proven in Section 6.1. The motion trajectory $\boldsymbol{y}(t)$ is then fed into a robot manipulator equipped with high-gain position control, which essentially implements an admittance control system for the physical human–robot interaction enhanced by a Virtual fixture (VF) [58]. In this setup, the admittance model restricts the user's movement to follow the trajectories defined by the GDMP, allowing forward or backward motion. As shown in [58], a feasible approach for ensuring a safe human–robot interaction is to impose the passivity of the system with respect to the energetic port $\langle \dot{\boldsymbol{y}}_m(t), \boldsymbol{F}_h(t)\rangle$ describing the interaction between the human and the controlled robot. In case the human behaves passively, this condition ensures the stability of the system composed of the human and the robot. As a final remark, it is worth noting that GDMP can be employed to implement different control design philosophies, such as an impedance-type virtual fixture, in which the user's input force directly influences the robot, while an elastic force constrains its motion towards the desired path. This strategy has been adopted in [31], where a Dynamical System (DS) is used to encode the motion, and the DS phase variable is updated based on the point on the learned trajectory that is closest to the robot's current pose.

Within this framework, the phase variable of the GDMP can be efficiently computed using a numerical optimization method, such as the Gauss–Newton algorithm, that minimizes the distance between the robot's pose and the reference path [32]. In this context, the fact that the first derivative is always non-zero is crucial in ensuring the convergence of the numerical optimization problem, while the arc-length parameterization offers a clear and interpretable representation of the progression along the trajectory.

6.1. Passivity analysis

Definition 1. Let $P_T(t) = \sum_{i=1}^{n} P_i(t) = \boldsymbol{u}^T(t)\boldsymbol{z}(t)$ be the total power flowing through the n energetic ports of a dynamical system, where $\boldsymbol{u}(t)$ and $\boldsymbol{z}(t)$ are the input and output vectors. Such a system is said to be passive if there exists a non-negative storage function $S(\boldsymbol{x}(t))$ of the state vector $\boldsymbol{x}(t)$ such that, for any time interval $[t_0, t_1]$ and for any initial state vector $\boldsymbol{x}_0$, and being $\boldsymbol{x}_1$ the state vector at time t_1, the following inequality holds true [59]:

$$S(\boldsymbol{x}_1) - S(\boldsymbol{x}_0) \leq \int_{t_0}^{t_1} P_T(t)\,dt \;\leftrightarrow\; \dot{S}(x(t)) \leq P_T(t). \tag{29}$$

Property 1. *The human–robot interaction framework of Fig. 10 is passive with respect to the energetic port $\langle \dot{\boldsymbol{y}}_m(t), \boldsymbol{F}_h(t)\rangle$, where $\dot{\boldsymbol{y}}_m(t)$ is the manipulator velocity.*

Proof. Let variables $\tilde{\boldsymbol{y}}$, $\dot{\tilde{\boldsymbol{y}}}$ and $\ddot{\tilde{\boldsymbol{y}}}$ be defined as:

$$\begin{aligned}\tilde{\boldsymbol{y}}(t) &= \boldsymbol{y}(t) - \boldsymbol{g} - \boldsymbol{E}\boldsymbol{y}^{\star}(s(t)) + \boldsymbol{E}\boldsymbol{g}_r,\\ \dot{\tilde{\boldsymbol{y}}}(t) &= \frac{d\,\tilde{\boldsymbol{y}}(t)}{dt} = \dot{\boldsymbol{y}}(t) - \boldsymbol{E}\dot{\boldsymbol{y}}^{\star}(s(t)),\\ \ddot{\tilde{\boldsymbol{y}}}(t) &= \frac{d\,\dot{\tilde{\boldsymbol{y}}}(t)}{dt} = \ddot{\boldsymbol{y}}(t) - \boldsymbol{E}\ddot{\boldsymbol{y}}^{\star}(s(t)),\end{aligned} \tag{30}$$

where $\boldsymbol{g}$ and $\boldsymbol{g}_r$ are the goals of the desired and parameterized demonstrated trajectories $\boldsymbol{y}(t)$ and $\boldsymbol{y}^{\star}(s(t))$, respectively. Using (30), (28) can be rewritten as:

$$\ddot{\tilde{\boldsymbol{y}}}(t) + \boldsymbol{A}\dot{\tilde{\boldsymbol{y}}}(t) + \boldsymbol{A}\boldsymbol{B}\tilde{\boldsymbol{y}}(t) = \boldsymbol{F}_h(t), \tag{31}$$

The following storage function $S(\tilde{\boldsymbol{y}}(t), \dot{\tilde{\boldsymbol{y}}}(t))$ for the system in (27) and (31) can be considered:

$$S(\tilde{\boldsymbol{y}}(t), \dot{\tilde{\boldsymbol{y}}}(t)) = \underbrace{\frac{1}{2}\dot{\tilde{\boldsymbol{y}}}^T(t)\dot{\tilde{\boldsymbol{y}}}(t)}_{E_{K_1}} + \underbrace{\frac{1}{2}\tilde{\boldsymbol{y}}^T(t)\boldsymbol{A}\boldsymbol{B}\tilde{\boldsymbol{y}}(t)}_{E_U} + \underbrace{\frac{1}{2}m\dot{s}^2(t)}_{E_{K_2}}, \tag{32}$$

where E_{K_1} and E_U represent the kinetic and potential energies, respectively, stored in the Geometric DMP of Fig. 10, while E_{K_2} is the kinetic energy of the Phase Calculation block of Fig. 10. Differentiating the storage function $S(\tilde{\boldsymbol{y}}(t), \dot{\tilde{\boldsymbol{y}}}(t))$ in (32) with respect to time yields:

$$\dot{S}(t) = \dot{\tilde{\boldsymbol{y}}}^T(t)\left[\ddot{\tilde{\boldsymbol{y}}}(t) + \boldsymbol{A}\boldsymbol{B}\tilde{\boldsymbol{y}}(t)\right] + m\dot{s}(t)\ddot{s}(t). \tag{33}$$

Substituting $\ddot{\tilde{\boldsymbol{y}}}(t)$ from (31) in (33) yields:

$$\begin{aligned}\dot{S}(t) =& \dot{\tilde{\boldsymbol{y}}}^T(t)\left[-\boldsymbol{A}\dot{\tilde{\boldsymbol{y}}}(t) - \boldsymbol{A}\boldsymbol{B}\tilde{\boldsymbol{y}}(t) + \boldsymbol{F}_h(t) + \boldsymbol{A}\boldsymbol{B}\tilde{\boldsymbol{y}}(t)\right] +\\ &+ m\dot{s}(t)\ddot{s}(t)\,.\end{aligned} \tag{34}$$

Substituting $\ddot{s}(t)$ from (27) in (34) yields:

$$\begin{aligned}\dot{S}(t) =& -\dot{\tilde{\boldsymbol{y}}}^T(t)\boldsymbol{A}\dot{\tilde{\boldsymbol{y}}}(t) + \dot{\tilde{\boldsymbol{y}}}^T(t)\boldsymbol{F}_h(t) +\\ &- b\dot{s}^2(t) + \dot{s}(t)\boldsymbol{y}^{\star\prime}(s(t))^T\boldsymbol{E}\boldsymbol{F}_h(t)\,.\end{aligned} \tag{35}$$

Finally, substituting $\dot{\tilde{\boldsymbol{y}}}(t)$ from (30) in (35) and recalling that $\boldsymbol{y}^{\star\prime}(s(t))\dot{s}(t) = \dot{\boldsymbol{y}}^{\star}(s(t))$ from (12) yields:

$$\begin{aligned}\dot{S}(t) =& -\dot{\tilde{\boldsymbol{y}}}^T(t)\boldsymbol{A}\dot{\tilde{\boldsymbol{y}}}(t) + \left[\dot{\boldsymbol{y}}(t) - \boldsymbol{E}\dot{\boldsymbol{y}}^{\star}(s(t))\right]^T\boldsymbol{F}_h(t) +\\ &- b\dot{s}^2(t) + \dot{\boldsymbol{y}}^{\star T}(s(t))\boldsymbol{E}\boldsymbol{F}_h(t),\\ =& -\dot{\tilde{\boldsymbol{y}}}^T(t)\boldsymbol{A}\dot{\tilde{\boldsymbol{y}}}(t) + \dot{\boldsymbol{y}}^T(t)\boldsymbol{F}_h(t) - b\dot{s}^2(t).\end{aligned} \tag{36}$$

(a) (b) (c)

Fig. 11. The considered experimental tests: (a) Internal/External rotation of the shoulder, (b) Bi-manual grabbing, (c) Stop at a point.

Considering the power $P_T = \dot{\boldsymbol{y}}_m^T(t)\boldsymbol{F}_h(t)$ flowing through the energetic port $\langle \dot{\boldsymbol{y}}_m(t), \boldsymbol{F}_h(t)\rangle$ and using (36), the passivity condition (29) is fulfilled. Indeed, assuming perfect tracking by the robot of the reference trajectory generated by the GDMP, i.e. $\boldsymbol{y}_m(t) = \boldsymbol{y}(t)$ and consequently $\dot{\boldsymbol{y}}_m(t) = \dot{\boldsymbol{y}}(t)$, the inequality

$$-\tilde{\dot{\boldsymbol{y}}}^T(t)\boldsymbol{A}\tilde{\dot{\boldsymbol{y}}}(t) + \dot{\boldsymbol{y}}^T(t)\boldsymbol{F}_h(t) - b\dot{s}^2(t) \leq \dot{\boldsymbol{y}}_m^T(t)\boldsymbol{F}_h(t) \tag{37}$$

transforms into $-\tilde{\dot{\boldsymbol{y}}}^T(t)\boldsymbol{A}\tilde{\dot{\boldsymbol{y}}}(t) - b\dot{s}^2(t) \leq 0$, which is always satisfied for $\boldsymbol{A} > 0$ and $b > 0$.

7. Human–robot interaction: Experimental evaluation

This section presents and discusses the experimental tests conducted to validate the human–robot interaction framework of Fig. 10. Section 7.1 outlines the utilized setup and details the experimental tests performed for the tuning of parameters m and b in (27) through the practical stability analysis of Section 7.2. The results have shown the emergence of instability phenomena such as vibrations and oscillations despite the passivity proof of Section 6.1, where perfect tracking of the GDMP trajectory is assumed. This motivates the introduction of the stability analysis performed in Section 7.2. Finally, Sections 7.3 and 7.4 address the comparison of GDMP with other DMP solutions with reference to an insertion task and a simulated welding task, showing the enhanced performance of GDMP [3] .

7.1. Experimental setup

The experimental setup involves the use of a Franka Emika Panda robot in the GDMP framework of Fig. 10 for co-manipulation tasks. Three distinct experimental tests have been designed as illustrated in Fig. 11, requiring the user to manipulate the robot end-effector along the desired position trajectory $\boldsymbol{y}(t)$ generated by the GDMP. The manipulator position control is developed in the MATLAB/Simulink environment. The experiments involved 12 participants of different gender, aged between 20 and 60, out of whom only three possessed prior expertise in robotics. All the participants provided informed consent prior to participation.

In the test of Fig. 11(a), the user is required to perform an internal and external rotation of the shoulder in the upright position. In the test of Fig. 11(b), the user is required to lead the end-effector from the initial to the goal position using a bi-manual grabbing in a seated position. Finally, in the experiment of Fig. 11(c), the user is required to move the end-effector from the initial position to a given point located along the demonstrated trajectory. For each task, an expert user demonstrated the reference position trajectory $\boldsymbol{y}_r(t)$ through kinesthetic guidance. The parameterized position trajectory $\boldsymbol{y}^\star(s(t))$ has then been generated as depicted in Fig. 5.

[3] The video can be found at: https://youtu.be/t2qeiCsfRZE

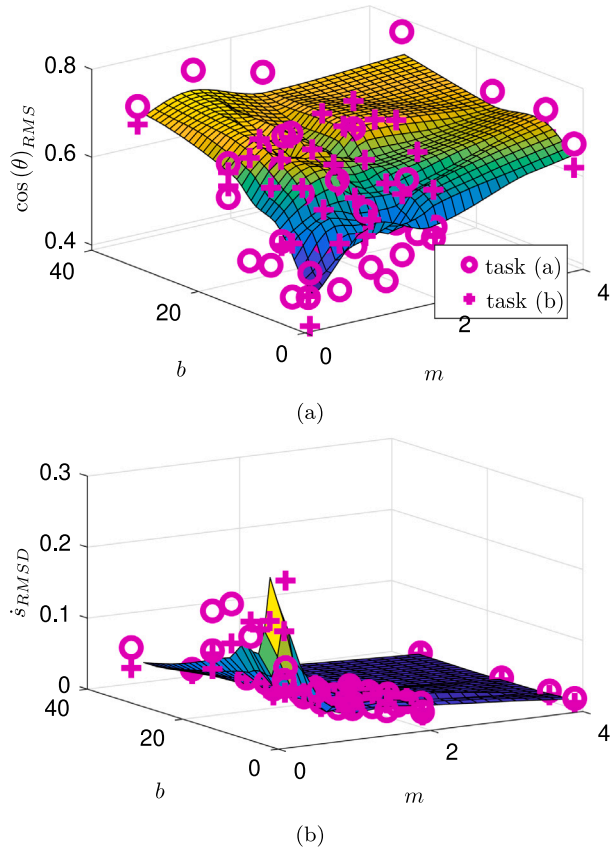


Fig. 12. Trend of (a) external force application angle $\cos(\theta)_{RMS}$ and (b) peak-to-peak oscillations $\dot{s}_{RMSD}$ on the phase velocity.

7.2. Parameters tuning through practical stability analysis

In the GDMP of (15), the parameters $\alpha_x = \alpha_y = \alpha_z = 40$ and $\beta_x = \beta_y = \beta_z = 10$ have been assigned to the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, in order to ensure a sufficiently high stiffness in (31) guaranteeing $\boldsymbol{y}(t) \approx \boldsymbol{y}^\star(s(t))$. The choice of m and b in (27) is pivotal in practical applications. In order to extract first-attempt values for m and b, the Franka Emika Panda specifications have been considered: $\dot{\boldsymbol{y}}_{\text{MAX}} = 1.7$ m/s, $\ddot{\boldsymbol{y}}_{\text{MAX}} = 13.0$ m/s^2 and a maximum payload of 3 kg, which implies a force limit of $F_{\text{MAX}} = 30$ N. Under the nominal condition $\boldsymbol{E} = \boldsymbol{I}_3$ and recalling (21), it follows that $\|\dot{\boldsymbol{y}}^\star(s(t))\| = |\dot{s}(t)|$ and, if the curvature is negligible, $\|\ddot{\boldsymbol{y}}^\star(s(t))\| \approx |\ddot{s}(t)|$ from (12). Consequently, the following specifications can be derived:

$$\dot{s}_{\text{MAX}} = 1.7 \text{ m/s} \quad \text{and} \quad \ddot{s}_{\text{MAX}} = 13.0 \text{ m/s}^2. \tag{38}$$

When the velocity $\dot{s}(t)$ is sufficiently low, the following values of m and b can be determined from (27) and (38):

$$m = \frac{F_{\text{MAX}}}{\ddot{s}_{\text{MAX}}} \approx 2 \text{ kg} \quad \text{and} \quad b = \frac{F_{\text{MAX}}}{\dot{s}_{\text{MAX}}} \approx 17 \text{ N s/m}. \tag{39}$$

Starting from (39), the following ranges have been experimentally investigated:

$$\mathcal{N}_m = [0.2,\ \ 4] \text{ kg} \quad \text{and} \quad \mathcal{N}_b = [1.7,\ \ 34] \text{ N s/m}. \tag{40}$$

The three tests in Fig. 11 have been executed by each user for 60 s using m and b in (40). The results of the tests of Fig. 11(a) and Fig. 11(b) are shown in Fig. 12. The metric in Fig. 12(a) is defined as follows:

$$\cos(\theta)_{RMS} = \text{rms}\big(\cos(\theta(t))\big), \quad \text{with} \quad \cos(\theta(t)) = \frac{F_\tau(t)}{\|\boldsymbol{F}_h(t)\|},$$

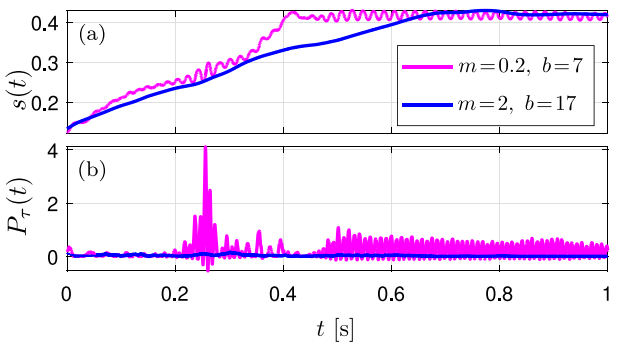


Fig. 13. (a) Phase variable $s(t)$ and (b) power $P_\tau(t) = F_\tau(t)\dot{s}(t)$ exchanged between human and the robot.

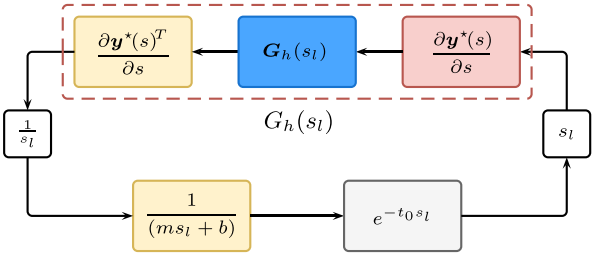


Fig. 14. Linearized Human–robot interaction framework through kinesthetic guidance.

giving the normalized effort that the user has to make in order to move along the desired trajectory. The metric in Fig. 12(b) is defined as follows:

$$\dot{s}_{RMSD} = \mathrm{rms}\big(\dot{s}(t) - \dot{s}(t)_{\mathrm{AVG}}\big),$$

where $\dot{s}(t)_{\mathrm{AVG}}$ is the moving average of the function $\dot{s}(t)$. This metric quantifies the average magnitude of the oscillations affecting the phase velocity $\dot{s}(t)$. The surface plot of Fig. 12(a) suggests that, as parameters m and b decrease, the user effort required to execute the task results to be lower as expected. However, as shown in Fig. 12(b), this also leads to higher oscillations affecting the phase velocity $\dot{s}(t)$, which may cause the robot to stop because of violation of the safe limit constraints. These oscillations are also evident in Fig. 13, showing the results of the test in Fig. 11(c). Consequently, for lower values of m and b falling within the yellow peak in Fig. 12(b), the user's attempt to counteract the robot oscillations results in a larger recorded power $P_\tau(t)$.

This oscillating behavior can be explained using the following practical stability analysis on the system of Fig. 10 based on the Lyapunov's indirect method [60]. Although the robot is typically assumed to perfectly track the desired reference, the bandwidth of the position control (Fig. 10) is finite in practical applications. To account for this, a finite delay t_0 between the robot position $\boldsymbol{y}_m(t)$ and the desired reference $\boldsymbol{y}(t)$ has been introduced. Although providing an accurate description of the human is non-trivial [61], the following simplified spring-damper model exhibiting isotropic behavior across all spatial directions is employed in this work:

$$\boldsymbol{F}_h = G_h(s_l)\boldsymbol{I}_3\dot{\boldsymbol{y}}_m, \qquad \text{where} \qquad G_h(s_l) = \frac{B_h s_l + K_h}{s_l}$$

and where s_l is the Laplace complex variable. Assuming that the GDMP behaves as an ideal trajectory generator, i.e. $\boldsymbol{y}(t) \approx \boldsymbol{y}^\star(s(t))$, and linearizing the system of Fig. 10 around the equilibrium point $(s_e, \dot{s}_e) = (\bar{s}, 0)$, where $\bar{s}$ represents the stop position, the scheme illustrated in Fig. 14 can be derived, which is characterized by the following input–output transfer function:

$$G_{IO}(s_l) = \frac{G_{LG}(s_l)}{1+G_{LG}(s_l)}, \quad \text{where} \quad G_{LG}(s_l) = \frac{G_h(s_l)e^{-t_0 s_l}}{ms_l+b} \tag{41}$$

is the loop-gain function of the system. The qualitative Nyquist diagram of $G_{LG}(s_l)$ is shown in Fig. 15. The Nyquist criterion states that the

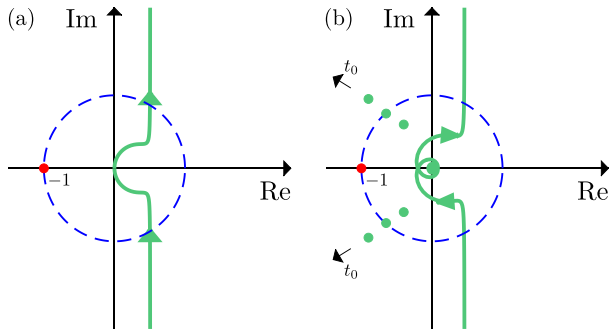


Fig. 15. Nyquist diagram of the loop-gain function $G_{LG}(s_l)$ without (a) and with (b) finite delay t_0.

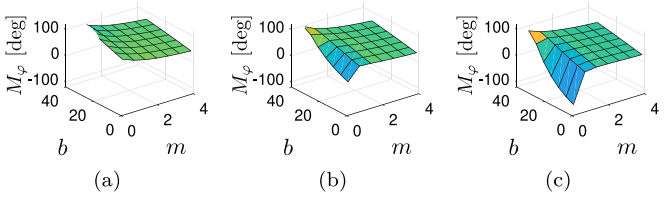


Fig. 16. Phase margin M_φ of the loop-gain function $G_{LG}(s_l)$ as a function of m, b for different values of the delay t_0: $t_0 = 0$ s (a), $t_0 = 0.01$ s (b), and $t_0 = 0.02$ s (c).

closed-loop system is asymptotically stable as long as the Nyquist contour of $G_{LG}(s_l)$ does not touch or encircle the critical point $-1 + j0$ [62]. Based on this, Fig. 15 clearly shows that a large delay t_0 can lead to an unstable system. The stability margin M_φ [62] of the closed-loop human–robot interaction framework is shown in Fig. 16. Fig. 16(a) shows that $M_\varphi > 0$ is verified $\forall m, b$ when $t_0 = 0$, proving the system stability as predicted in the passivity proof of Section 6.1. As t_0 increases, Fig. 16(b) and Fig. 16(c) show that the system becomes unstable for small values of m and b because $M_\varphi < 0$. This well agrees with the oscillations affecting the phase variable $s(t)$ in the experimental results of Fig. 12(b) and Fig. 13. It can be concluded that the parameters m and b in the phase calculation Eq. (27) play an important role in determining the effort perceived by the user when moving along the desired trajectory, and they need to be sufficiently large in order ensure the stability of the considered human-in-the-loop system.

7.3. Insertion task

To further evaluate the GDMP architecture in Fig. 5, a further co-manipulation task was conducted using $m = 2$ and $b = 17$ for robustness, as discussed previously. The task involved the guidance of the end-effector handle in order to insert its tip into one of the four holes in the red box shown in Fig. 17.

The insertion task was initially demonstrated via kinesthetic guidance. Starting from the position in Fig. 17(a), the robot was guided above the target hole (Fig. 17(c)), paused for few seconds, and then moved into the hole to complete the task (Fig. 17(d)- 17(e)). During the execution, the goal position of the DMP architecture was varied in order to select each of the four holes of the box. The GDMP architecture was compared with three other DMP methodologies: reversible DMP (RDMP) [19], speed-scaled DMP (SSDMP) [63], and arc-length DMP (ALDMP) [53]. Each approach was tested three times for all four holes of the box, and the results are presented in Fig. 18 and Fig. 19.

Fig. 18 shows the 3D plots of the task execution for each DMP architecture. The position trajectories are color-coded based on the magnitude of the measured force norm $\|\boldsymbol{F}_h\|$ during the execution, where the force data have been acquired using a Schunk FT-AXIA force/torque sensor. To allow the comparison across the different DMP

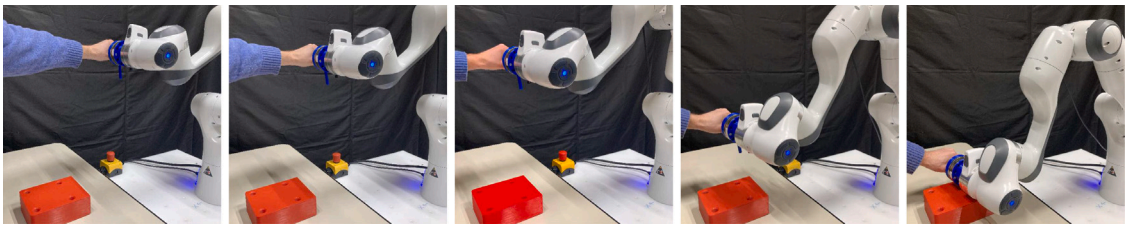


Fig. 17. Snapshots of the insertion task. During the demonstration by kinesthetic teaching, the user moves horizontally the robot over the target hole from its start position (Fig. 17(a)- 17(b)), then the user stops for few seconds (Fig. 17(c)) and moves vertically the end-effector tip towards the hole in the red box to complete the insertion task (Fig. 17(d)- 17(e)).

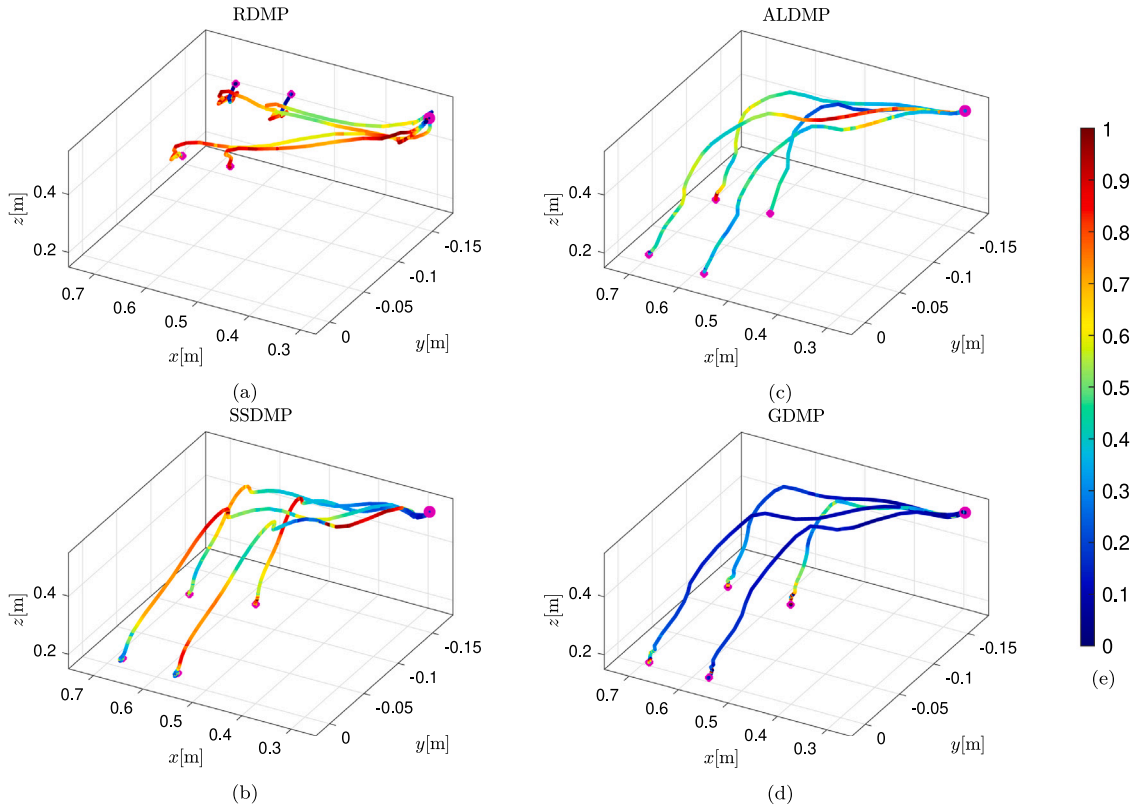


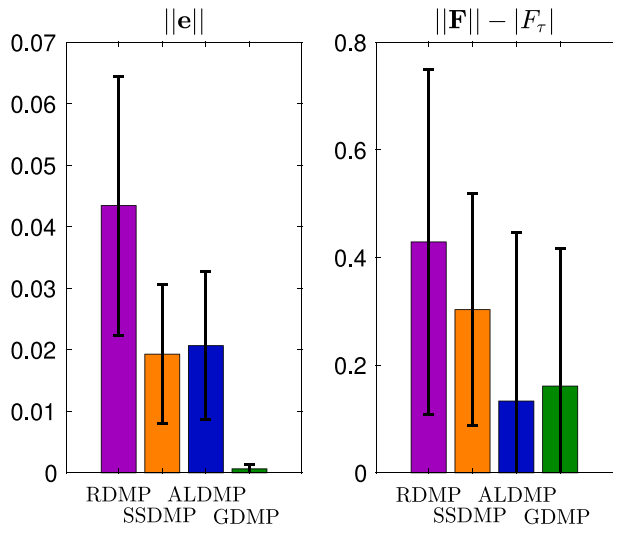
Fig. 18. Plots of the recorded trajectories when varying the goal position. Each position trajectory has been reported with varying color, which encodes the measured force norm $\|\boldsymbol{F}_h\|$ normalized to the maximum measured component inside the same experimental trial.

methods, all executions were normalized to their maximum measured force. A colorbar for the force magnitude is provided in Fig. 18(e).

For what concerns the RDMP case shown in Fig. 18(a), a high force demand is evident due to the lack of arc-length parameterization, which prevents proper tangential projection of the forces along the trajectory. This issue becomes critical near the stopping interval recorded during the demonstration, where the tangential component is undefined, making it impossible for the user to complete the insertion task. In the SSDMP case shown in Fig. 18(b), although a higher force demand is observed, the user successfully completes the task. This is because the SSDMP autonomously progresses its phase, actively guiding the user. However, the tests have shown that the SSDMP's phase velocity adjustment based on the user's feedback lacks reactivity, leading to difficulty switching between active and passive guidance, and resulting in a higher force demands during the execution.

Significantly lower force measurements are observed for the ALDMP case shown in Fig. 18(c). This is because the arc-length parameterization used by ALDMP allows a consistent computation of the tangential component of the desired curve at each instant, ensuring the correct projection of the measured force $\boldsymbol{F}_h$. However, the color code shows frequent force variations, as the ALDMP algorithm does not maintain a constant tangential norm, unlike the proposed SS algorithm. In the GDMP case shown in Fig. 18(d), the SS algorithm ensures that the tangential component of the geometric curve has unitary norm, as guaranteed by the filtering action obtained in property (21) which ensures the regularity of the curve. This results in consistent projection of $\boldsymbol{F}_h$ throughout the task through (27), reducing the user's effort when moving the robot, as evident from the trajectory color code.

The task execution is further analyzed in the results of Fig. 19. The statistical results are reported in Fig. 19(a) based on: (i) the norm of the displacement error $\|\boldsymbol{e}\|$, where $\boldsymbol{e}(t) = \boldsymbol{y}(t) - \boldsymbol{y}_m(t)$; and (ii) the difference $\|\boldsymbol{F}_h\| - |F_\tau|$, representing the force used to keep the end-effector near the trajectory $\boldsymbol{y}$ provided by the DMP solution. The proposed GDMP achieves more accurate position tracking, as shown in Fig. 18, thanks to its more effortless guidance compared to other DMP methods. Additionally, the lower difference $\|\boldsymbol{F}_h\| - |F_\tau|$ indicates that, like ALDMP, the GDMP provides more intuitive guidance. This is because a greater portion of the applied force is projected tangentially to the curve, making the robot guidance easier to follow.



(a)

	Reversibility	Insertion Task Completion	Managing Pausing Intervals
RDMP	✓	✗	✗
SSDMP	✗	✓	✓
ALDMP	✗	✓	✓
GDMP	✓	✓	✓

(b)

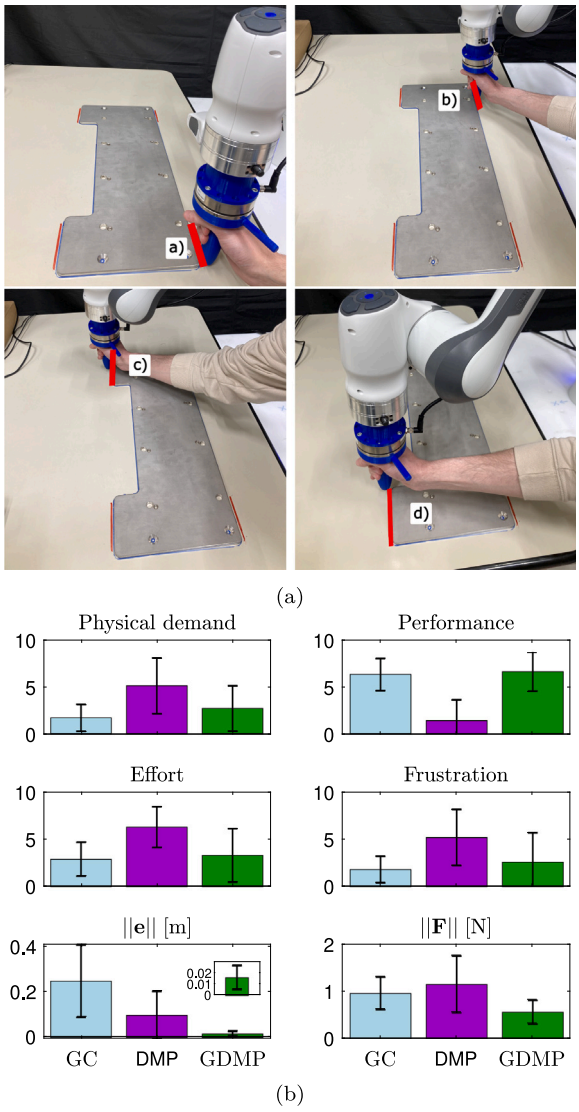
Fig. 19. Comparison of GDMP with other DMP solutions in addressing the insertion task. (a) Obtained results in term of average norm of the error displacement $\|\boldsymbol{e}\|$, representing the spatial accuracy, and the average norm of the non-tangential force component $\|\boldsymbol{F}_h\| - |F_\tau|$, which reflects the undesired forces orthogonal to the task direction. (b) Final evaluation table.

The limitations of the analyzed DMP approaches when compared with GDMP are summarized in the table in Fig. 19(b), where "Reversibility", "Insertion Task Completion", and "Managing Pausing Intervals" refer to the DMP's ability to be reversible, complete the insertion task, and handle pausing intervals recorded during the demonstration, respectively. The RDMP fails to accomplish the task due to its reliance on time parameterization, which prevents it from managing pausing intervals. The SSDMP and ALDMP lack reversibility, making them unsuitable for co-manipulation tasks. While the SSDMP can handle pausing intervals by actively guiding the user and resuming movement after the pause, its execution remains time-dependent on the demonstration, posing limitations. In contrast, the proposed GDMP satisfies all the requirements, demonstrating its enhanced performance over the other approaches.

7.4. User experience in a welding co-manipulation task

With this final experiment, we aim to evaluate the impact of adopting the GDMP framework in a co-manipulation task, directly from the user's point of view. The considered task simulates the welding of a mechanical part, performed with robotic guidance. For this purpose, the same handle used in the previous section was employed to mimic a welding tool. The participants were asked to trace a designated welding path across four different sections of a metal plate, as shown in Fig. 20(a). Even in this case, the task was initially demonstrated via kinesthetic guidance.

A total of 10 participants were involved in the study. Each participant received verbal instructions and performed the task in a one-shot



(a)

(b)

Fig. 20. Welding task. (a) An illustration of the target task. (b) User-experience results. The first and second row report the results from the NASA-TLX questionnaire, while the third row reports the measurements of the error and force norm.

trial, without prior training. The task was executed under three different robot control conditions: gravity compensation (GC), Classical DMP, and GDMP. Quantitative measurements were recorded, including displacement error ($\|\mathbf{e}\|$) and force norm ($\|\mathbf{F}\|$). In addition to the quantitative data, the participants completed a NASA Task Load Index questionnaire [64], rating the following aspects on a scale from 0 to 9:

- **Physical demand**: How physically demanding was the task?
- **Performance**: How successful were you in accomplishing the task?
- **Effort**: How hard did you have to work to achieve your performance level?
- **Frustration**: How insecure, discouraged, irritated, stressed, or annoyed did you feel?

All participants provided informed consent according to the institutional policies.

The results are summarized in Fig. 20(b). Among the three modes, DMP yielded the poorest performance. In this configuration, participants frequently triggered safety threshold violations due to excessive applied force, which caused the robot to stop. This behavior was likely caused by speed variations in the task demonstration, where

users unintentionally applied excessive force while the DMP controller operated more slowly, as it is constrained to the learned phase profile. Consequently, users were often unable to complete the task, leading to a negative experience.

In contrast, the GC and GDMP modes yielded similar results in terms of user experience, with GC being perceived slightly more favorably. This outcome may be attributed to the one-shot nature of the experiment, where participants had no prior training. In the GC condition, the robot is fully compliant, which facilitates task execution for inexperienced users. However, robot guidance, as employed in the GDMP mode, significantly improved the task performance, as indicated in the final row of Fig. 20(b). Significantly lower values of $\|\mathbf{e}\|$ and $\|\mathbf{F}\|$ in the GDMP case indicate much better accuracy and reduced effort compared to GC, where users must exert more force in the absence of guidance.

8. Discussion

A key feature of GDMP is the use of arc-length parameterization of the learned trajectory. While this concept is not novel [53], we have significantly enhanced its application through the proposed Spatial Sampling (SS) algorithm. It is important to emphasize that SS, despite its simple and intuitive formulation, ensures consistency and strong guarantees in terms of curve regularity. Moreover, it is capable of handling pause intervals within demonstrated trajectories, which was not solved before. This feature has proven particularly valuable in co-manipulation tasks, where maintaining the regularity of the trajectory is essential to ensure consistent external force projection along the desired path. Such consistency directly affects task execution and user performance.

The GDMP framework was initially validated in a simulated environment (Section 5), where we computed the optimal profile of the phase variable to minimize task duration, subject to joint and workspace constraints on velocity and acceleration. The resulting phase profile serves as input to the forcing term, enabling fast task execution while respecting these constraints. Although the current implementation operates offline, limiting adaptability to real-time changes in start and goal positions, it outperformed other DMP variants in the literature, showing promising results.

Unlike related approaches, GDMP was then extensively validated across five co-manipulation tasks, demonstrating adaptability to varying task requirements. By designing a custom phase law (Eq. (27)), we analyzed the influence of parameters m and b on task execution, allowing real-time modulation by the user. As discussed in Section 7, although the GDMP is theoretically passive, modeling assumptions can introduce approximation errors that may compromise stability. This limitation was qualitatively observed across different users and (m, b) parameter combinations in the first three exercises. To address this, we defined a practical procedure to tune these parameters for ensuring stability and passivity in human-in-the-loop settings. This tuning process offers a valuable tool for embedding safety and stability, critical but non-trivial aspects in human–robot collaboration due to user unpredictability.

Finally, GDMP was tested in two real-world tasks to evaluate its performance: one quantitative (insertion) and one qualitative (welding). On one hand, in the insertion task GDMP demonstrated consistent user force projection along the path constraint, showcasing the benefits of the SS algorithm. Although GDMP outperformed baseline methods, the end-effector's orientation was fixed, and future work should explore whether force consistency is preserved under varying orientations. On the other hand, the welding task emphasized user perception. Despite lower objective performance, the gravity compensation mode was initially preferred by users as it felt less frustrating and cognitively demanding. This outcome, although counterintuitive given the purposes of virtual constraints [65], likely reflects users' initial reluctance to accept guidance. However, the robot guidance of GDMP significantly improved the task performance, since significantly lower values of tracking error and user force norm were measured. While beyond this paper's scope, extended user studies could assess how task load indices evolve with increased training, potentially improving user confidence and familiarity with the co-manipulation framework.

9. Conclusions

The Canonical System of the DMP controls the evolution of phase variable, and thus of the forcing term, shaping the attractor landscape for the desired output dynamics. While standard DMP ensure time modulation, no unified framework exists in the literature to compactly handle multiple phase profiles without significantly altering the DMP formulation. The study proposed in this paper introduces a new concept called Geometric Dynamic Movement Primitives (GDMP), thanks to which the phase variable can be freely chosen depending on the desired application. The concept of GDMP is based on the proposed spatial sampling algorithm, which decouples the demonstrated curve from its timing law thus allowing to generate an arc-length parameterized geometric path. The proposed spatial sampling algorithm guarantees the parameterized curves to always be regular, thus ensuring a consistent projection of the human force throughout the task in a human-in-the-loop scenario. The effectiveness of the GDMP has been demonstrated through two main applications. The first one is an offline optimization problem for minimum task duration, subject to velocity and acceleration constraints, which has highlighted the GDMP's superiority over previous DMP solutions thanks to the separation of path and velocity. The second application of GDMP is human-in-the-loop involving different co-manipulation tasks. For this application, an analytical passivity analysis and an analytical/experimental stability analysis have been carried out. Finally, the proposed human-in-the-loop architecture is further validated with reference to an insertion task and a simulated welding task, where GDMP is experimentally compared against other DMP solutions and is shown to consistently outperform them.

Nomenclature

$\boldsymbol{g}$	goal position
$\boldsymbol{y}_r(t), \dot{\boldsymbol{y}}_r(t), \ddot{\boldsymbol{y}}_r(t)$	demonstrated trajectories
T_f	task duration
T	sampling-time
$\boldsymbol{f}(s(t))$	forcing term
$\boldsymbol{f}^\star(s(t))$	parameterized forcing term
$s(t)$	phase variable
$\boldsymbol{y}^\star(t), \dot{\boldsymbol{y}}^\star(t), \ddot{\boldsymbol{y}}^\star(t)$	parameterized task trajectories
$\boldsymbol{y}(t), \dot{\boldsymbol{y}}(t), \ddot{\boldsymbol{y}}(t)$	GDMP trajectories
$\alpha/\boldsymbol{A}$	friction coefficient/matrix
$\alpha\beta/\boldsymbol{AB}$	spring stiffness/matrix
$\eta/\boldsymbol{E}$	scaling coefficient/matrix
$\boldsymbol{q}^\star(t), \dot{\boldsymbol{q}}^\star(t), \ddot{\boldsymbol{q}}^\star(t)$	parameterized joint trajectories
$\boldsymbol{y}_m(t), \dot{\boldsymbol{y}}_m(t), \ddot{\boldsymbol{y}}_m(t)$	actual robot trajectories
$\boldsymbol{Y}_T$	input samples matrix
$\boldsymbol{t}_T$	time vector associated with $\boldsymbol{Y}_T$
$\boldsymbol{y}_L(t)$	linearly interpolated trajectory
Δ	spatial sampling interval
$\boldsymbol{y}_{\Delta,k}(\boldsymbol{s}_k)$	spatially-sampled trajectory
$\boldsymbol{s}_k$	phase vector
$\boldsymbol{t}_\Delta$	time vector associated with $\boldsymbol{s}_k$
$\boldsymbol{Y}_{\Delta,k}$	output samples matrix
ΔM	length of $\boldsymbol{y}_r(t)$
m	mass parameter
b	damping parameter
$\boldsymbol{F}_h(t)$	applied user force vector
$F_\tau(t)$	user tangential force
s_l	Laplace complex variable

CRediT authorship contribution statement

Giovanni Braglia: Writing – review & editing, Project administration, Formal analysis, Writing – original draft, Supervision, Visualization, Resources, Investigation, Validation, Software, Methodology, Data curation, Conceptualization. **Davide Tebaldi:** Methodology, Investigation, Writing – review & editing, Supervision, Software, Resources, Project administration, Formal analysis. **Luigi Biagiotti:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition, Formal analysis, Conceptualization.

Funding

The work was partly supported by the University of Modena and Reggio Emilia, Italy through the action FARD (Finanziamento Ateneo Ricerca Dipartimentale) 2022/2023 and 2023/2024, and funded under the National Recovery and Resilience Plan (NRRP), Mission 04 Component 2 Investment 1.5 – NextGenerationEU, Call for tender n. 3277 dated 30/12/2021 Award Number: 0001052 dated 23/06/2022.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The link to data, code and video are available in the main pdf manuscript.

References

[1] L. Zhang, S. Guo, Q. Sun, Development and assist-as-needed control of an end-effector upper limb rehabilitation robot, Appl. Sci. 10 (19) (2020) 6684, http://dx.doi.org/10.3390/app10196684.

[2] X. Yu, S. Liu, S. Zhang, W. He, H. Huang, Adaptive neural network force tracking control of flexible joint robot with an uncertain environment, IEEE Trans. Ind. Electron. 71 (6) (2024) 5941–5949, http://dx.doi.org/10.1109/TIE.2023.3290250.

[3] A.D. Deshpande, Novel biomedical technologies: Rehabilitation robotics, Curr. Opin. Biomed. Eng. 22 (2022) http://dx.doi.org/10.1016/j.cobme.2022.100371.

[4] C. Lauretti, F. Cordella, E. Guglielmelli, L. Zollo, Learning by demonstration for planning activities of daily living in rehabilitation and assistive robotics, IEEE Robot. Autom. Lett. 2 (3) (2017) 1375–1382, http://dx.doi.org/10.1109/LRA.2017.2669369.

[5] N. Hogan, D. Sternad, Dynamic primitives in the control of locomotion, Front. Comput. Neurosci. 7 (2013) 71, http://dx.doi.org/10.3389/fncom.2013.00071.

[6] X. Yu, B. Li, W. He, Y. Feng, L. Cheng, C. Silvestre, Adaptive-constrained impedance control for human–robot co-transportation, IEEE Trans. Cybern. 52 (12) (2022) 13237–13249, http://dx.doi.org/10.1109/TCYB.2021.3107357.

[7] S. Calinon, F. Guenter, A. Billard, On learning, representing, and generalizing a task in a humanoid robot, IEEE Trans. Syst. Man Cybern. B 37 (2) (2007) 286–298, http://dx.doi.org/10.1109/TSMCB.2006.886952.

[8] A. Billard, S. Calinon, R. Dillmann, S. Schaal, Survey: Robot programming by demonstration, Springer Handb. Robot. (2008) 1371–1394, http://dx.doi.org/10.1007/978-3-540-30301-5_60.

[9] S. Schaal, Is imitation learning the route to humanoid robots? Trends Cogn. Sci. 3 (6) (1999) 233–242, http://dx.doi.org/10.1016/s1364-6613(99)01327-3.

[10] G.E. Hovland, P. Sikka, B.J. McCarragher, Skill acquisition from human demonstration using a hidden markov model, in: Proceedings of IEEE International Conference on Robotics and Automation, 3, Ieee, 1996, pp. 2706–2711, http://dx.doi.org/10.1109/ROBOT.1996.506571.

[11] L. Sciavicco, B. Siciliano, L. Villani, G. Oriolo, Robotics: Modelling, planning and control, 2011, http://dx.doi.org/10.1007/978-1-84628-642-1.

[12] M. Saveriano, F.J. Abu-Dakka, A. Kramberger, L. Peternel, Dynamic movement primitives in robotics: A tutorial survey, 2021, http://dx.doi.org/10.1177/02783649231201196, arXiv preprint arXiv:2102.03861.

[13] A.J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, S. Schaal, Dynamical movement primitives: Learning attractor models for motor behaviors, Neural Comput. 25 (2) (2013) 328–373, http://dx.doi.org/10.1162/NECO_a_00393.

[14] A. Ijspeert, J. Nakanishi, S. Schaal, Movement imitation with nonlinear dynamical systems in humanoid robots, in: Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292), 2, 2002, pp. 1398–1403 vol.2, http://dx.doi.org/10.1109/ROBOT.2002.1014739.

[15] R.L. Devaney, Reversible diffeomorphisms and flows, Trans. Amer. Math. Soc. 218 (1976) 89–113, http://dx.doi.org/10.2307/1997429.

[16] C. Atkeson, Using local models to control movement, Adv. Neural Inf. Process. Syst. 2 (1989).

[17] P. Pastor, H. Hoffmann, T. Asfour, S. Schaal, Learning and generalization of motor skills by learning from demonstration, in: 2009 IEEE International Conference on Robotics and Automation, 2009, pp. 763–768, http://dx.doi.org/10.1109/ROBOT.2009.5152385.

[18] B. Nemec, A. Gams, M. Deni0161a, A. Ude, Human-robot cooperation through force adaptation using dynamic motion primitives and iterative learning, in: 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014), 2014, pp. 1439–1444, http://dx.doi.org/10.1109/ROBIO.2014.7090536.

[19] A. Sidiropoulos, Z. Doulgeri, A reversible dynamic movement primitive formulation, in: 2021 IEEE International Conference on Robotics and Automation, ICRA, 2021, pp. 3147–3153, http://dx.doi.org/10.1109/ICRA48506.2021.9562059.

[20] B. Nemec, A. Gams, A. Ude, Velocity adaptation for self-improvement of skills learned from user demonstrations, in: 2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids), 2013, pp. 423–428, http://dx.doi.org/10.1109/HUMANOIDS.2013.7030009.

[21] E.A. Rückert, A. d'Avella, Learned parametrized dynamic movement primitives with shared synergies for controlling robotic and musculoskeletal systems, Front. Comput. Neurosci. 7 (2013) http://dx.doi.org/10.3389/fncom.2013.00138.

[22] A. Sidiropoulos, Z. Doulgeri, Dynamic via-points and improved spatial generalization for online trajectory planning with dynamic movement primitives, 2023, https://Arxiv.Org/Abs/2212.13473.

[23] B. Kiss, E. Szádeczky-Kardoss, Time-scaling in the control of mechatronic systems, New Dev. Robot. Autom. Control. (2008) 411–426, http://dx.doi.org/10.5772/6270.

[24] M. Sampei, K. Furuta, On time scaling for nonlinear systems: Application to linearization, IEEE Trans. Autom. Control 31 (5) (1986) 459–462, http://dx.doi.org/10.1109/TAC.1986.1104290.

[25] R. Samant, L. Behera, G. Pandey, Adaptive learning of dynamic movement primitives through demonstration, in: 2016 International Joint Conference on Neural Networks, IJCNN, 2016, pp. 1068–1075, http://dx.doi.org/10.1109/IJCNN.2016.7727316.

[26] T. Kulvicius, K. Ning, M. Tamosiunaite, F. Wörgötter, Joining movement sequences: Modified dynamic movement primitives for joining movement sequences, in: 2011 IEEE International Conference on Robotics and Automation, IEEE, 2011, pp. 2275–2280, http://dx.doi.org/10.1109/TRO.2011.2163863.

[27] B. Nemec, M. Simoni010D, T. Petri010D, A. Ude, Incremental policy refinement by recursive regression and kinesthetic guidance, in: 2019 19th International Conference on Advanced Robotics, ICAR, 2019, pp. 344–349, http://dx.doi.org/10.1109/ICAR46387.2019.8981606.

[28] A. Kramberger, E. Shahriari, A. Gams, B. Nemec, A. Ude, S. Haddadin, Passivity based iterative learning of admittance-coupled dynamic movement primitives for interaction with changing environments, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2018, pp. 6023–6028, http://dx.doi.org/10.1109/IROS.2018.8593647.

[29] B. Nemec, A. Ude, et al., Speed adaptation for self-improvement of skills learned from user demonstrations, Robotica 34 (12) (2016) 2806–2822, http://dx.doi.org/10.1109/HUMANOIDS.2013.7030009.

[30] G. Braglia, M. Tagliavini, F. Pini, L. Biagiotti, Online motion planning for safe human-robot cooperation using B-splines and hidden Markov models, MDPI Robot. (ISSN: 2218-6581) 12 (4) (2023) http://dx.doi.org/10.3390/robotics12040118.

[31] Z.D. D. Papageorgiou, A control scheme for haptic inspection and partial modification of kinematic behaviors, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2020, pp. 9752–9758, http://dx.doi.org/10.1109/IROS45743.2020.9341594.

[32] G. Braglia, S. Calinon, L. Biagiotti, A minimum-jerk approach to handle singularities in virtual fixtures, IEEE Robot. Autom. Lett. 9 (11) (2024) 10256–10263, http://dx.doi.org/10.1109/LRA.2024.3469814.

[33] T. Lipp, S. Boyd, Minimum-time speed optimisation over a fixed path, Int. J. Control. 87 (6) (2014) 1297–1311, http://dx.doi.org/10.1080/00207179.2013.875224.

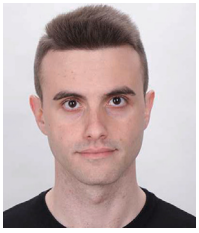
[34] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, M. Diehl, Time-optimal path tracking for robots: A convex optimization approach, IEEE Trans. Autom. Control 54 (10) (2009) 2318–2327, http://dx.doi.org/10.1109/TAC.2009.2028959.

[35] G. Chen, F. Dellaert, S. Hutchinson, Generalizing trajectory retiming to quadratic objective functions, in: 2024 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2024, pp. 9823–9829, http://dx.doi.org/10.1109/ICRA57147.2024.10610854.

[36] A. Dahlin, Y. Karayiannidis, Adaptive trajectory generation under velocity constraints using dynamical movement primitives, IEEE Control. Syst. Lett. 4 (2) (2019) 438–443, http://dx.doi.org/10.1109/LCSYS.2019.2946761.

[37] A. Dahlin, Y. Karayiannidis, Temporal coupling of dynamical movement primitives for constrained velocities and accelerations, IEEE Robot. Autom. Lett. 6 (2) (2021) 2233–2239, http://dx.doi.org/10.1109/LRA.2021.3058874.
[38] A. Sidiropoulos, D. Papageorgiou, Z. Doulgeri, A novel framework for generalizing dynamic movement primitives under kinematic constraints, Auton. Robots 47 (1) (2023) 37–50, http://dx.doi.org/10.1007/s10514-022-10067-4.
[39] B. Nemec, L. Žlajpah, S. Šlajpa, J. Piškur, A. Ude, An efficient pbd framework for fast deployment of bi-manual assembly tasks, in: 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids), IEEE, 2018, pp. 166–173, http://dx.doi.org/10.1109/HUMANOIDS.2018.8625010.
[40] I. Iturrate, C. Sloth, H.G. Kramberger, E.H. 00D8stergaard, T.R. Savarimuthu, Towards reversible dynamic movement primitives, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2019, pp. 5063–5070, http://dx.doi.org/10.1109/IROS40897.2019.8968270.
[41] L. Koutras, Z. Doulgeri, A novel dmp formulation for global and frame independent spatial scaling in the task space, in: 2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), IEEE, 2020, pp. 727–732, http://dx.doi.org/10.1109/RO-MAN47096.2020.9223500.
[42] R.J. Escarabajal, J.L. Pulloquinga, P. Zamora-Ortiz, Á. Valera, V. Mata, M. Vallés, Imitation learning-based system for the execution of self-paced robotic-assisted passive rehabilitation exercises, IEEE Robot. Autom. Lett. (2023) http://dx.doi.org/10.1109/LRA.2023.3281884.
[43] S. Wiggins, S. Wiggins, M. Golubitsky, Introduction to Applied Nonlinear Dynamical Systems and Chaos, Springer, 2003.
[44] H. Zhao, Y. Chen, X. Li, H. Ding, Robotic peg-in-hole assembly based on reversible dynamic movement primitives and trajectory optimization, Mechatronics 95 (2023) 103054, http://dx.doi.org/10.1016/j.mechatronics.2023.103054.
[45] J. Jankowski, L. Brudermüller, N. Hawes, S. Calinon, VP-STO: Via-point-based stochastic trajectory optimization for reactive robot behavior, in: 2023 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2023, pp. 10125–10131, http://dx.doi.org/10.48550/arXiv.2210.04067.
[46] G. Raiola, X. Lamy, F. Stulp, Co-manipulation with multiple probabilistic virtual guides, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2015, pp. 7–13, http://dx.doi.org/10.1109/IROS.2015.7353107.
[47] S.S. Restrepo, G. Raiola, P. Chevalier, X. Lamy, D. Sidobre, Iterative virtual guides programming for human-robot comanipulation, in: 2017 IEEE International Conference on Advanced Intelligent Mechatronics, AIM, IEEE, 2017, pp. 219–226, http://dx.doi.org/10.1109/AIM.2017.8014021.
[48] H.B. Amor, G. Neumann, S. Kamthe, O. Kroemer, J. Peters, Interaction primitives for human-robot cooperation tasks, in: 2014 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2014, pp. 2831–2837, http://dx.doi.org/10.1109/ICRA.2014.6907265.
[49] D.J. Berndt, J. Clifford, Using dynamic time warping to find patterns in time series, in: KDD Workshop, 1994.
[50] L. Koutras, Z. Doulgeri, A correct formulation for the orientation dynamic movement primitives for robot control in the cartesian space, in: Conference on Robot Learning, PMLR, 2020, pp. 293–302.
[51] A. Gams, T. Petrič, M. Do, B. Nemec, J. Morimoto, T. Asfour, A. Ude, Adaptation and coaching of periodic motion primitives through physical and visual interaction, Robot. Auton. Syst. 75 (2016) 340–351, http://dx.doi.org/10.1016/j.robot.2015.09.011.
[52] V.A. Toponogov, Differential Geometry of Curves and Surfaces, Springer, 2006, http://dx.doi.org/10.1007/b137116.
[53] T. Gašpar, B. Nemec, J. Morimoto, A. Ude, Skill learning and action recognition by arc-length dynamic movement primitives, Robot. Auton. Syst. (ISSN: 0921-8890) 100 (2018) 225–235, http://dx.doi.org/10.1016/j.robot.2017.11.012.
[54] A. Kanazawa, J. Kinugawa, K. Kosuge, Adaptive motion planning for a collaborative robot based on prediction uncertainty to enhance human safety and work efficiency, IEEE Trans. Robot. 35 (4) (2019) 817–832, http://dx.doi.org/10.1109/TRO.2019.2911800.
[55] H. Gattringer, A. Mueller, M. Oberherber, D. Kaserer, Time-optimal robotic manipulation on a predefined path of loosely placed objects: Modeling and experiment, Mechatronics 84 (2022) 102753, http://dx.doi.org/10.1016/j.mechatronics.2022.102753.
[56] J.A.E. Andersson, J. Gillis, G. Horn, J.B. Rawlings, M. Diehl, CasADi – a software framework for nonlinear optimization and optimal control, Math. Program. Comput. 11 (1) (2019) 1–36, http://dx.doi.org/10.1007/s12532-018-0139-4.
[57] M. Simonič, T. Petrič, A. Ude, B. Nemec, Analysis of methods for incremental policy refinement by kinesthetic guidance, J. Intell. Robot. Syst. 102 (1) (2021) 1–19, http://dx.doi.org/10.1007/s10846-021-01328-y.
[58] A.Q. Keemink, H. van der Kooij, A.H. Stienen, Admittance control for physical human–robot interaction, Int. J. Robot. Res. 37 (11) (2018) 1421–1444, http://dx.doi.org/10.1177/0278364918768950.
[59] E. Colgate, N. Hogan, The interaction of robots with passive environments: Application to force feedback control, in: Advanced Robotics: 1989: Proceedings of the 4th International Conference on Advanced Robotics Columbus, Ohio, June 13–15, 1989, Springer, 1989, pp. 465–474.
[60] H.K. Khalil, Nonlinear systems third edition, Patience Hall 115 (2002).
[61] F. Ficuciello, A. Romano, L. Villani, B. Siciliano, Cartesian impedance control of redundant manipulators for human-robot co-manipulation, in: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2014, pp. 2120–2125, http://dx.doi.org/10.1109/IROS.2014.6942847.
[62] K. Ogata, Y. Yang, Modern Control Engineering, vol. 5, Prentice hall India, 2002.
[63] B. Nemec, N. Likar, A. Gams, A. Ude, Human robot cooperation with compliance adaptation along the motion trajectory, Auton. Robot. 42 (2018) 1023–1035, http://dx.doi.org/10.1007/s10514-017-9676-3.
[64] S. Hart, L. Staveland, Development of NASA-TLX (task load index): Results of empirical and theoretical research, in: Advances in Psychology, 52, Elsevier, 1988, pp. 139–183, http://dx.doi.org/10.1016/S0166-4115(08)62386-9.
[65] S. Bowyer, B. Davies, F.R. y Baena, Active constraints/virtual fixtures: A survey, IEEE Trans. Robot. 30 (1) (2013) 138–157, http://dx.doi.org/10.1109/TRO.2013.2283410.

Giovanni Braglia received the M.Sc. degree in electronics engineering in 2021 from the University of Modena and Reggio Emilia, Modena, Italy, and from the Federal Technological University of Paraná, Curitiba, Brazil. He has recently complete the PhD at the University of Modena and Reggio Emilia and he is now a Post-Doc at the Italian Institute of Technology in the Biomedical Robotics Laboratory. His research interests include motion planning and control of robotics systems, human–robot interaction, robot-assisted surgery.



Davide Tebaldi received the Bachelor's degree in electronics engineering and the Master's degree (cum laude) in electronics engineering from the University of Modena and Reggio Emilia, Italy, in 2015 and 2018, respectively, and the Ph.D. degree in information and communication technologies from the University of Modena and Reggio Emilia, in 2022. His Ph.D. thesis is titled "Mathematical Modeling Control and Simulation of Hybrid Electric Vehicles." From October 2021 to January 2022, he was a Visiting Scholar with the Center for Automotive Research, The Ohio State University, OH, USA. From March 2022 to February 2023, he held a postdoctoral research position with the University of Modena and Reggio Emilia, where he has been an Assistant Professor, since March 2023. He is author or co-author of several publications in international scientific journals and international scientific conferences. His research interests include the energetic modeling, control, energy efficiency analysis and simulation of mechatronic systems, with main application in the automotive, electrical machines, and power electronics fields. His research interests also include collaborative robotics, energy efficiency improvement, and safety in human–robot interaction.

Luigi Biagiotti 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.