Journal of the Franklin Institute 363 (2026) 108536

Contents lists available at ScienceDirect

Journal of the Franklin Institute

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

A Machine Learning approach to passivity-preserving safety-critical control

Arturo Maiani *, Federico Baldisseri, Antonio Pietrabissa

Department of Computer, Control and Management Engineering "Antonio Ruberti", Sapienza University of Rome, Via Ariosto 25, Rome, 00185, Italy

ARTICLE INFO

Keywords:
Control barrier functions
Machine learning
Neural networks
Passivity-preserving control
Safety-critical control
Intelligent systems

ABSTRACT

Passivity-Based Control enables to constructively synthetize controllers with guarantees on the stability of the overall system. Although safety-critical control based on Control Barrier Functions is able to synthetize controllers that effectively cope with state constraints, passivity and input bounds are not addressed in general. This paper proposes a novel model-based Machine Learning methodology aimed at synthesizing Control Barrier Functions such that passivity of the closed loop system is preserved under safety-critical control and input saturation. Numerical simulations on a cart-pole system and on a 2R robot validate the effectiveness of the proposed control strategy in terms of performances and passivity preservation.

1. Introduction

This paper proposes an approach for safety-critical control of passive systems based on Machine Learning (ML). A system is defined passive if it can store or dissipate energy without generating or adding energy internally. Passivity-based control (PBC) includes several Lyapunov-based techniques, aimed at designing closed-loop generalised storage functions to provide stability guarantees for the controlled system while addressing desired performance requirements [1]. PBC has several practical advantages:

- Passive systems tolerate modeling errors, uncertainties, and even time delays better than non-passive ones [2].
- PBC methods such as PBC-IDA [3] allow to satisfy performance requirements by shaping the desired energy function and assigning desired damping;
- Passive controllers represent a feasible solution to make closed-loop systems robustly stable in presence of unknown environmental interactions, a pivotal aspect in real-world safety-critical applications.
- Storage functions can be qualified as Lyapunov functions under weak conditions, providing stability guarantees: this ensures that energy flows can be monitored, bounded, and safely controlled.

Safety-critical control concerns how forward invariance of a safe subset of the state space is obtained through control barrier functions (CBFs). This control methodology has been investigated in different applications, such as multi-task control of switched systems [4] or spacecraft attitude control in presence of uncertainties [5]. The problem of using CBFs in PBC is that, given a closed loop system coupled with a nominal passive controller, the optimal safety-critical control action does not generally preserve passivity.

The remainder of the paper is organized as follows: Section 2 presents related works in the context of PBC and CBF, and reports the proposed innovations; Section 3 provides background on PBC and CBFs; Section 4 presents the formulation for the ML-based

* Corresponding author.
E-mail addresses: maiani@diag.uniroma1.it (A. Maiani), baldisseri@diag.uniroma1.it (F. Baldisseri), pietrabissa@diag.uniroma1.it (A. Pietrabissa).

https://doi.org/10.1016/j.jfranklin.2026.108536
Received 26 November 2024; Received in revised form 3 December 2025; Accepted 17 February 2026
Available online 19 February 2026

optimization aimed at finding the desired CBF; Section 5 validates the proposed approach by numerical simulations; Section 6 draws the conclusions and outlines future research directions.

2. Related works and proposed innovations

The concept of passivity-preserving safety-critical control, specifically tailored to mechanical systems, was introduced by [6], which puts in light the passivity-preserving behavior of the CBFs proposed in [7]. The latter are not derived based on an optimality criterion nor address input contraints. Conversely, inspired by works such as [8], where optimal CBFs are found through ML-based optimization, this paper provides a constructive method to find optimal passivity-preserving CBFs for any generic passive systems. Neural networks can be trained to approximate control barrier functions and Lyapunov functions, enabling the synthesis of controllers that maintain both safety and stability under arbitrary switching laws, even with limited data [9], but they often lack guarantees of smoothness or conservativeness, which may compromise passivity and input constraint handling. Recurrent neural networks can be used to model system dynamics for each mode with online learning of the models in real time, allowing the MPC to adapt to disturbances and mode changes while ensuring closed-loop stability and safety by accounting for model uncertainty [10]. A limitation of this approach is that online learning and real-time MPC can be computationally expensive, especially with nonlinear dynamics or long horizons. Control barrier functions can also be integrated with Reinforcement Learning to guarantee safety and stability [11,12]: the cost-to-go function is augmented with safety constraints, and iterative learning ensures the invariance of the safe set and optimal performance. Off-policy RL methods further enhance flexibility and safety, but safety guarantees typically only emerge asymptotically, and guaranteeing safety during learning is a challenging task. While not focusing on passivity, classical robust control approaches such as H_∞ control address performance under worst-case disturbances and model uncertainty [13,14]. In H_∞ control synthesis, the objective is to minimize the worst-case gain from exogenous inputs (e.g., disturbances) to regulated outputs, which guarantees robust stability and performance even in the presence of bounded uncertainties and external perturbations. H_∞ control can even guarantee passivity for robotic systems [15], voltage-source converters [16], and port-Hamiltonian systems [17] in general; however, such methods do not inherently incorporate safety constraints such as set invariance or input saturation.

Among the works that focus on integrating concepts from PBC and CBFs, in [18] a CBF based on the energy tank framework is implemented: the state of the CBF represents the difference between the energy stored in a tank and a desired minumum energy threshold; a safety filtering control action guarantees that the tank never depletes, hence preserving passivity. In [19], the control action for a multi-robot autonomous system is found by solving a Quadratic Programming (QP) problem; a passivity-based CBF is enforced as a constraint of the QP for each robot. Both [18] and [19] do not consider obstacle avoidance and input bounds.

In the context of CBFs, the adoption of data-driven approaches is growing due to their ability to handle high dimensional models and reduce computational costs, while matching the performance of classical controllers [20]. The work in [21] provides a data-driven approach to the synthesis of provably safe controllers and Lyapunov functions. In [8], an actor-critic method is defined to synthesize CBFs under input saturation. In [22], in addition to handling input saturation constraints, the class-$\mathcal{K}$ function defining the CBF is learned as well. Existing data-driven works focus either on (i) enforcing stability [21], (ii) guaranteeing safety under input saturation [22], or (iii) synthetizing Robust Control Lyapunov Barrier Functions without input bounds [20].

To the best of the authors' knowledge, while existing works have separately addressed input saturation (e.g., [8,22]) and safety constraints (e.g., [20,23]), this paper presents the first *unified* framework that simultaneously ensures (i) passivity preservation via CBFs, (ii) input saturation constraints through ML-based optimization, (iii) safety guarantees under both conditions. Unlike previous methods (e.g., [18,19]), the proposed approach explicitly integrates passivity preservation with obstacle avoidance and input bounds, which is a key innovation, in the field of PBC. This innovative framework based on ML-optimization is applicable to any input-affine nonlinear system.

3. Preliminaries

Consider a nonlinear input-affine system of the form:

$$\dot{x} = F(x, u) = f(x) + g(x)u, \tag{1}$$

with relative degree r, where $x \in X \subseteq \mathbb{R}^n$, $u \in U \subseteq \mathbb{R}^m$, with the functions $F : \mathbb{R}^{n+m} \to \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ continuously differentiable. In additon, f and the columns of g are smooth vector fields (i.e., C^∞). The system (1) is equipped with an output function $c : \mathbb{R}^{n+m} \to \mathbb{R}^m$ such that $y = c(x, u)$ is the measured system output, with the same dimension of the input, and c is a smooth mapping.

3.1. Passivity and passivity-based control

Definition 1 (Passivity [24]): the system 1 is passive if there exists a positive definite, continuously differentiable storage function $V : \mathbb{R}^n \to \mathbb{R}$ such that, for all x and u,:

$$\dot{V} = \frac{\partial V}{\partial x} F(x, u) \le u^T y. \tag{2}$$

If $\dot{V} = u^T y$, the system is called lossless.

Definition 2 (Passivity-Based Control (PBC)): the PBC objective for system (1) is to find a state feedback law:

$$u(x) = u_{des}(x) + v, \tag{3}$$

where v is an output feedback and $u_{des}(x)$ is such that $f_{cl}(x) = f(x) + g(x)u_{des}(x)$ is passive with respect to a closed-loop storage function $V_{cl}(x)$ and an input-output pair (v, y), such that the closed-loop system is in the form:

$$\dot{x} = f_{cl}(x) + g(x)v, \tag{4a}$$

$$y = g(x)^T \frac{\partial V_{cl}(x)}{\partial x}, \tag{4b}$$

where $u_{des}(x)$ is a generic passive state-feedback control law and a typical v takes the form $v = -Ky$, with K positive definite [1]. In PBC, by defining the natural dissipation of the passively controlled system as $d_p(x) := -L_{f_{cl}} V_{cl}(x)$, passivity reduces to $d_p(x) \geq 0$. The performance of the controlled system along a task depends on the choice of $V_{cl}(x)$ and $f_{cl}(x)$, whose computation, in general, requires solving matching PDEs. However, some specific cases can be addressed by means of energy balance design methods when V_{cl} is given by the sum of kinetic energy and a desired potential function.

3.2. Passivity preserving control barrier functions

Considering system (1), with relative degree $r = 1$, CBFs are used to ensure the forward invariance of the so-called *safe set* $S \in X$. The control objective is then to design a state feedback controller satisfying the condition:

$$x(t) \in S \quad \forall t > 0, \forall\, x(0) \in S. \tag{5}$$

The safe set S is defined as the superlevel set of a continuously differentiable function $\phi : \mathbb{R}^n \to \mathbb{R}$, as $S := \{x \in X : \ \phi(x) \geq 0\}$. The function $\phi(x)$ is a CBF on X if $\frac{\partial \phi(x)}{\partial x} \neq 0$, for all $x \in \partial S$, where $\partial S = \{x \in X : \ \phi(x) = 0\}$ is the frontier of the set S and it holds that $\sup_{u \in U} \left[L_f \phi(x) + L_g \phi(x) u\right] \geq 0$, for all $u \in U$.

Theorem 1 [23]: *let $\phi(x) \in C^1$ be a CBF on X for the system (1) and $L_g\phi(x) \neq 0\ \forall\ x \in X$. Any locally Lipschitz controller $u(x)$ such that $L_f \phi(x) + L_g \phi(x) u(x) \geq 0,\ \forall x \in \partial S$, provides forward invariance of the safe set S. Additionally, the set S is asymptotically stable on X.*

The controller synthesis induced by CBFs is referred to as *safety-critical control*, and the control law is found by solving the following QP problem:

$$\begin{aligned} u^*(x) &= \arg\min_{u \in U} \|u(x) - u_{\text{des}}(x)\|^2, \\ \text{s.t.} \quad & L_f \phi(x) + L_g \phi(x) u(x) \geq 0, \end{aligned} \tag{6}$$

where $u_{des}(x)$ is the first term of a PBC controller (3) satisfying Def. 2. QP 6 admits the closed form solution of the form:

$$u^*(x) = u_{\text{des}}(x) + u_{\text{sc}}(x), \tag{7}$$

where $u_{sc}(x)$ represents the safety-critical part, defined as

$$u_{\text{sc}}(x) = \begin{cases} -\frac{L_g\phi(x)^T}{||L_g\phi(x)||^2} \psi(x, u_{\text{des}}) & \text{if } \psi(x, u_{\text{des}}) < 0 \\ 0 & \text{if } \psi(x, u_{\text{des}}) \geq 0 \end{cases}, \tag{8}$$

where $\psi(x, u) := \dot{\phi}(x, u(x))$.

Theorem 2 [6]: *let $u(x) = u^*(x) + v$, where $u^*(x)$ is the safety-critical controller (7) induced by a CBF $\phi(x)$. The resultant closed-loop system (4) is passive with respect to $V_{cl}(x)$ and the input-output pair (v, y) if and only if $L_g \phi(x) \neq 0$ and the following inequality holds when $\psi(x; u_{des}) < 0$:*

$$-\frac{L_g V_{cl}(x) L_g \phi(x)^T}{||L_g \phi(x)||^2} \psi(x, u_{des}(x)) \leq d_p(x), \tag{9}$$

where $d_p(x) = -L_{f_{cl}} V_{cl}(x)$ is the natural dissipation of the controlled system.

If such a CBF $\phi(x)$ exists, it will be referred to as a Passivity Preserving Control Barrier Function (PP-CBF).

4. ML-based approach for passivity-preserving safety-critical control

In this Section, the closed loop system (4) is considered with the control law (3). The term v of (3) is hereafter neglected. The objective is to learn a CBF that is suitable to induce a safety-filtering control which preserves passivity and satisfies input saturation bounds.

4.1. Neural SC-PBC

The methodology presented in this work is inspired by [8], where a ML approach is used to synthesize CBFs for input constrained systems. Assume that state space constraints are represented by function $h(x)$ with relative degree $r > 1$ and differentiable r times, inducing the safe set $H = \{x \in X : h(x) \geq 0\}$. A preliminary CBF, $\rho(x)$, that does not consider input constraints and passivity, is defined as:

$$\rho(x) = \left[\prod_{i=1}^{r-1} \left(1 + \gamma_i \frac{d}{dt}\right)\right] h(x), \tag{10}$$

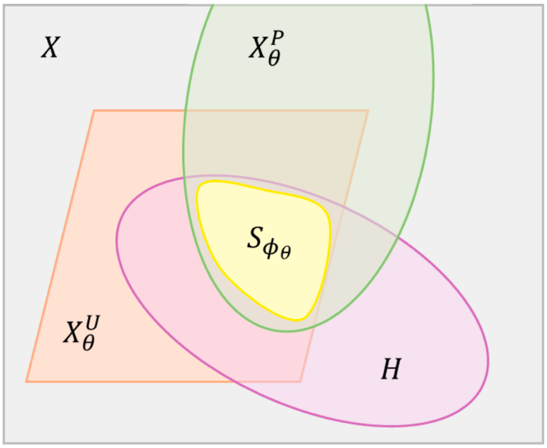

Fig. 1. Graphical depiction of problem (14).

with $\gamma_i \in \mathbb{R}_{>0}$, $i = 1, \dots, r-1$. Thanks to the assumptions on $h(x)$ and to the fact that the γ_i's are positive, it holds that the safe set $S_\rho = \{x \in X : \rho(x) \geq 0\}$ defines a positively invariant set in H [25]. In [8], the input constraints are tackled by subtracting a semi-positive definite term to the CBF (10), obtaining the modified neural CBF (NCBF):

$$\phi_\theta(x) = \rho(x) - K_\theta(x), \tag{11}$$

where $K_\theta(x)$ is the output of a neural network with parameters θ. No particular architecture for $K_\theta(x)$ is required, hereinafter considered as a Multi Layer Perceptron, except from $i)$ being composed of C^1 activation functions, to comply with Th. 1, and $ii)$ having a positive definite output layer, i.e., $K_\theta(x) \geq 0$. As proved in [25], condition $ii)$ guarantees that the safe set $S_{\phi_\theta} = \{x \in X : \phi_\theta(x) \geq 0\}$ defines a positively invariant set in H . Hereafter, according to the Universal Approximation Theorem in [26], we assume that $\phi_\theta(x)$ is chosen sufficiently wide and deep. Once the function $\phi_\theta(x)$ is found, a safety-critical control is defined according to Eqs. (7), (8):
Definition 3: the neural safety-critical control $u_\theta(x)$ is defined as:

$$u_\theta(x) = u_{des}(x) + u_{sc,\theta}(x), \tag{12}$$

with:

$$u_{\mathrm{sc},\theta}(x) = \begin{cases} -\frac{L_g\phi_\theta(x)^T}{||L_g\phi_\theta(x)||^2}\psi_\theta(x, u_{\mathrm{des}}) & \text{if } \psi_\theta(x, u_{\mathrm{des}}) < 0 \\ 0 & \text{if } \psi_\theta(x, u_{\mathrm{des}}) \geq 0 \end{cases}, \tag{13}$$

where $\psi_\theta(x, u) = \dot{\phi}_\theta(x, u)$. Hence, the problem corresponds to finding, if it exists, a function $K_\theta(x)$ such that the safety-critical controller (12)-(13) satisfies both the passivity condition (9) and the input bounds. The performance and feasibility of $u_\theta(x)$ depends on $\phi_\theta(x)$. The control law becomes less conservative as the safe set S_{ϕ_θ} grows, and the objective is to find the best parametrization θ to obtain the largest set S_{ϕ_θ}. The cost index $\mathcal{L}_\theta = -\lambda(S_{\phi_\theta})$ is then defined, where $\lambda(\cdot)$ represents a Lebesgue measure and the set S_{ϕ_θ} is assumed to be Lebesgue measurable – note that $\mathcal{L}_\theta$ is bounded as $S_{\phi_\theta} \subseteq H$. By considering such cost function, and that passivity and input bounds must be satisfied, we can state the OP for the determination of neural PP-CBFs and neural SC-PBC.
OP: *let the OP be defined as:*

$$\min_\theta \quad \mathcal{L}_\theta \tag{14a}$$

$$s.t. \quad A_\theta(x)\psi_\theta(x, u_{des}(x)) \leq d_p(x), \tag{14b}$$

$$u_\theta(x) \in U, \quad \forall x \in \partial S_{\phi_\theta}, \tag{14c}$$

where (14b) *enforces the passivity condition* (9) *with:*

$$A_\theta(x) = -\frac{L_g V_{cl}(x) L_g \phi_\theta(x)^T}{||L_g\phi_\theta(x)||^2}. \tag{15}$$

If the OP (14) is feasible, the optimal solution θ^* determines the function $K_{\theta*}(x)$ such that $\phi_{\theta*}(x)$ is a neural PP-CBF (PP-NCBF), the control law $u_{\theta*}(x)$ computed by Eq. (12) is a neural SC-PBC, and the set $S_{\phi_{\theta*}}$ is maximized. If the problem is not feasible, a SC-PBC in the form (12) does not exist.

Fig. 1 depicts the problem as the one of finding the optimal parametrization θ inducing the largest set S_{ϕ_θ} within the intersection of the sets H, where the state constraints are met, $X_\theta^P := \{x \in X \mid (14b)\}$, which is the region where safety critical control can be enforced passively, and $X_\theta^U := \{x \in X \mid (14c)\}$, where the input bounds constraints hold.

4.2. Deep learning optimization

The search for the optimal parametrization θ is cast as a supervised Deep Learning (DL) OP, due to its ability to: (i) approximate PP-CBFs in high-dimensional spaces via NN parameterization for OP (14) which does not admit a general closed form solution, (ii) encode constraints (14b-14c) directly into the training loss, and (iii) iteratively refine the safe set S_{ϕ_θ} through adaptive sampling. Unlike gradient-based solvers for QPs or Lyapunov methods, DL scales to complex dynamics without requiring convexity or closed-form solutions.

The cost function and constraints in (14) are relaxed to define a multi-task Lagrangian cost function, which is to be minimized iteratively by stochastic gradient descent (SGD). Concerning the cost function (14a), the evaluation of the volume of the set S_{ϕ_θ} cannot be computed in closed form as a function of θ. Hence, $\mathcal{L}_\theta$ must be replaced with a cost index which equivalently favours an enlargement of the set. Such index has been introduced in [8] as a *regularization term* and takes the form:

$$\mathcal{L}^r_\theta = \int_X \big(1 - \sigma(\phi_\theta(x)\big) dx, \tag{16}$$

where $\sigma(\cdot) : \mathbb{R} \to (0, 1)$ is the logistic sigmoid function defined as $\sigma(s) = \frac{1}{1+e^{-s}}$. By definition, it holds that $\sigma(\phi_\theta(x)) = 0.5$ if $x \in \partial S_{\phi_\theta}$, $\sigma(\phi_\theta(x)) > 0.5$ if $x \in S^{\circ}_{\phi_\theta}$, and $\sigma(\phi_\theta(x)) < 0.5$ if $x \notin S_{\phi_\theta}$. Since $\sigma(\cdot)$ is monotonically increasing, minimizing (16) corresponds to maximizing the values of $\phi_\theta(x)$ over X, thus increasing the Lebesgue measure of the region of X where $\sigma(\phi_\theta(x)) > 0.5$.
As we need to define a Lagrangian relaxation, the constraints of OP (14) are cast as two additional cost functions. The passivity constraint (14b) can be transformed into a cost index which becomes positive when (14b) is violated, i.e.:

$$\mathcal{L}^p_\theta(x) = \max\{0, A_\theta(x)\psi_\theta(x, u_{des}(x)) - d_p(x)\}, \tag{17}$$

(similarly to the Lyapunov-based loss function in [21]). The input constraint (14c) can be cast into the following cost index:

$$\mathcal{L}^u_\theta(x) = \inf_{u \in U} \big[d(u_\theta(x), u)\big], \tag{18}$$

where $d(u, v)$ denotes a distance measure between the vectors $u, v \in \mathbb{R}^m$. The relaxed unconstrained min-max OP is written as:

$$\min_\theta \left\{ \mathcal{L}^r_\theta + \max_{x \in S_{\phi_\theta}} \left[\lambda^p \mathcal{L}^p_\theta(x) + \lambda^u \mathcal{L}^u_\theta(x)\right] \right\}, \tag{19}$$

where λ^p and λ^u are weights, with sufficiently large values to let constraints (14b)-(14c) be verified: in this case, the max operator in (19) is such that the minimization seeks to drive the second and third terms to 0 for all $x \in S_{\phi_\theta}$.

4.3. Data sampling, critic function and loss function

The first term of the cost function (19) requires the evaluation of the integral of the cost index over the state space X, whereas the other terms are maximized over points belonging to ∂S_{ϕ_θ}, which is the boundary of the safe set. The data points on ∂S_{ϕ_θ} cannot be obtained as closed-form solutions of $\phi_\theta(x) = 0$, and, hence, similarly to [8], ∂S_{ϕ_θ} is approximated by means of Monte Carlo Sampling (MCS). The term in (16) is relaxed as the mean of the integrand function $1 - \sigma(\phi_\theta(x))$ over the sampled points. Hence, we need to find sample points for the evaluation of (19). To facilitate the task, we consider that it is not necessary to look for samples over the whole set X (for the first term of (19)) or on the exact boundary ∂S_{ϕ_θ} (for the other terms).

1) Since $S_{\phi_\theta} \subset H$, only the points belonging to H should be sampled, i.e., the points $x \in X$ such that $h(x) \geq 0$. Furthermore, only those points where safety-filtering control action is active should be selected through the condition:

$$\psi_\theta(x, u_{des}(x)) \leq 0. \tag{20}$$

2) Considering that the function $1 - \sigma(\phi_\theta(x))$ takes values mostly close to 0 in $S^{\circ}_{\phi_\theta}$, and values mostly close to 1 in $X \setminus S_{\phi_\theta}$, since on both these regions $\frac{d\sigma(s)}{ds}|_{\phi_\theta(x)}$ takes values close to zero, the contribution of such data points to the gradient of (19) is negligible. Therefore, the region where MCS is performed is chosen close to the boundary ∂S_{ϕ_θ}, in the set:

$$S^\epsilon_{\phi_\theta} = \{x \in X : \; 0 \leq \phi_\theta(x) \leq \epsilon\}, \tag{21}$$

where ϵ is a positive parameter so that $S^\epsilon_{\phi_\theta} \supset \partial S_{\phi_\theta}$. Moreover, since the probability of sampling data points exactly on the boundary ∂S_{ϕ_θ} is 0, the same points in $S^\epsilon_{\phi_\theta}$ will be used for the evaluation of the last two terms of (19).

Thus, the set where the samples are collected is

$$\tilde{S}^\varepsilon_{\phi_\theta} = \{x \in S^\epsilon_{\phi_\theta} | h(x) \geq 0, \psi_\theta(x, u_{des}(x)) \leq 0\}. \tag{22}$$

MCS-based optimization for high dimensional control problems would require a method to uniformly sample on $\tilde{S}^\varepsilon_{\phi_\theta}$; however, the use of such techniques goes beyond the scope of this work, and the simpler sampling strategy used in [8] is adopted.
Let N be the number of points in $\tilde{S}^\varepsilon_{\phi_\theta}$ sampled for the cost function evaluation, collected in the set D_θ. Such points will be used for the evaluation of the first term of (19), whereas the evaluation of the other two terms is performed over a subset of k points in D_θ.
Let the so-called *critic function* be defined as:

$$C_\theta(x) = \lambda^p \mathcal{L}^p_\theta(x) + \lambda^u \mathcal{L}^u_\theta(x). \tag{23}$$

The selected k points, collected in the set $\mathrm{Top}_k(C_\theta, D_\theta)$, are the ones for which $C_\theta(x)$ takes the highest values, i.e., the points representing the worst-case constraint violations. Finally, the loss function (needed for the NN training) is defined based on the cost function (19) and on the described sampling strategy. In practice, problem (19) is solved through SGD on the following loss function:

$$L(\theta) = \frac{1}{|D_\theta|} \sum_{x \in D_\theta} R_\theta(x) + \sum_{x \in \mathrm{Top}_k(C_\theta, D_\theta)} C_\theta(x), \tag{24}$$

where $R_\theta(x) = 1 - \sigma(\phi_\theta(x))$.

4.4. Training procedure

Algorithm 1 describes the pseudo-code detailing the training procedure. Preliminarily, the values of the learning rate l_r, the number N of samples to be selected in $\tilde{S}^{\epsilon}_{\phi_\theta}$, the number M and the threshold ξ used for the algorithm termination criterion must be selected. At each iteration i, the training set D^{train} is updated by sampling N states where Eqs. (21)-(20) are met, and the parameters

Algorithm 1 PP-NCBF training procedure.

1: Initialize θ and $D^{train} = \emptyset$
2: **for** $i = 1, 2, \dots, T$ **do**
3: Find P as a set of N samples in (22)
4: $D^{train} \leftarrow D^{train} \cup P$
5: **for** $k = 0, 1, 2, \dots, E-1$ **do**
6: $\theta \leftarrow \theta - l_r \nabla_\theta [L(\theta, D^{train})]$
7: $D^{train} \leftarrow \{x \in D^{train} | x \in \text{(22)}\}$
8: **end for**
9: Find D_θ as a set of N samples in (22)
10: Evaluate the loss function $L(\theta, D_\theta)$ (eq. (24))
11: Exit if $L(\theta, D_\theta) < \xi$ for M iterations over i.
12: **end for**

θ are updated via gradient descent over the selected states. This update phase lasts for E epochs: at each epoch, the parameters θ are updated and the states in D^{train} where Eqs. (21)-(20) are not satisfied for the updated parameters are excluded from the training. The parameters θ obtained by the last training epoch are then used to collect N new samples satisfying Eqs. (21)-(20) in the test set D_θ, and the set $\mathrm{Top}_k(C_\theta, D_\theta)$ is found by evaluating the states in D_θ with the critic function (23). Finally, the obtained parametrization θ is tested over D_θ by means of the cost function $L(\theta)$ defined in (24): the algorithm terminates if $L(\theta) < \xi$ for M iterations.

The set $\tilde{S}^{\epsilon}_{\phi_\theta}$ (22) (where the training data are sampled) varies with the parameters θ, causing the MCS data distribution to shift. This is a considerable issue in the context of training DL algorithms – see [27]. Here, two operations are proposed to mitigate such problem: i) to guarantee that sampled data distributions between consecutive iterations are similar, D^{train} is defined by the points belonging to the training set of the past iteration which still belong to (22), plus N new sampled states (lines 4–5 of the algorithm), and then D^{train} is clipped to a maximum allowed size equal to $2N$; ii) the learning rate l_r is chosen small, equal to $1e^{-5}$, so as to reduce changes in D^{train} within the training epochs.

Remark 1: the choice of carrying out the OP (19) on X instead of the whole $\mathbb{R}^n$ is motivated by computational reasons. Without taking additional measures, the drawback is that, if the system dynamics exits the sampling region X, stability and feasibility guarantees would not hold anymore since the neural network has not been trained on such region. A possible and simple solution, presented in [28], is to define the function $h(x)$ in (10) such that it is obtained by merging both the safety critical constraints and constraints enforcing the forward invariance of X.

Remark 2: convergence of the training process is addressed by (i) the strict monotonicity of $\sigma(\cdot)$ in $\mathcal{L}^r_\theta$, which drives $\phi_\theta(x)$ to expand S_{ϕ_θ}, (ii) the Lagrange multipliers λ^p, λ^u in (19), which force $\mathcal{L}^p_\theta$ and $\mathcal{L}^u_\theta$ to zero upon convergence, and (iii) the sampling strategy focusing on $\tilde{S}^{\epsilon}_{\phi_\theta}$, which ensures gradient updates are dominated by constraint-sensitive regions. While global optima cannot be guaranteed due to non-convexity, the experiments illustrated in Section 5 show consistent convergence to feasible PP-CBFs that satisfy all constraints even for non-convex problems, with $\mathcal{L}^p_\theta$ and $\mathcal{L}^u_\theta$ decaying to near-zero.

5. Simulations

The proposed methodology is validated in the domain of mechanical systems, with state $x = [q, p]$, where q and p represent the generalized coordinates and momentum respectively. In the context of mechanical systems, the NCBF (11), learned through Algorithm 1 takes the form:

$$\phi_\theta(q, p, u(q, p)) = \dot{h}(q, p, u(q, p)) + \gamma h(q) - K_\theta(q, p), \tag{25}$$

where $h(q)$ describes the geometrical constraints for the system. For what concerns the data sampling presented in 4.4, at each trial i the buffer D^{train} is refilled with $N = 1000$ newly sampled datapoints and augmented as in line 4 of Algorithm 1. A maximum size of 2000 samples is set for D^{train} . The chosen architecture for the network is a MLP with three layers of 100 units each, with

Exponential Linear activation Unit, a C^∞ function; the output layer is the square function. The parameters of the loss function (24) are $\lambda^u = \lambda^p = 10^2$, and a weight of 10^{-2} is used for standard L2 regularization. The chosed optimizer is Adam [29]. The exit condition is specified by $M = 100$ and $\xi = 1e^{-2}$.

Four types of CBFs are compared: (i) the CBF defined by (10), called Limit Blind CBF (LB-CBF), which does not take into account input bounds and passivity; (ii) the NCBF (25), taking into account input bounds but not passivity, as in [8]; (iii) the proposed PP-NCBF, still given by (25) but taking into account both input bounds and passivity; (iv) the PP-CBF developed in [6], based on the computation of the kinetic energy. The latter approach takes the form $\phi(q,p) = \alpha h(q) - K(q,p)$, where $\alpha \in \mathbb{R}^+$ and $K(q,p) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$ is the kinetic energy of the system. Since $K(q,p) \geq 0$, safety is preserved and the safe set S_ϕ defined by $\phi(q,p)$ is a subset of the safe set H defined by $h(q)$, with S_ϕ approaching H as $\alpha \to \infty$. Moreover, as in [7], the forward invariance condition in (6) is relaxed into $L_f\phi(q,p) + L_g\phi(q,p)u_{\text{des}}(q,p) > -\gamma_{\mathcal{K}}\phi$, where $\gamma_{\mathcal{K}}$ is a positive constant (*zeroing CBF* approach).

Remark 2: it is important to point out the difference in how passivity is enforced in our method compared to [6]. Passivity inequality (9) is satisfied in PP-CBF by guaranteeing that the left side of the inequality (9) is negative independently of $d_p(x)$. Conversely, PP-NCBF allows the left hand side of (9) to be positive, provided that it does not exceed the upper bound $d_p(x)$.

Two simulations are presented in the following sections, aimed at controlling a cart-pole system and a 2R vertical planar robot. Integration is performed with fourth order Runge-Kutta method and sampling time $T_s = 5e^{-4}$.

5.1. Cart-pole system

The cart-pole system presented in [6] is considered, with state space $q = [x \quad \beta]^T$, where x and β denote the cart position and the pole angle respectively. The task is to let the cart reach the desired position $x_d = 1$, with the pole in downwards position treated as a payload to be carried around. The cart has to be kept away from a wall placed at $x_w = 1.5$. The equation representing the geometry of the obstacle is then $h(x) = x_w - x$. As in [6], the cart-pole system and the nominal control do not contain dissipative terms. The passive control action is $u_{des,1}(x) = -k(x - x_d)$, with $k = 1$, and $u_{des,2}(\beta) = 0$. The expected behaviour is to have the cart position avoiding the wall and oscillating around $x = 1$, with the pendulum wobbling in response to the cart's movements. The parameters are $\gamma = 5$ for the CBFs (i)-(iii) and $\gamma_{\mathcal{K}} = 10$ and $\alpha = 100$ for (iv). The parameter γ in the CBF formulations determines the aggressiveness of the safety-critical control action. This choice must be guided by the need to balance responsiveness and stability. A larger value of γ ensures faster convergence to the safe set but may lead to overly conservative behavior or input saturation. The values chosen in the experiments are empirically validated to achieve a reasonable trade-off. The parameter $\gamma_{\mathcal{K}}$ is selected to ensure the zeroing CBF condition $L_f\phi(q,p) + L_g\phi(q,p)u_{\text{des}}(q,p) > -\gamma_{\mathcal{K}}\phi$ is satisfied without excessive conservatism. This aligns with the theoretical requirement that $\gamma_{\mathcal{K}}$ must be positive and sufficiently large to guarantee safety. The parameter α in the PP-CBF formulation $\phi(q,p) = \alpha h(q) - K(q,p)$ determines the relative weight of the safety constrain $h(q)$ (e.g., obstacle avoidance) versus the kinetic energy $K(q,p)$. A larger value of α expands the safe set $S_\phi = \{x \mid \phi(x) \geq 0\}$ by making $h(q)$ dominate over $K(q,p)$. Thus, α must be sufficiently large to ensure $S_\phi \approx H$, namely the desired safe set, but not so large that numerical instabilities arise. A conservative choice for the numerical value of α in ensures that $K(q,p)$ never overwhelms $h(q)$, which is critical for passive systems where energy growth must be bounded.

All the dynamic parameters of the cart-pole system are equal to 1. The desired storage function is $V_{cl}(q) = \mathcal{H}(q,\dot{q}) + \frac{1}{2}k(x - x_d)^2$, where $\mathcal{H}(q,\dot{q})$ is the open-loop Hamiltonian. The bounds on the overall control actions are $|u_i| < 25, i = 1,2$ for both the cart and pole. For the data-driven OP, the portion of interest of the state space is defined as $X = \{x \in [0,1.5], \beta \in [-\pi,\pi], \dot{x} \in [-4,4], \dot{\beta} \in [-4,4]\}$.

The simulations, shown in Fig. 2, are initialized from $(x,\beta,\dot{x},\dot{\beta}) = (0,0,3.5,0)$, i.e., with the cart approaching the wall at high speed. The input plots and the storage function plot show that, with LB-CBF, the control signal violates the input bounds and also passivity is strongly violated ($V_{cl}(t)$ steeply grows in the interval 0.5-1s); with NCBF, the input lies within the bounds but passivity is still violated; with the PP-CBF, input bounds are violated but passivity is preserved; with the proposed PP-NCBF, both passivity and control constraints are satisfied.

Note that i) the PP-CBF control exceeds the bounds as soon as the safety-critical control is activated and shows the fastest energy dissipation rate ($V_{cl}(t)$ is almost 0 at 0.5s); ii) the PP-NCBF yields a conservative behavior due to the approximations in the methodology (such as sampling over the neighborhood $S^{\epsilon}_{\phi_\theta}$ of the boundary ∂S_{ϕ_θ}) and keeps a safe distance from the wall.

5.2. 2R vertical planar robot

The proposed method is applied to a second passively controlled mechanical system, i.e., the PD plus gravity compensation control of a 2R vertical robot. The robot is composed of two links, each one 0.35m long and with mass equal to 5kg, and the gains for the PD terms in the controller are $K_p = 1$, $K_{\text{d}} = 6$. In the Euclidean plane (x_1, x_2), the obstacle is represented by a circle centered in $C = (0.6, 0.6)^T$ with radius $r = 0.2$m; hence, the equation representing the geometry of the obstacle is $h(x) = \sqrt{||p(x) - C||^2} - r$, where $p(x) \in \mathbb{R}^2$ is the position of the end effector in the vertical plane. The input bounds are $|u_i| < 50,\ i = 1,2$ for both control inputs.

The algorithms NCBF, PP-CBF and PP-NCBF are tested, with parameters $\gamma = 5$ for the neural CBFs and $\gamma_{\mathcal{K}} = 1$ and $\alpha = 100$ for the PP-CBF. The desired storage function is $V_{cl}(q) = K(q,\dot{q}) + \frac{1}{2}K_p||q - q_d||^2$, where q is the joint space and $K(q,\dot{q})$ is the kinetic energy. For the data-driven OP, the portion of interest of the state space is $X = \{q_1 \in [0,\pi/2], q_2 \in [-\pi/2,\pi/2], \dot{q}_1 \in [-0.1,1], \dot{q}_2 \in [-1,1]\}$. To compare the behavior of the zeroing CBF in [6], implemented on the interior of the safe set, versus the proposed approach, which acts only on its boundary, the safety-critical control action is enforced in both cases if the distance from the obstacle is equal or less than a threshold, chosen equal to $\epsilon = 5$ cm, where ϵ is the parameter in (21).

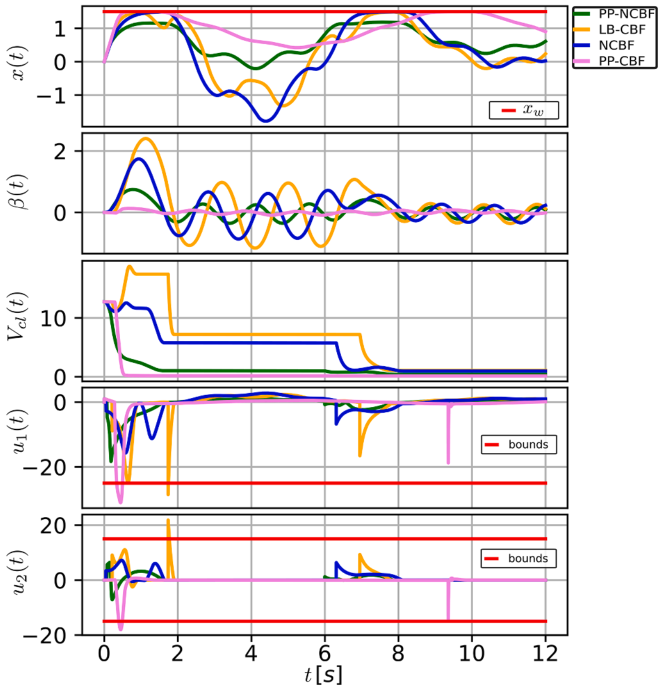

Fig. 2. Cart-pole simulations: a) position $x(t)$; b) pendulum angle; c) storage function $V_{cl}(t)$; d) control signal $u_1(t)$; e) control signal $u_2(t)$.

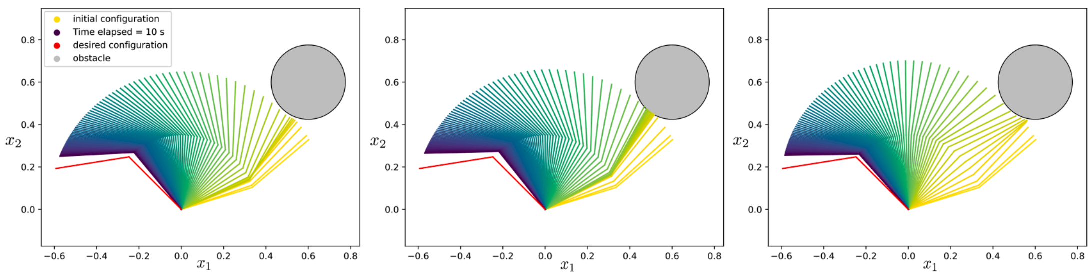

Fig. 3. Simulations on the 2R robot: PP-NCBF (left); PP-CBF (centre); NCBF (right).

The simulated robot trajectories, initialized from the state $(q_1, q_2, \dot{q}_1, \dot{q}_2) = (0.3, 0.4, 0, 0)$, are shown in Fig. 3. Fig. 4 shows the corresponding storage function plot, showing that PP-CBF and PP-NCBF preserve passivity ($V_{cl}(t)$ never grows), whereas the NCBF yields a non-passive behaviour. This is reflected by the different behaviour of the robot while approaching the obstacle. In the first two cases, the robot overcomes the obstacle without changing the sign of the angle between the 2 links (i.e., with $q_2 > 0$). Conversely, with the NCBF the robot approaches the obstacle by reducing q_2 down to a negative value: this effort causes an increase of the potential component of the energy storage function.

When the robot approaches the obstacle, PP-CBF and PP-NCBF slow down to avoid increasing the storage function: the time required to overcome the obstacle can be then computed as the time from the start to the end of the plateau in the storage function plots of Fig. 3. The plots show that, even if it preserves passivity, the time needed by PP-NCBF to overcome the obstacle is comparable to the time needed by NCBF. Conversely, the time needed by PP-CBF is about twice the time needed by PP-NCBF: as soon as PP-CBF enters the zone where $h(x) < r + \epsilon$, it rapidly dissipates kinetic energy and slows down almost completely the robot. The advantage of PP-NCBF is that the CBF is obtained by solving the OP (24), whereas PP-CBF does not perform any optimization.

A visualization of the learned PP-NCBF is shown in Fig. 7, representing the safe set in the joint space at $(\dot{q}_1, \dot{q}_2) = (0.5, 0.5)$. Note that, to achieve safety and feasibility guarantees, the size of the learned unsafe zone is about twice the size of the actual obstacle. Moreover, the learned unsafe zone does not include the whole obstacle region, since the workspace of the robot does not extend to the upper-right part of the obstacle itself, which is therefore not sampled.

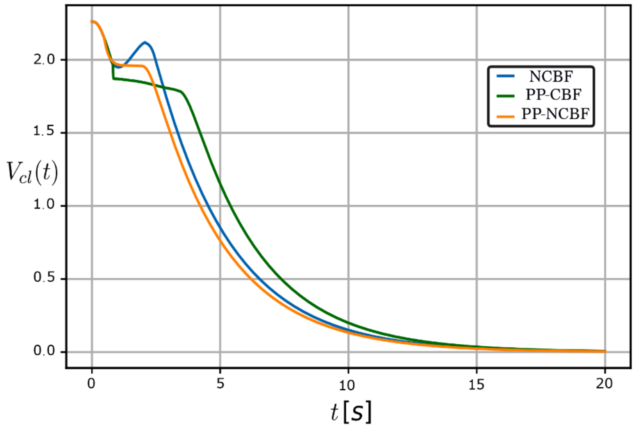

Fig. 4. Simulations on the 2R robot: storage function $V_{cl}(t)$.

Table 1
Convergence statistics for different learning rates.

	1×10^{-3}	1×10^{-4}	5×10^{-5}	1×10^{-5}	5×10^{-6}
% Convergence	0%	70%	85%	100%	100%
Steps to Converge	None	150	200	400	800

5.3. Convergence analysis

While deep learning-based optimization algorithms often lack formal convergence guarantees due to the non-convexity of their loss landscapes, we seek to provide empirical evidence of convergence behavior under various training configurations. In order to address the stability and reliability of our optimization process, we conducted a structured ablation study focusing on two main points: i) the effect of hyperparameter changes on convergence of the learning process; ii) the variability between the learned CBFs between each training run. With respect to point i), we point out that the hyperparameter which in the current scenario has the greatest effect on the learning process is the learning rate. The authors point out that the quantity $|D_\theta|$, which could be interpreted as the batch size, does not have a great impact on the learning process and is kept at 1000 in order to sample a high number of points. The gradient for the second term in the loss (24) is independent on $|D_\theta|$ and depends only on the quantity k in the Top_k operation. The regularization term in (24) depends on the other hand on $|D_\theta|$, but its effect is negligible and hence does not justify a dedicated ablation study.

On the other hand, we evaluated five distinct learning rates cases with 20 independent training runs per configuration. This setup allows us to assess convergence stability, i.e. the probability with which a training run will reach convergence, and number of iterations to convergence. The results, summarized in Table 1, reveal trends in convergence behavior. In particular, it shows how high initial learning rates degrade the convergence speed but, at the same time, when successful, convergence is achieved in a smaller number of steps. This intuitive result can be explained by the fact that initial high values in gradients bring the learned CBF into undesired local minima from which it is not trivial to escape. On the other hand, we found the best learning rate in the interval $[5e^{-5}, 1e^{-5}]$ with an almost 100% convergence rate in a relatively low number of gradient steps.

With respect to point ii) in Fig. 5 we show that the learning problem leads to very similar solutions across all runs. This experiment has been conducted by repeating the training run 10 times starting from randomly independent network initializations and the plots are obtained by fixing the values for $\dot{x}, \dot{\beta}$ and plotting in blue the region of the space (x, β) where the output of the CBF function has values in $[-0.05, 0.05]$, representing an approximation of the boundary region.

These findings suggest that, while theoretical convergence proofs remain elusive, the optimization process exhibits consistent empirical convergence across a range of reasonable hyperparameter settings.

5.4. Safe set comparison with nonlinear MPC

Following the methodology proposed in [8], we perform a comparative analysis between the safe set derived from a learned Control Barrier Function (CBF) and that obtained using Nonlinear Model Predictive Control (NMPC)[30]. The aim of this study is to put in light the size of the safe sets induced by the two classes of algorithms. Unlike the learned CBF, which yields an explicit representation of the safe set boundary, NMPC defines its safe set implicitly. Consequently, the safe region associated with NMPC cannot be described by an analytical function, and characterizing its boundary requires extensive sampling, making the computation of its exact volume or geometry prohibitively expensive.

To approximate the extent of the NMPC safe set, we consider a representative scenario in which the system is initialized with moderate velocities, specifically $(\dot{x}, \dot{\beta}) = (0.5, 0.5)$. This configuration simulates a case where both the cart and the pole are moving

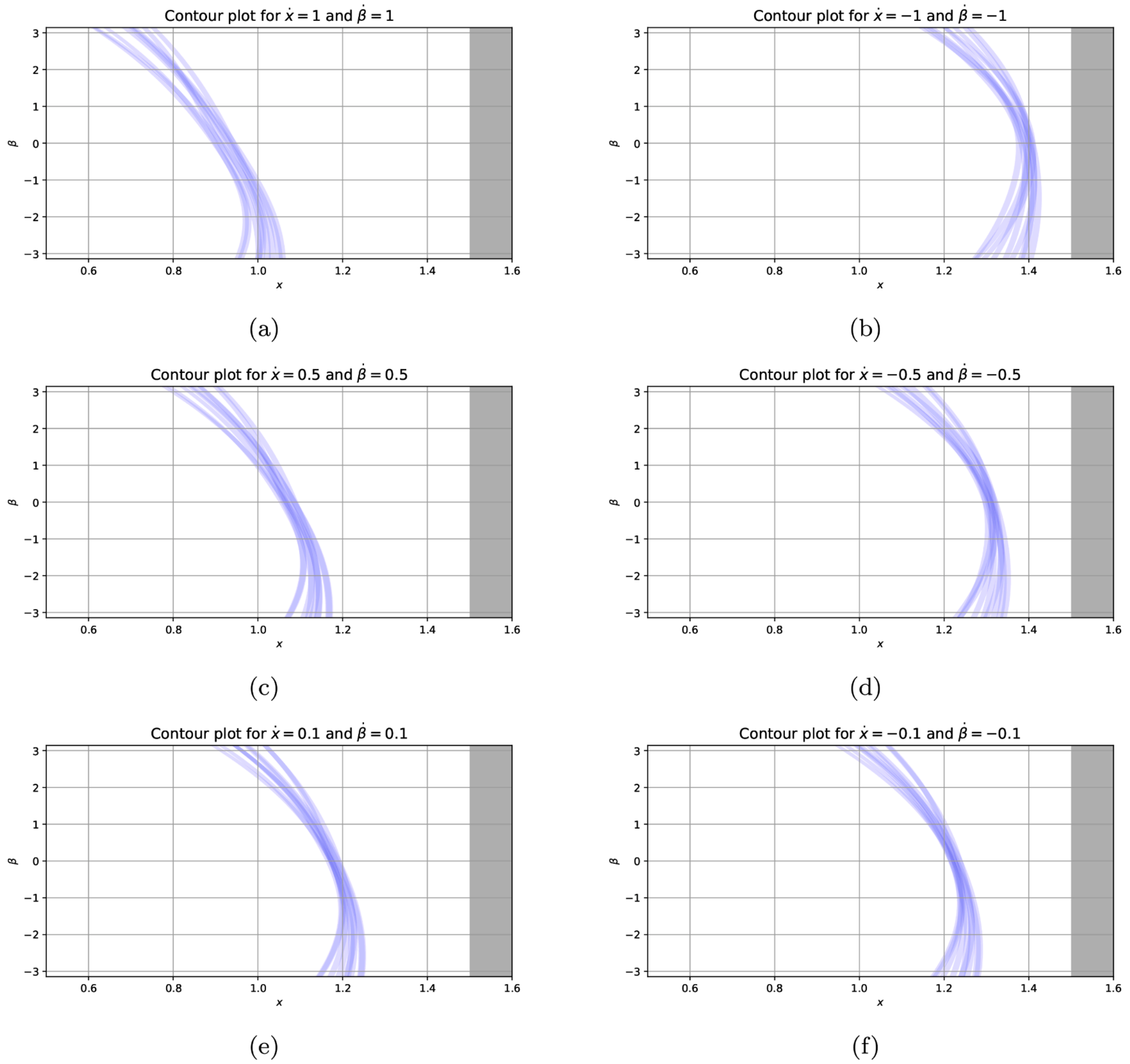

Fig. 5. Contour plots for the CBF obtained after training on the cart-pole system. The plots are obtained fixing the values for $\dot{x}, \dot{\beta}$ as specified in the title of each figure and plotting in blue the region of the space (x, β) where the output of the CBF function has values in $[-0.05, 0.05]$. The region in gray represents the area where the wall is defined, i.e. $x > 1.5$, the area on the right of the contour plot represents the learned unsafe zone, which can be interpreted as a virtual enlargement of the wall. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

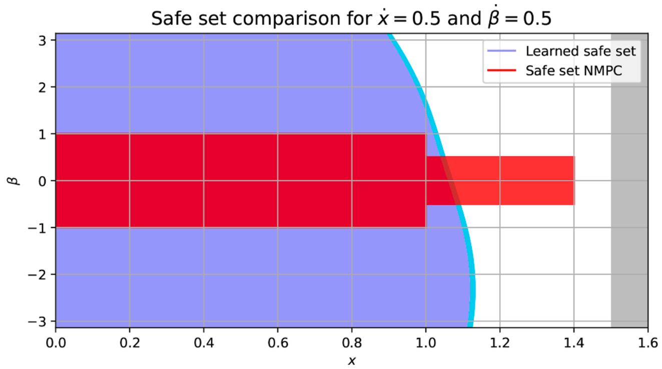

Fig. 6. NMPC safe set obtained at $(\dot{x}, \dot{\beta}) = (0.5, 0.5)$.

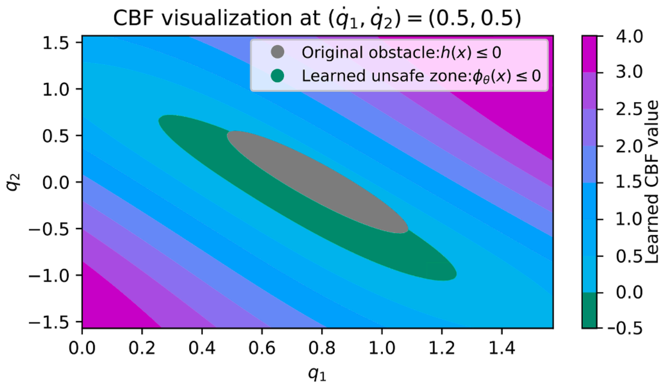

Fig. 7. Comparison of original (grey) and learned (green) unsafe sets. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

toward the wall at a moderate speed. The corresponding safe set for the learned CBF under the same conditions is illustrated in the left column of Fig. 5.

To estimate the safe set under nonlinear model predictive control (NMPC), we evaluate control feasibility over a grid of initial conditions in the (x, β) state space. For each initial state, the NMPC algorithm is applied over a prediction horizon of $N = 80$ steps, with a sampling interval of $dt = 0.025$ s. A total of 50 iterations are conducted per initial condition to assess whether a feasible control trajectory exists that respects all system constraints. This procedure yields an empirical characterization of the safe region, distinguishing initial states that lead to successful control from those that result in constraint violations. As illustrated in Fig. 6, the empirically derived safe set under NMPC represents approximately 60% of the safe set obtained from the CBF algorithm, as shown in Fig. 5c. This discrepancy highlights the limitations imposed by a finite prediction horizon: when the initial state lies beyond a certain threshold, specifically, pendulum deflections exceeding ± 1 radian, the controller is often unable to generate feasible solutions, leading to failure. On the other hand the forward invariance property of CBF algorithms combined with input saturation properties guarantees feasibility.

6. Conclusions and future works

This paper presented a Machine Learning-based approach for safety-critical control of passive systems, that is the first approach (up to the authors' knowledge) aimed at preserving passivity and stability guarantees while avoiding obstacles and satisfying input constraints.

The challenges encountered during this work are summarized in the following to provide insights into the complexities of the study. Regarding sampling efficiency, note that the uniform sampling of $\tilde{S}^{\varepsilon}_{\phi_\theta}$ in Eq. (22) can become computationally expensive. Therefore an hybrid sampling technique was implemented by combining Monte Carlo with targeted sampling near ∂S_{ϕ_θ}, and used a buffer replay for retained past samples meeting the condition $\phi_\theta(x, u_{des}(x)) \leq 0$ to stabilize training. This enabled to learn PP-NCBFs generalized to unseen states, though the safe set was conservative due to limited sampling resolution. Regarding the real-time responsiveness of the proposed method, note that the online evaluation of neural CBFs, namely of $K_\theta(x)$ must be fast enough for real-time control in real-world applications. Therefore a lightweight MLP was trained, namely with 3 layers with 100 units each ensured inference times are $< 1\mathrm{ms}$ on standard hardware. A limitation of this approach is that adaptation to dynamic obstacles (for future works) may require online fine-tuning. Finally, regarding the trade-off between passivity and safety in CBF design, note that ensuring that the learned CBFs simultaneously preserve passivity and enforce safety was non-trivial. The two objectives can conflict, since aggressive safety filters may inject energy into the system violating passivity, while Conservative CBFs may fail to guarantee safety under input constraints. Therefore the passivity-preserving inequality in Eq. (9) was used as an hard constraint in the ML-based optimization problem, and the Lagrangian relaxation in (19) to iteratively tune the trade-off between safety and passivity via λ^p.

Future works shall involve the generalization of the proposed method to the case of obstacles of varying geometry and position, following the direction of [31]. In addition, application of robustness in the sense of [20] will be investigated in order to preserve passivity under model uncertainty.

CRediT authorship contribution statement

Arturo Maiani: Writing – original draft, Software, Methodology; **Federico Baldisseri:** Writing – review & editing, Visualization, Validation, Data curation; **Antonio Pietrabissa:** Supervision, Investigation, Conceptualization.

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.

References

[1] R. Ortega, A.J. Van Der Schaft, I. Mareels, B. Maschke, Putting energy back in control, IEEE Control Syst. Mag. 21 (2) (2001) 18–33. https://doi.org/10.1109/37.915398
[2] L.P. Pagani, G.E. Apostolakis, P. Hejzlar, The impact of uncertainties on the performance of passive systems, Nucl. Technol. 149 (2) (2005) 129–140.
[3] M.E. Guerrero, D.A. Mercado, R. Lozano, C.D. García, IDA-PBC Methodology for a quadrotor UAV transporting a cable-suspended payload, in: 2015 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, 2015, pp. 470–476.
[4] C. Huang, L. Long, Safety-critical multi-tasks control for switched systems via several barrier functions, J. Franklin Inst. 361 (9) (2024) 106906. https://doi.org/10.1016/j.jfranklin.2024.106906
[5] X. Zhang, Z.H. Zhu, S. Xie, H. Gao, G. Li, Barrier function-based prescribed-performance adaptive attitude tracking control for spacecraft with uncertainties, J. Franklin Inst. 360 (12) (2023) 8075–8095. https://doi.org/10.1016/j.jfranklin.2023.06.022
[6] F. Califano, Passivity-Preserving Safety-Critical Control using Control Barrier Functions (2023). arXiv:2303.10981
[7] A. Singletary, S. Kolathaya, A.D. Ames, Safety-Critical kinematic control of robotic systems (2021) 14–19. https://doi.org/10.23919/ACC50511.2021.9482954
[8] S. Liu, C. Liu, J. Dolan, Safe Control Under Input Limits with Neural Control Barrier Functions, 2022. arXiv:2211.11056
[9] C. Fan, K.-F. Chu, X. Wang, K.-W. Kwok, F. Iida, State transition learning with limited data for safe control of switched nonlinear systems, Neural Netw. 180 (2024) 106695.
[10] C. Hu, Z. Wu, Model predictive control of switched nonlinear systems using online machine learning, Chem. Eng. Res. Des. 209 (2024) 221–236.
[11] X. Li, L. Dong, L. Xue, C. Sun, Hybrid reinforcement learning for optimal control of non-linear switching system, IEEE Trans. Neural Netw. Learn. Syst. 34 (11) (2022) 9161–9170.
[12] H. Zhang, C. Zhao, J. Ding, Robust safe reinforcement learning control of unknown continuous-time nonlinear systems with state constraints and disturbances, J. Process Control. 128 (2023) 103028.
[13] M. Green, D.J.N. Limebeer, Linear robust control, Courier Corporation, 2012.
[14] A.T. Azar, F.E. Serrano, I.A. Hameed, N.A. Kamal, S. Vaidyanathan, Robust H-infinity decentralized control for industrial cooperative robots, in: International Conference on Advanced Intelligent Systems and Informatics, Springer, 2019, pp. 254–265.
[15] D. Larby, F. Forni, A passivity preserving h-infinity synthesis technique for robot control, in: 2022 IEEE 61st Conference on Decision and Control (CDC), IEEE, 2022, pp. 1416–1422.
[16] J. Serrano-Delgado, S. Cobreces, E.J. Bueno, M. Rizo, Passivity-based fixed-order H-infinity controller design for grid-forming VSCs, in: 2021 IEEE Applied Power Electronics Conference and Exposition (APEC), IEEE, 2021, pp. 2459–2466.
[17] P. Schwerdtner, M. Voigt, Fixed-order H-infinity controller design for port-Hamiltonian systems, Automatica 152 (2023) 110918.
[18] B. Capelli, C. Secchi, L. Sabattini, Passivity and control barrier functions: optimizing the use of energy, IEEE Rob. Autom. Lett. 7 (2) (2022) 1356–1363. https://doi.org/10.1109/LRA.2021.3139951
[19] G. Notomista, X. Cai, M. Egerstedt, Passivity-Based Decentralized Control of Multi-Robot Systems With Delays Using Control Barrier Functions, CoRR abs/1904.04801 (2019). arXiv:1904.04801
[20] C. Dawson, Z. Qin, S. Gao, C. Fan, Safe Nonlinear Control Using Robust Neural Lyapunov-Barrier Functions, 2021. arXiv:2109.06697
[21] Y. Chang, N. Roohi, S. Gao, Neural Lyapunov Control, CoRR abs/2005.00611 (2020). arXiv:2005.00611
[22] W. Xiao, T.-H. Wang, R. Hasani, M. Chahine, A. Amini, X. Li, D. Rus, Barriernet: differentiable control barrier functions for learning of safe robot control, IEEE Trans. Rob. 39 (3) (2023) 2289–2307. https://doi.org/10.1109/TRO.2023.3249564
[23] A.D. Ames, X. Xu, J.W. Grizzle, P. Tabuada, Control barrier function based quadratic programs for safety critical systems, IEEE Trans. Automat. Contr. 62 (8) (2017) 3861–3876. https://doi.org/10.1109/TAC.2016.2638961
[24] H.K. Khalil, Nonlinear systems; 3rd ed, Prentice-Hall, Upper Saddle River, NJ, Upper Saddle River, NJ, 2002. The book can be consulted by contacting: pH-AID: Wallet, Lionel.
[25] C. Liu, M. Tomizuka, Control in a safe set: addressing safety in human-Robot interactions, 3, 2014. https://doi.org/10.1115/DSCC2014-6048
[26] K. Hornik, Approximation capabilities of multilayer feedforward networks, Neural Netw. 4 (1991) 251–257.
[27] J. Schulman, S. Levine, P. Moritz, M.I. Jordan, P. Abbeel, Trust Region Policy Optimization, CoRR abs/1502.05477 (2015). arXiv:1502.05477
[28] T.G. Molnar, A.D. Ames, Composing control barrier functions for complex safety specifications, IEEE Control Syst. Lett. (2023).
[29] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, 2017. arXiv:1412.6980
[30] D. Menegatti, A. Giuseppi, A. Pietrabissa, Tractable data-Driven model predictive control using one-Step neural networks predictors, IEEE Trans. Autom. Sci. Eng. 22 (2025) 6740–6751. https://doi.org/10.1109/TASE.2024.3453668
[31] Y. Yang, Y. Zhang, W. Zou, J. Chen, Y. Yin, S. Eben Li, Synthesizing control barrier functions with feasible region iteration for safe reinforcement learning, IEEE Trans. Automat. Contr. 69 (4) (2024) 2713–2720. https://doi.org/10.1109/TAC.2023.3336263