IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Dynamic Movement Primitives With Control Barrier Functions for Constrained Trajectory Planning

Federico Vesentini, Daniele Meli, Nicola Sansonetto, Luca Di Persio, and Riccardo Muradore, *Member, IEEE*

***Abstract*—Dynamic Movement Primitives (DMPs) form a robust framework for trajectory generation based on imitation learning, aiming to replicate the shape of reference trajectories from demonstrations closely. DMPs have been extensively employed for *trajectory planning* in robotic systems. However, they cannot safely guarantee complex *nonlinear constraints*, which is essential *at the control level*. On the other hand, Control Barrier Functions (CBFs) are used to modulate the input of control-affine dynamic systems subject to state-dependent constraints, guaranteeing that the system remains within predefined safe sets while converging towards target states. This letter proposes Constrained Movement Primitives (CMPs), a novel framework that integrates DMPs with CBFs to generate safe-by-construction trajectories subject to nonlinear constraints. We represent DMPs in control-affine form and combine them with the closed-form input provided by CBFs, overcoming the limitations of existing iterative optimisation methods for constrained DMPs. We demonstrate that CBFs preserve the goal convergence guarantees of DMPs. Moreover, we validate our approach in simulation and on a real mobile robot subject to *nonlinear kinodynamic constraints*, concerning maximum Cartesian velocity, obstacle avoidance, and maximum centrifugal acceleration to avoid slippery over curved trajectories.**

***Index Terms*—Imitation learning, collision avoidance, constrained motion planning, nonholonomic motion planning.**

I. Introduction

DYNAMIC Movement Primitives (DMPs) [1] is a framework for trajectory learning and planning, where a parametric perturbation or *forcing* term is added to the equation for acceleration to "mimic" a reference profile, for example, a human demonstration. Specifically, in the *learning* phase, a reference trajectory is recorded, and the parameters of the forcing term are learned to closely approximate the corresponding acceleration profile. In the *execution* phase, the learnt forcing term is used to generate new trajectories of similar shape but generalising to different initial and goal positions, time durations, and obstacle avoidance. Researchers have improved the original DMP formulation [2] to address practical robotic issues, e.g., planning in the quaternion space [3], learning from multiple reference trajectories [4], and dealing with dynamic obstacles [5]. This has fostered the application of DMPs to challenging robotic domains, e.g., deformable tissue manipulation [6], exoskeletons [7] and surgical automation [8].

However, DMPs do not guarantee *feasibility* in the execution phase when the trajectory is generalised to different workspace configurations. This is a crucial limitation in scenarios such as human-robot interaction [9], and autonomous driving [10], where the *safety* of the robot must be preserved under dynamic and human constraints. Currently, only linear constraints have been considered on DMP-driven robotic systems, e.g., to satisfy kinematic and actuation constraints [11], [12]. Associative skill memories were proposed by [13] to replicate force profiles from human demonstrations. [14] introduced a modulation factor with admittance control to generalise the force profile over different (curved) surfaces. However, acting only on the forcing term still does not guarantee proper constraint satisfaction, and embedding complex kinodynamic constraints is still an open area of research [15]. Recently, [16] embedded more complex task-specific constraints into the framework of probabilistic DMPs, which still do not guarantee constraint satisfaction.

More generally, provably safe motion planning is an open area of research. While exact constrained optimization techniques exist [17], they often require significant computational resources and multiple iterations, especially when optimizing over a time horizon [11]. On the other hand, more efficient learning-based approaches have been proposed [18], but they lack formal safety guarantees [19].

In this letter, we propose a novel framework to embed *nonlinear kinodynamic constraints* into DMPs with Control Barrier Functions (CBFs) [20]. CBFs are a well-established approach to ensure that the trajectories generated by a dynamic system in control-affine form stay within a safe set while minimising goal tracking. In particular, they allow the definition of an additional control input to the dynamic system *in closed form*, hence overcoming current iterative optimisation approaches to linearly constrained DMPs [11]. Then CBFs provide a provably safe and computationally effective solution for constrained complex robotic applications, e.g., single-robot navigation [21] and multi-robot coordination [22].

Received 17 June 2025; accepted 14 July 2025. Date of publication 4 August 2025; date of current version 12 August 2025. This article was recommended for publication by Associate Editor S. Chang Ryu and Editor C. Gosselin upon evaluation of the reviewers' comments. *(Federico Vesentini and Daniele Meli contributed equally to this work.) (Corresponding authors: Federico Vesentini.)*

Federico Vesentini and Riccardo Muradore are with the Department of Engineering for Innovation Medicine, University of Verona, 37129 Verona, Italy (e-mail: federico.vesentini@univr.it; riccardo.muradore@univr.it).

Daniele Meli, Nicola Sansonetto, and Luca Di Persio are with the Department of Computer Science, University of Verona, 37129 Verona, Italy (e-mail: daniele.meli@univr.it; nicola.sansonetto@univr.it; luca.persio@univr.it).

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2025.3594987, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3594987

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Our contributions are summarised as follows:

- we formulate a novel framework to integrate DMPs and CBFs for the generation of *Constrained Movement Primitives* (CMPs), subject to *nonlinear constraints*;
- we theoretically prove the safe convergence of CMPs to the original DMPs goal, thus also extending the state of the art in provably safe constrained motion planning;
- we show the advantages of CMPs in simulated and real robotic navigation, subject to realistic *kinodynamic constraints*, such as velocity limits, obstacle avoidance, and maximum centrifugal acceleration to avoid slipping.

The article is organised as follows: Section II provides a brief overview of the theory behind DMPs and CBFs. In Section III, we introduce CMPs, the proposed framework that unifies DMPs and CBFs, and the nonlinear kinodynamic constraints under discussion. Section IV presents and analyses the results from the numerical and physical implementation of CMPs. Finally, Section V summarises our findings and draws some conclusions.

II. Background

In this section, we recall some fundamental notions about Cartesian DMPs and CBFs that are necessary to explain the main contribution of this article.

A. Dynamic Movement Primitives

A DMP is a system of differential equations with an attractor (goal) and a learnt forcing term encoding the shape of the trajectory. Let $\boldsymbol{x} = (p, v) \in \mathbb{R}^3 \times \mathbb{R}^3$ be the system's state, encoding Cartesian position and velocity. The DMP framework models a trajectory as follows

$$\tau \dot{p} = v \tag{1a}$$

$$\tau \dot{v} = K(p_g - p) - Dv - K(p_g - p_0)s + Kf(s) \tag{1b}$$

where $\tau > 0$ is time-scaling factor, p_g is the goal position, p_0 is the initial position, K and D are stiffness and damping coefficients, respectively. The forcing term $f(s)$ to be learnt is given by

$$f(s) = \frac{\sum_{i=1}^{N} \psi_i(s) w_i}{\sum_{i=1}^{N} \psi_i(s)} s,$$

where ψ_i are basis functions, e.g., *Gaussian radial basis functions*, and $s \in [0, 1]$ is a reparametrization of time, solution to the *canonical system*

$$\tau \dot{s} = -\alpha_s s \tag{2}$$

where $s(0) = 1$ and $\alpha_s > 0$ is the decay rate of s.

Theorem 1: [23] The dynamic system (1a)–(1b) with s governed by the canonical system (2) converges to the goal state $\boldsymbol{x}_g = (p_g, 0)$.

It is possible to embed collision avoidance in DMPs, as proposed by [5], [13], by defining a suitable potential function, $U(p, v)$, and adding its gradient

$$f_{obs}(p, v) = -\nabla_p \left(U(p, v) \right) \tag{3}$$

to the dynamic system (1b).

B. Control Barrier Functions

CBFs [20], [24], [25] are applied in the context of nonlinear dynamic systems in control-affine form, i.e.

$$\dot{\boldsymbol{x}} = f_0(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u} \tag{4}$$

with state $\boldsymbol{x} \in \mathbb{R}^n$, input $\boldsymbol{u} \in \mathbb{R}^m$ and $f_0 : \mathbb{R}^n \to \mathbb{R}^n, g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ assumed to be Lipschitz-continuous vector valued functions.

Definition 1: **(Control Barrier Function)**: Let $h : \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable function and let $\mathcal{S} \subset \mathbb{R}^n$ be a non-empty set such that

$$\mathcal{S} = \{\boldsymbol{x} \in \mathbb{R}^n : h(\boldsymbol{x}) \geq 0\},$$

$$\partial\mathcal{S} = \{\boldsymbol{x} \in \mathbb{R}^n : h(\boldsymbol{x}) = 0\},$$

$$\text{Int}\left(\mathcal{S}\right) = \{\boldsymbol{x} \in \mathbb{R}^n : h(\boldsymbol{x}) > 0\}.$$

If every $\boldsymbol{x} \in \partial\mathcal{S}$ is a regular point[1] for h and there exists an extended class $\mathcal{K}$ function[2] α such that

$$\nabla h(\boldsymbol{x})^T f_0(\boldsymbol{x}) + \nabla h(\boldsymbol{x})^T g(\boldsymbol{x})\boldsymbol{u} \geq -\alpha\left(h(\boldsymbol{x})\right) \tag{5}$$

for all $\boldsymbol{x} \in \mathcal{S}$, then h is a **control barrier function**.

Definition 2: **(Forward Invariant Set)**: A set $\mathcal{S} \subseteq \mathbb{R}^n$ is forward invariant for a control system if, for any initial state $\boldsymbol{x}_0 \in \mathcal{S}$, the state of the system remains in $\mathcal{S}$ for every $t \geq 0$.

From [25], CBFs ensure that the control input keeps the system (4) within the safety set $\mathcal{S}$ defined by the constraint function $h(\boldsymbol{x})$ for every initial state $\boldsymbol{x}_0 \in \mathcal{S}$. In other words, they give necessary and sufficient conditions such that $\mathcal{S}$ is *forward invariant* for the control system.

Theorem 2: [24] Let $h : \mathbb{R}^n \to \mathbb{R}$ be a CBF and let $\mathcal{S}$ be the associated safety set. For any Lipschitz-continuous control that satisfies the following constraint

$$\dot{h}(\boldsymbol{x}, \boldsymbol{u}) = \nabla h(\boldsymbol{x})^T f_0(\boldsymbol{x}) + \nabla h(\boldsymbol{x})^T g(\boldsymbol{x})\boldsymbol{u} \geq -\alpha\left(h(\boldsymbol{x})\right), \tag{6}$$

$\mathcal{S}$ is forward invariant for system (4), hence safe.

The constraint (5) is affine in $\boldsymbol{u}$, then (6) can be exploited to defined the following a quadratic programming problem

$$\boldsymbol{u}^* = \underset{\boldsymbol{u} \in \mathbb{R}^m}{\arg\min} ||\boldsymbol{u} - \boldsymbol{u}_{des}||^2,$$

subject to

$$\nabla h(\boldsymbol{x})^T f_0(\boldsymbol{x}) + \nabla h(\boldsymbol{x})^T g(\boldsymbol{x})\boldsymbol{u} \geq -\alpha\left(h(\boldsymbol{x})\right), \tag{7}$$

which has an *explicit solution*, given by $\boldsymbol{u}^* = \boldsymbol{u}_{des} + \boldsymbol{u}_{safe}$, where $\boldsymbol{u}_{des}$ is the control input satisfying the unconstrained optimisation problem, and $\boldsymbol{u}_{safe}$ is the safety input

$$\boldsymbol{u}_{safe} = \begin{cases} -\frac{L_g h(\boldsymbol{x})^T}{L_g h(\boldsymbol{x}) L_g h(\boldsymbol{x})^T} \Psi(\boldsymbol{x}, t, \boldsymbol{u}_{des}), & \text{if } \Psi < 0 \\ 0, & \text{if } \Psi \geq 0 \end{cases} \tag{8}$$

with $\Psi(\boldsymbol{x}, t, \boldsymbol{u}_{des})$ given by

$$\Psi(\boldsymbol{x}, t, \boldsymbol{u}_{des}) \triangleq L_f h(\boldsymbol{x}) + L_g h(\boldsymbol{x})\boldsymbol{u}_{des} + \alpha\left(h(\boldsymbol{x})\right), \tag{9}$$

where $L_g h(\boldsymbol{x}) = \nabla h(\boldsymbol{x})^T g(\boldsymbol{x})$ and $L_f h(\boldsymbol{x}) = \nabla h(\boldsymbol{x})^T f_0(\boldsymbol{x})$.

[1] A point $\boldsymbol{x}_0$ is a regular point for a function h if $\nabla h(\boldsymbol{x}_0) \neq 0$.

[2] An extended class $\mathcal{K}$ function is a strictly increasing function $\alpha : \mathbb{R} \to \mathbb{R}$ such that $\alpha(0) = 0$.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

III. Methodology

As mentioned in Section II-B, CBFs require a system expressed in control-affine form. In this section, we then express the DMPs in control-affine form and derive the theory behind the unified DMP-CBF framework.

A. DMPs in Control-Affine Form

Let $\boldsymbol{x} = (p, v)$ be the state a planar system, where $p = (x, y) \in \mathbb{R}^2$ is the Cartesian position and $v = (v_x, v_y) \in \mathbb{R}^2$ is the Cartesian velocity. DMPs (1a)–(1b) can be expressed in the form (4) as follows

$$f_0(\boldsymbol{x}) = \frac{1}{\tau}\begin{bmatrix} v_x \\ v_y \\ K(x_g - x) - Dv_x \\ K(y_g - y) - Dv_y \end{bmatrix}, \tag{10a}$$

$$g(\boldsymbol{x}) = \frac{1}{\tau}\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T, \tag{10b}$$

$$\boldsymbol{u}(s) = \frac{1}{\tau}\begin{bmatrix} K_1 f_1(s) - K_1(x_g - x_0)s \\ K_2 f_2(s) - K_2(y_g - y_0)s \end{bmatrix}, \tag{10c}$$

where $\boldsymbol{u}(s) \in \mathbb{R}^2$ is the control vector containing the learnt DMP forcing term $f(s)$ and $K(p_g - p_0)$ is the goal attractor. Since s represents a re-parametrisation of time and is governed by the canonical system (2), $\boldsymbol{u}$ is ultimately a function of time.

B. Constrained Movement Primitives

We can now integrate CBFs into DMPs to get safe CMPs. Assume that a CBF $h(\boldsymbol{x})$ is defined. A safety input $\boldsymbol{u}_{\text{safe}}$ can be computed explicitly as in (8). The following theorem proves that the new DMP system driven by the original DMP input and the safety input converges to p_g while respecting $h(\boldsymbol{x})$.

Theorem 3: **(Safety and Convergence of CMPs)** Let $\boldsymbol{x}_0$ and $\boldsymbol{x}_g$ be the initial and final states, respectively, and $h(\boldsymbol{x})$ be a CBF (with associated forward-invariant set $\mathcal{S}$). The DMP system described by (10a)–(10b) and controlled by $\boldsymbol{u} = \boldsymbol{u}_{\text{safe}} + \boldsymbol{u}_{\text{des}}$, with $\boldsymbol{u}_{\text{safe}}$ defined in (8) and $\boldsymbol{u}_{\text{des}}$ defined in (10c), is safe with respect to $h(\boldsymbol{x})$ and goal-convergent iff ("$\iff$") $\boldsymbol{x}_0$ and $\boldsymbol{x}_g$ are safe, i.e. $\boldsymbol{x}_0, \boldsymbol{x}_g \in \mathcal{S}$.

Proof: "$\Longrightarrow$" If the control system is safe with respect to $h(\boldsymbol{x})$, this implies that $\mathcal{S}$ is forward invariant. As a result, $\boldsymbol{x}_0$ belongs to $\mathcal{S}$ by Definition 2. Moreover, since Theorem 1 guarantees the convergence of DMPs, it follows that $\boldsymbol{x}_g$ is also contained within $\mathcal{S}$.

"$\Longleftarrow$" Assume by contradiction that $\boldsymbol{x}_0$ and $\boldsymbol{x}_g$ do not belong to $\mathcal{S}$. By hypothesis, the DMP is governed by $\boldsymbol{u}_{safe} + \boldsymbol{u}_{des}$, which from [25] satisfies

$$\nabla h(\boldsymbol{x})^T f_0(\boldsymbol{x}) + \nabla h(\boldsymbol{x})^T g(\boldsymbol{x}) u \geq -\alpha\left(h(\boldsymbol{x})\right).$$

Therefore, according to Theorem 2, $\mathcal{S}$ is forward invariant and this leads to a contradiction: $\mathcal{S}$ must contain $\boldsymbol{x}_0$. Moreover, forward invariance implies that the control system evolves within $\mathcal{S}$ for every $t \geq 0$, and DMPs converge to $\boldsymbol{x}_g$ from Theorem 1. Hence, also $\boldsymbol{x}_g \in \mathcal{S}$. □

Remark 1: Theorem 3 yields the safe control action (8) only in the case of *one single CBF*. However, we show in Section IV-C that multiple CBFs can be considered in our framework for specific cases, e.g., obstacle avoidance.

To illustrate our methodology, we present the following example, where the goal is to constrain the maximum Cartesian velocity of the trajectory generated by the DMP. It is worth mentioning that this represents a *nonlinear constraint* in Cartesian coordinates.

Example 1: **(Maximum Cartesian velocity)** The constraint on the maximum velocity is formalised as

$$h(\boldsymbol{x}) = v_{\max} - \sqrt{v_x^2 + v_y^2} \geq 0.$$

Since $\nabla h(\boldsymbol{x})$ has a discontinuity for $(v_x, v_y) = 0$, $h(\boldsymbol{x})$ is not a C^1 function. Hence, to apply Theorem 1, we need the following approximation[3]:

$$h_v(\boldsymbol{x}) = v_{\max} - \sqrt{v_x^2 + v_y^2 + \epsilon_v} \tag{11}$$

with $\epsilon_v > 0$. The gradient ∇h_v is then continuous and equal to zero only at the origin of the state space, which does not belong to the level set

$$\partial\mathcal{S}_v = \{\boldsymbol{x} \in \mathbb{R}^4 \,|\, h_v(\boldsymbol{x}) = 0\}$$

so Definition 1 is satisfied.

We can then compute $\boldsymbol{u}_{safe}$ from (8)–(9), with $L_g h$ and $L_f h$ given by

$$L_g h(\boldsymbol{x}) = -\left[\frac{v_x}{\sqrt{v_x^2 + v_y^2 + \epsilon_v}}, \frac{v_y}{\sqrt{v_x^2 + v_y^2 + \epsilon_v}}\right],$$

$$L_f h(\boldsymbol{x}) = -\frac{K(x_g - x)v_x - Dv_x^2}{\sqrt{v_x^2 + v_y^2 + \epsilon_v}} - \frac{K(y_g - y)v_y - Dv_y^2}{\sqrt{v_x^2 + v_y^2 + \epsilon_v}}.$$

IV. Experimental Results

In this section, we validate our methodology on three constraints of increasing complexity designed for DMPs controlling a mobile robot. Specifically, the following constraints are defined:

- *maximum Cartesian velocity*, as in Example 1;
- *obstacle avoidance* in the presence of dynamic obstacles;
- *maximum centrifugal acceleration*, which represents a *dynamic constraint* to avoid slippery during navigation.

We have extended the library of DMPs[4] provided by [4].

A. DMP Training

For all experiments, we learnt the forcing term $f(s)$ of the DMP (1a)–(1b) from an elliptical trajectory parametrized as

$$\begin{cases} x(t) = a_0 \cos(t) \\ y(t) = a_1 \sin(t) \end{cases} \quad t \in [0, \pi], \tag{12}$$

[3]It always holds $h_v(\boldsymbol{x}) > h(\boldsymbol{x})$, hence (11) is a conservative over-approximation of the original CBF, which preserves safety.

[4]Link to source code: https://github.com/Vese91/DD-Robot-DMPs

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

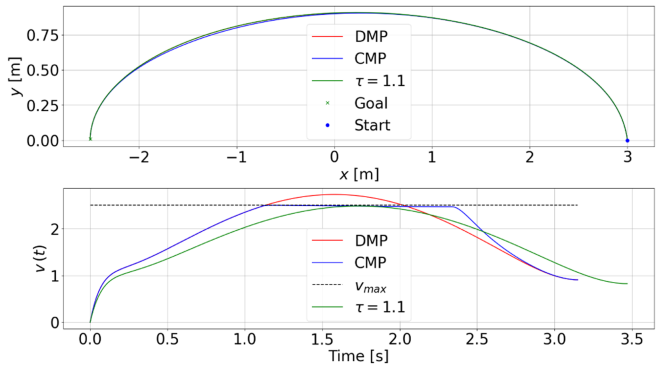

Fig. 1. Trajectories generated using DMPs and CMPs, incorporating the constraint on *maximum velocity* (11). The DMP trajectory with a different τ for temporal re-scaling is also included.

where $a_0 = 1.0$, $a_1 = 3.0$ and the number of samples is equal to $N = 1000$.

Unless otherwise specified, during the *execution* phase, initial and goal positions are set to $p_0 = (3.0,\ 0.0)\,\mathrm{m}$ and $p_g = (-2.5,\ 0.0)\,\mathrm{m}$. In every experimental setup, for the control law (8) we have chosen the straight line $\alpha(h(\boldsymbol{x})) = \alpha h(\boldsymbol{x})$ as extended class $\mathcal{K}$ function, with $\alpha = 50$.

B. Maximum Cartesian Velocity

Fig. 1 (top) illustrates the trajectory executed by the DMPs both with and without the influence of the CBF in (11). From the bottom plot, we notice that the CBF enforces a maximum speed constraint on the trajectory, limiting it to $v_{\max} = 2.50$ m/s (blue line), whereas no such limitation is present without the CBF (red line). It is worth noting that the execution speed of a DMP can be reduced by simply adjusting the parameter τ of the canonical system in (2). In fact, the speed profile obtained without CBFs, using $\tau = 1.10$ (green line), satisfies the constraint $v_{\max}$. However, in this case, the speed is scaled down *uniformly along the whole trajectory*, leading to higher execution time (from $t = 3.14$ s to $t = 3.50$ s). In contrast, CBFs ensure compliance with the velocity constraint while preserving the execution time by reducing the velocity only when needed, as long as the specified maximum velocity is not excessively low.

C. Obstacle Avoidance

Repulsive potential fields for static obstacles in trajectory planning have already been embedded in CBFs [24]. However, as evidenced by recent literature on DMPs [5], it is necessary to extend the framework to *dynamic* (moving) obstacles. To this aim, we consider an obstacle described by position $p_o = (x_o, y_o)$ and velocity $v_o = (v_{o,x}, v_{o,y})$. We introduce, for the i-th obstacle, the repulsive dynamic potential defined by [26]

$$U_{rep,i}(p, v) = \eta \left[\frac{1}{r_i(p, v)} - \frac{1}{r_{\min}} \right], \tag{13}$$

where $\eta > 0$ is a scaling factor and $r_{\min}$ is the radius over which the repulsive field is active. $r_i(p, v)$ is defined as

$$r_i(p, v) = r_i(p) - \frac{v_{ro,i}^2}{2\gamma}, \tag{14}$$

with $\gamma > 0$, and

$$r_i(p) = \sqrt{(x - x_{o,i})^2 + (y - y_{o,i})^2}$$

with

$$v_{ro,i} = (v - v_{o,i}) \cdot \boldsymbol{n}_{ro,i},$$

where $\boldsymbol{n}_{ro,i}$ is unit vector from the robot to the i-th obstacle.

Similarly to [24], for $n_o \in \mathbb{N}$ obstacles, we define

$$U_{rep}(p, v) = \sum_{i=1}^{n_o} U_{rep,i}(p, v) \tag{15}$$

and the candidate CBF as

$$h_o(\boldsymbol{x}) = \frac{1}{1 + \sum_{i=1}^{n_o} U_{rep,i}(p, v)} - \delta_0, \tag{16}$$

where $\delta_0 > 0$ is a small constant. The gradient of (16) is

$$\nabla h_o(\boldsymbol{x}) = -\frac{1}{(1 + \sum_{i=1}^{n_o} U_{rep,i}(p, v))^2} \sum_{i=1}^{n_o} \nabla U_{rep,i}(p, v).$$

We have that $\partial\mathcal{S}_o$ is given by

$$\partial\mathcal{S}_o = \left\{ \boldsymbol{x} \in \mathbb{R}^4 \ :\ U_{rep}(p, v) = \frac{1 - \delta_0}{\delta_0} \right\},$$

and the value of $\nabla h_o(\boldsymbol{x})$ for $\boldsymbol{x} \in \partial\mathcal{S}_o$ is

$$\nabla h_o(\boldsymbol{x}) = -\delta_0^2 \nabla U_{rep}(p, v).$$

Since, by construction, the potential (13) is a strictly decreasing monotonic function as $r(p, v)$ increases, and it tends to infinity as $r(p, v)$ approaches zero, it does not have critical points. Therefore, its gradient is never zero on $\partial\mathcal{S}_o$. The control law $\boldsymbol{u}_{safe}$ is calculated as in the case discussed in Example 1, we omit the basic mathematics for space constraints.

To model more realistic volumetric obstacles, we represent them as point obstacles along the surface. Specifically, we compare the blue trajectory generated with CBFs with the red one generated by a DMP with an additional forcing term for obstacle avoidance, $f_{obs}(p, v)$, as in (3). The expression of $f_{obs}(p, v)$ is taken from [26]:

$$f_{obs}(p, v) = -\left[\frac{\eta}{r^2(p, v)} \left(1 + \frac{v_{ro}}{\gamma} \right) \quad \frac{\eta v_{ro} v_{ro\perp}}{\gamma r(p) r^2(p, v)} \right],$$

where

$$v_{ro\perp} = \sqrt{v^2 - v_{ro}^2}.$$

The experiments are run with the best empirically tuned parameters $\gamma = 100$ m/s^2, $\delta_0 = 0.05$, $r_{\min} = 0.25$ m and $\eta = 0.05$.

Simulation results are shown in Fig. 2, where the robot must follow the elliptical reference trajectory, while constrained by surrounding fixed obstacles (black dots). This replicates a challenging, narrow corridor scenario. Furthermore, two obstacles

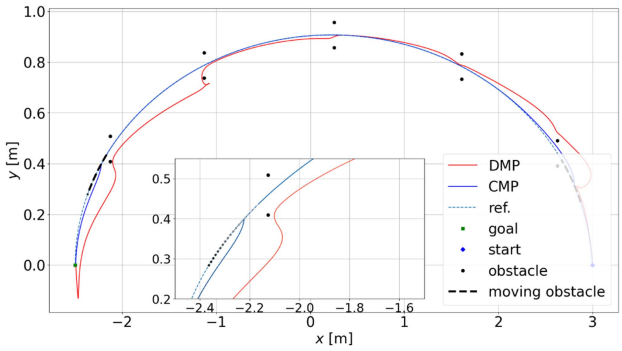

Fig. 2. Trajectories generated using DMPs and CMPs, including the *obstacle avoidance* constraint (16) in a cluttered environment scenario. Nested plot: We used color-scaled lines to show the motion of the moving obstacles.

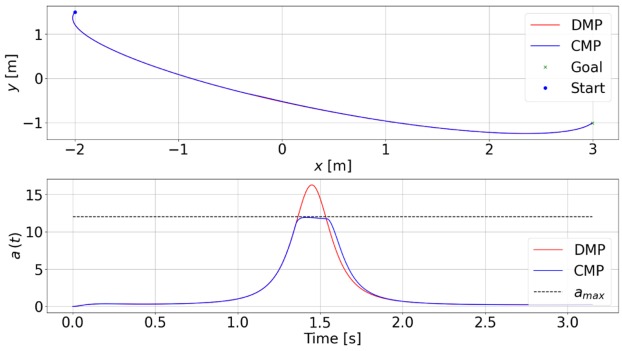

Fig. 3. Trajectories generated using DMPs and CMPs, incorporating the constraint on *maximum centrifugal acceleration* (18).

(black dashed lines) move forth and back along the reference trajectory, thus significantly obstructing the motion of the robot and making the scenario more challenging and cluttered. CMPs (continuous red curve) avoid obstacles by staying closer to the reference trajectory (dashed green curve) than the original DMPs. Indeed, the root mean squared error (RMSE) for the CMP is 0.015 ± 0.013 m, while for the DMP it is 1.304 ± 0.737 m.

Notice that there may be cases where the gradient $\nabla U_{rep,i}(p, v)$ of (13) is zero (e.g., when the position of the controlled system is in between two identical obstacles), thus nullifying $\nabla h_0(\boldsymbol{x})$. In such exceptional cases, the convergence to the goal position is still guaranteed by Theorem 3: if the gradient $\nabla U_{rep}(p, v)$ of (15) is equal to zero, then $\nabla h_o(\boldsymbol{x}) = 0$ and $\boldsymbol{u}^* = \boldsymbol{u}(s)$, where $\boldsymbol{u}(s)$ is given by (10c).

D. Maximum Centrifugal Acceleration

We now consider the problem of avoiding the slipperiness of a mobile robot when moving along curved trajectories too fast. This results in a *dynamic constraint* on the acceleration profile of the robot.

More specifically, during an elliptical trajectory (e.g., described by a DMP), the centrifugal acceleration a_c acting on the robot expressed in polar coordinates is $a_c = \rho\omega^2$ where $\rho > 0$ is the radius, and $\omega \in \mathbb{R}$ is the angular velocity.

Since DMPs are expressed in Cartesian coordinates, we need to use the following transformations

$$\rho = \sqrt{x^2 + y^2}, \quad \omega = \frac{(xv_y - yv_x)^2}{(x^2 + y^2)^2},$$

obtained from the polar coordinates, to rewrite the centrifugal acceleration as

$$a_c = \frac{(xv_y - yv_x)^4}{\sqrt{(x^2 + y^2)^3}}. \tag{17}$$

To avoid slippery, it must hold $h(\boldsymbol{x}) = a_{\max} - a_c > 0$, where $a_{\max}$ is the maximum allowed centrifugal acceleration[5]. Since a_c in (17) and hence $h(\boldsymbol{x})$ are not continuously differentiable, we introduce a similar over-approximation as in Example 1, as follows

$$h_c(\boldsymbol{x}) = a_{\max} - \sqrt{\frac{(v_y x - v_x y)^4}{(x^2 + y^2)^3} + \epsilon_c}, \tag{18}$$

with $\epsilon_c > 0$. In this case, to study the gradient ∇h_c, it is convenient to work in polar coordinates, where we have

$$h_c(\rho, \omega) = a_{\max} - \sqrt{\rho^2\omega^4 + \epsilon_c},$$

$$\nabla h_c(\rho, \omega) = \left(-\frac{\rho\omega^4}{\sqrt{\rho^2\omega^4 + \epsilon_c}}, -\frac{2\rho^2\omega^3}{\sqrt{\rho^2\omega^4 + \epsilon_c}} \right).$$

$\nabla h_c(\rho, \omega) = (0, 0)$ only for points in the form $(\rho, 0)$. The boundary of the safety set $\mathcal{S}$ is

$$\partial\mathcal{S}_c = \left\{ (\rho, \omega) \in \mathbb{R}_+ \times \mathbb{R} \,:\, \rho = \frac{\sqrt{a_{\max}^2 - \epsilon_c}}{\omega^2} \right\},$$

and points in the form $(\rho, 0)$ do not belong to it. We also have that $\nabla h_c(\rho, \omega) \neq (0, 0)$ for every $x \in \partial\mathcal{S}_c$, thus Definition 1 is satisfied. The control law $\boldsymbol{u}_{safe}$ is calculated as in the case discussed in Example 1; we omit the explicit expression for space constraints.

Fig. 3 (top) shows the trajectory obtained via DMPs during the execution phase with and without the control barrier function in (18). Although the trajectories appear to be overlapping, the constraint profiles in Fig. 3 (bottom) show that the action of the CBF limits the centrifugal acceleration to the maximum value given by $a_{\max} = 12\,\mathrm{m/s^2}$.

E. Validation on the Real Robot

We test our methodology on a custom-built mobile robot controlled via a ROS2 interface. We replicate the three CMPs (with parameters) presented in the previous sections, the video is attached to this submission. In particular, for the *obstacle avoidance constraint*, we consider a volumetric static circular obstacle (radius $r \approx 0.08$m) placed along the robot's trajectory. We enlarge r by the half-dimension of the robot (≈ 0.07m, hence total radius $r_{\text{tot}} \approx 0.15$m), in order to account for its shape. We then perform a study by varying the parameter $\alpha \in [10, 50]$ (the CBF coefficient).

Fig. 4 confirms the results of Fig. 2: CMPs is closer to the original elliptical DMP trajectory (green dashed curve).

[5]Typically, for grounded robots, $a_{\max} = \mu_s g$, where μ_s is the static friction coefficient and g is the gravity acceleration.

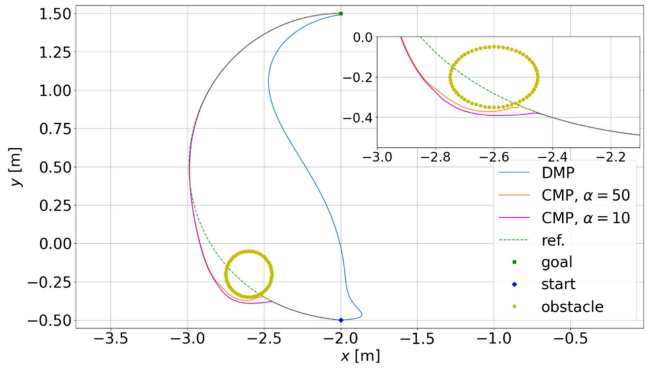

Fig. 4. Trajectories generated using DMPs and CMPs for various values of α, including the *obstacle avoidance* constraint (16), collected on a real robot.

Moreover, the value of α can be used to tune the smoothness of the trajectory and regulate the proximity to obstacles. In the attached video, we set $\alpha = 10$ to replicate the smoother, more conservative trajectory case.

V. Conclusion

We presented the Constrained Movement Primitives (CMPs), the first framework that integrates Dynamic Movement Primitives (DMPs) with Control Barrier Functions (CBFs) for the generation of robotic trajectories satisfying nonlinear constraints. We provided theoretical guarantees on the convergence of DMPs to the goal position under CBF constraints. Moreover, a key advantage of CMPs is that the CBF-based control law can be efficiently computed in closed form. We tested three nonlinear kinodynamic constraints of increasing complexity for mobile robots: (i) maximum Cartesian speed, (ii) obstacle avoidance, and (iii) maximum centrifugal acceleration to avoid slipping on curved trajectories. Our experiments in simulation and on a real robot show that CMPs are superior to classical methods for obstacle avoidance and velocity limitation with DMPs, generating trajectories that better preserve the shape of the learning trajectory while meeting the desired constraints.

In future research, we will extend the framework to three-dimensional motion in presence of dynamic and impredictable obstacles, and to combine different CBFs in a single DMP (currently possible only in restricted cases [27]) and to account for uncertainty in robotic applications, e.g., by combining CBFs with probabilistic DMPs.

References

[1] S. Schaal, "Dynamic movement primitives-A framework for motor control in humans and humanoid robotics," in *Adaptive Motion of Animals and Machines*, Tokyo, Japan: Springer, 2006, pp. 261–280.

[2] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proc. 2002 IEEE Int. Conf. Robot. Automat.*, 2002, vol. 2, pp. 1398–1403.

[3] D.-H. Park, H. Hoffmann, P. Pastor, and S. Schaal, "Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields," in *Proc. 8th IEEE-RAS Int. Conf. Humanoid Robots*, 2008, pp. 91–98.

[4] M. Ginesi, N. Sansonetto, and P. Fiorini, "Overcoming some drawbacks of dynamic movement primitives," *Robot. Auton. Syst.*, vol. 144, 2021, Art. no. 103844.

[5] M. Ginesi, D. Meli, A. Roberti, N. Sansonetto, and P. Fiorini, "Dynamic movement primitives: Volumetric obstacle avoidance using dynamic potential functions," *J. Intell. Robotic Syst.*, vol. 101, 2021, Art. no. 79.

[6] E. Tagliabue, D. Meli, D. Dall'Alba, and P. Fiorini, "Deliberation in autonomous robotic surgery: A framework for handling anatomical uncertainty," in *Proc. 2022 Int. Conf. Robot. Automat.*, 2022, pp. 11080–11086.

[7] R. Huang, H. Cheng, H. Guo, Q. Chen, and X. Lin, "Hierarchical interactive learning for a human-powered augmentation lower exoskeleton," in *Proc. 2016 IEEE Int. Conf. Robot. Automat.*, 2016, pp. 257–263.

[8] M. Ginesi, D. Meli, A. Roberti, N. Sansonetto, and P. Fiorini, "Autonomous task planning and situation awareness in robotic surgery," in *Proc. 2020 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 3144–3150.

[9] J. Palmieri, P. Di Lillo, M. Lippi, S. Chiaverini, and A. Marino, "A control architecture for safe trajectory generation in human–robot collaborative settings," *IEEE Trans. Automat. Sci. Eng.*, vol. 22, pp. 365–380, 2025.

[10] W. Ding, C. Xu, M. Arief, H. Lin, B. Li, and D. Zhao, "A survey on safety-critical driving scenario generation—A methodological perspective," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 7, pp. 6971–6988, Jul. 2023.

[11] A. Sidiropoulos, D. Papageorgiou, and Z. Doulgeri, "A novel framework for generalizing dynamic movement primitives under kinematic constraints," *Auton. Robots*, vol. 47, no. 1, pp. 37–50, 2023.

[12] A. Dahlin and Y. Karayiannidis, "Temporal coupling of dynamical movement primitives for constrained velocities and accelerations," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 2233–2239, Apr. 2021.

[13] P. Pastor et al., "From dynamic movement primitives to associative skill memories," *Robot. Auton. Syst.*, vol. 61, no. 4, pp. 351–361, 2013.

[14] L. Han, H. Yuan, W. Xu, and Y. Huang, "Modified dynamic movement primitives: Robot trajectory planning and force control under curved surface constraints," *IEEE Trans. Cybern.*, vol. 53, no. 7, pp. 4245–4258, Jul. 2023.

[15] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel, "Dynamic movement primitives in robotics: A tutorial survey," *Int. J. Robot. Res.*, vol. 42, no. 13, pp. 1133–1184, 2023.

[16] F. Frank, A. Paraschos, P. van der Smagt, and B. Cseke, "Constrained probabilistic movement primitives for robot trajectory adaptation," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2276–2294, Aug. 2022.

[17] Z. Wang, X. Zhou, C. Xu, and F. Gao, "Geometrically constrained trajectory optimization for multicopters," *IEEE Trans. Robot.*, vol. 38, no. 5, pp. 3259–3278, Oct. 2022.

[18] L. Dong, Z. He, C. Song, and C. Sun, "A review of mobile robot motion planning methods: From classical motion planning workflows to reinforcement learning-based architectures," *J. Syst. Eng. Electron.*, vol. 34, no. 2, pp. 439–459, 2023.

[19] Z. Zhao et al., "A survey of optimization-based task and motion planning: From classical to learning approaches," *IEEE/ASME Trans. Mechatron.*, pp. 1–27, 2024, doi: 10.1109/TMECH.2024.3452509.

[20] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *Proc. 53rd IEEE Conf. Decis. Control*, 2014, pp. 6271–6278.

[21] C. Dawson, Z. Qin, S. Gao, and C. Fan, "Safe nonlinear control using robust neural Lyapunov-Barrier functions," in *Proc. 5th Conf. Robot Learn.*, 2022, pp. 1724–1735.

[22] Z. Wang, T. Hu, and L. Long, "Multi-UAV safe collaborative transportation based on adaptive control barrier function," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 53, no. 11, pp. 6975–6983, Nov. 2023.

[23] L. Markus, "Asymptotically autonomous differential systems," in *Contributions of the Theory of Nonlinear Oscillations*, vol. 3. Princeton, NJ, USA: Princeton Univ. Press, 2016, pp. 17–30.

[24] A. Singletary, K. Klingebiel, J. Bourne, A. Browning, P. Tokumaru, and A. Ames, "Comparative analysis of control barrier functions and artificial potential fields for obstacle avoidance," in *Proc. 2021 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 8129–8136.

[25] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017.

[26] J.-F. Duhé, S. Victor, and P. Melchior, "Contributions on artificial potential field method for effective obstacle avoidance," *Fractional Calculus Appl. Anal.*, vol. 24, pp. 421–446, 2021.

[27] A. Isaly, M. Ghanbarpour, R. G. Sanfelice, and W. E. Dixon, "On the feasibility and continuity of feedback controllers defined by multiple control barrier functions," *IEEE Trans. Autom. Control*, vol. 69, no. 11, pp. 7326–7339, Nov. 2024.