



SAPIENZA
UNIVERSITÀ DI ROMA

Department of Computer, Control and Management Engineering (DIAG)

“Antonio Ruberti”

Doctorate of Philosophy in Automatic Control, Bioengineering and
Operations Research

**Energy-Based and Machine Learning
Approaches to Passivity-Preserving and
Safety-Critical Control**

Thesis Advisor

Prof. Antonio Pietrabissa

Candidate

Arturo Maiani

ID 1738271

Academic Year MMXXIV–MMXXV (XXXVIII cycle)

*"Eventually all things fall into place. Until then, laugh at the confusion,
live for the moments, and know everything happens for a reason."*

Albert Schweitzer

Abstract

This thesis advances the synthesis and analysis of nonlinear, safety-aware controllers by leveraging an energy/passivity viewpoint and modern computational tools. Across three studies, we (i) compare how distinct Control Barrier Function (CBF) designs reshape energy flows and closed-loop performance in safety-critical control of robotic systems, (ii) develop a Physics-Informed Neural Networks (PINNs) framework to automate the core partial differential equation (PDE) step of Interconnection and Damping Assignment Passivity-Based Control (IDA-PBC), and (iii) introduce a model-based machine-learning method to synthesize CBFs that preserve passivity of the closed loop under input saturation. Collectively, the results provide a structured path to controllers that guarantee invariance or stabilization while making explicit the energetic mechanisms, shaping, dissipation, and transfer, through which those guarantees are realized.

Two research streams run through all three articles. An energy/passivity-centric perspective as a unifying language for design and analysis. The first study shows how energy-based and exponential CBFs alter energy transfers and impact performance in safety-critical tasks. The second exploits the port-Hamiltonian structure of IDA-PBC to assign a desired energetic architecture and achieve asymptotic stability through damping, with PINNs solving the kinetic-energy PDE up to small residuals. The third enforces safety with CBFs while ensuring that passivity, a property fundamentally tied to energy balance, is retained even with actuator limits. Constructive, computation-ready synthesis validated on realistic systems. We move beyond existence results by providing methods and evidence that scale to nonlinear and underactuated systems: a comparative CBF study on obstacle avoidance with simulations on a 3R planar robot and a 7-DoF manipulator plus hardware tests; a PINN-based IDA-PBC pipeline that reduces reliance on restrictive solvability assumptions and attains stabilization via appropriate damping when the kinetic-energy PDE is solved to a small error; and a passivity-preserving, learning-assisted CBF design evaluated on cart-pole and 2R manipulators under input saturation.

Together these contributions clarify the trade-offs among safety, performance, and energetic behavior; expand the toolbox for energy-shaping and safety-critical design with machine-learning.

Keywords: control barrier functions, passivity-based control, nonlinear control, safety-critical control, machine learning, deep learning.

Contents

1	General Introduction	2
2	Control Barrier Functions	5
2.1	Control-Affine Systems and Basic Notions	5
2.2	Control Barrier Functions (CBFs)	6
2.3	Control-Affine Examples	8
3	Passivity Based Control	10
3.1	Passivity-Based Control and Port–Hamiltonian Systems	10
3.2	Underactuated PH Systems and Why PDEs Arise in IDA–PBC	13
3.3	Applications to Robotic Manipulators	14
4	Machine Learning	15
4.1	Machine Learning Setup: Hypotheses, Risks, and Empirical Risk Minimization . . .	15
4.2	Neural Networks: Models and Architectures	16
4.3	Expressivity: Universal Approximation Theorems	16
4.4	Backpropagation and Reverse-Mode Automatic Differentiation	17
4.5	Optimization: Practical First-Order Methods	18
5	Comparative Analysis of Energy Transfers and Performance in Safety-Critical Control using Control Barrier Functions	20
5.1	Introduction	20
5.2	Background	21
5.2.1	Control Barrier Functions	21
5.2.2	Model of a Fully Actuated Robot Manipulator	24
5.2.3	Energy Based Analysis of CBFs on Mechanical Systems	24
5.3	CBFs for safety-critical control of Robotic Manipulators	26
5.3.1	Energy Based Z-CBFs	26
5.3.2	Energy Based R-CBF	27
5.3.3	Exponential CBFs	28
5.4	Simulations: 3R planar robot	29
5.5	Simulations and Experiments: KUKA LWR IV+	31
5.5.1	Simulation: single obstacle scenario	32
5.5.2	Simulation: multiple obstacles scenario	33
5.5.3	Experimental setup	34
5.6	Conclusions	36
6	A Physics-Informed Neural Networks based method for Interconnection and Damping Assignment Passivity-Based Control	37
6.1	Introduction	37
6.2	Background Theory	39
6.2.1	IDA-PBC	39

6.2.2	PINNs	41
6.3	Main Result	41
6.4	Simulations	46
6.5	Conclusions	48
7	A Machine Learning Approach to Passivity-Preserving Safety-Critical Control	51
7.1	Introduction	51
7.2	Related works and proposed innovations	52
7.3	Preliminaries	53
7.3.1	Passivity and Passivity-Based Control	53
7.3.2	Passivity Preserving Control Barrier Functions	54
7.4	ML-based Approach for Passivity-Preserving Safety-critical Control	55
7.4.1	Neural SC-PBC	55
7.4.2	Deep Learning Optimization	57
7.4.3	Data Sampling, Critic Function and Loss Function	58
7.4.4	Training procedure	59
7.5	Simulations	60
7.5.1	Cart-pole system	61
7.5.2	2R vertical planar robot	63
7.5.3	Convergence Analysis	64
7.5.4	Safe set comparison with Nonlinear MPC	65
7.6	Conclusions and future works	67
8	Other works: Neural Model Predictive Control of Safety-Critical Biological Systems: an Application to Tumor-Immune Cancer Dynamics	69
8.1	Introduction and Related Works	69
8.1.1	Problem Statement	69
8.1.2	Related Works and Proposed Innovations	70
8.2	Background on Neural MPC	71
8.3	Neural MPC Control of Cancer Dynamics	72
8.3.1	Tumor-Immune Cancer Dynamics	72
8.3.2	Background on System Controllability Analysis	75
8.3.3	Neural MPC Control Methodology	76
8.4	Simulations and Results	76
8.5	Conclusions and Future Works	83
9	Other works: Behavioural Cloning for Serious Games in Support of Pediatric Neurorehabilitation	85
9.1	Introduction	85
9.2	Related works	86
9.3	Adaptive Serious Game Design based on Neural Networks and Behavioural Cloning	87
9.3.1	Adaptive Serious Game via Neural Networks	88
9.3.2	Behavioural Cloning Implementation	90
9.4	Experimental Validation	91
9.5	Conclusions	93
	Summary and Outlook	96
A	Appendix	97
A.1	Asymptotic Stability and Small Perturbations	97
A.2	Proofs of injected power equation for each analyzed CBF	98
A.2.1	Power injection: Proof of (5.27)	98

A.2.2 Proof of Eq. (5.30)	98
Bibliography	99

List of Figures

5.1	3R Simulation results for case A	29
5.2	3R Simulation results for case B	30
5.3	KUKA LWR IV+ Robot.	31
5.4	KUKA 7-DoF simulation results	33
5.5	KUKA experimental results	34
5.6	KUKA simulation plots: multiple obstacle scenario.	35
6.1	Loss evolution over 100 epochs.	48
6.2	Simulation results of the learned IDA-PBC framework	50
7.1	Graphical depiction of problem (7.14)	57
7.2	Cart-pole simulations	62
7.3	Simulations on the 2R robot	64
7.4	Simulations on the 2R robot: storage function $V_{cl}(t)$	64
7.5	Comparison of learned safe sets	65
7.6	Contour plots for the learned CBF	66
7.7	NMPC safe set	67
8.1	Phase portraits	79
8.2	Performance comparison: NMPC and Optimal Control	80
8.3	Performance comparison: NMPC and Optimal Control - Safety Violations	82
8.4	NMPC Robustness Ablation	84
9.1	Gameplay screenshot of the game RhythmRun	88
9.2	Neural Network architecture	88
9.3	Behavioural Cloning framework.	90
9.4	Comparison of average error percentages	91
9.5	Data augmentation effects on validation accuracy	92
9.6	Classification performances under classical approaches	92
9.7	Classification performances under the adaptive approach	93
9.8	Average error comparison with threshold-based rules	93

List of Tables

5.1	Comparison of CBF formulations and stability-preserving properties	27
5.2	Nominal controller parameters for the 3R robot	30
5.3	Nominal controller parameters for the KUKA LWR robot	32
5.4	Parameters for the nominal controller and the CBFs in the experiment.	34
6.1	NNs hyperparameters used in the training process.	48
7.1	Convergence statistics for different learning rates	65
8.1	Parameters of the Modified Stepanova Model	74
9.1	Inputs to the NN Classifier	90
9.2	Comparison of dataset statistics	95
9.3	Comparison of classification performances	95

Notation

Spaces

$\mathbb{R}, \mathbb{R}_{\geq 0}, \mathbb{R}_{> 0}$	Reals, nonnegative, and positive reals
$\mathbb{Z}, \mathbb{Z}_{\geq 0}$	Integers, nonnegative integers
$\mathbb{R}^n, \mathbb{R}^{n \times m}$	n -dimensional Euclidean space; $n \times m$ matrices
$\mathcal{D} \subseteq \mathbb{R}^n$	State domain
$\mathcal{U} \subseteq \mathbb{R}^m$	Admissible input set
$\text{Int}(\mathcal{C}), \partial\mathcal{C}, \bar{\mathcal{C}}$	Interior, boundary, and closure of $\mathcal{C}$
$\mathbf{1}_{\mathcal{C}}(x)$	Indicator of set $\mathcal{C}$

Vectors and Matrices

$\mathbf{0}_n, \mathbf{0}_{n \times m}$	Zero vector/matrix
I_n	Identity matrix
x_i	i -th component of x
$\text{diag}(d_1, \dots, d_k)$	Diagonal matrix
$A^\top$	Transpose of A
$\text{rank}(A)$	Rank of A
$P \succeq 0$ ($\succ 0$)	Symmetric positive (semi)definite

Calculus and Systems

$\nabla h(x)$	Gradient of $h : \mathbb{R}^n \rightarrow \mathbb{R}$
$J_f(x) = \frac{\partial f}{\partial x}$	Jacobian of $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$
$\nabla^2 h(x)$	Hessian of h
$\text{div } f(x)$	Divergence of f
$\dot{z}, z^{(i)}$	First and i -th time derivative
$L_f h$	Lie derivatives $\nabla h^\top f$
$L_f^k h$	Higher-order Lie derivatives
$\mathcal{C} = \{x : h(x) \geq 0\}$	Safe set; $\text{Int}(\mathcal{C}), \partial\mathcal{C}$ as usual

Acronyms

CBF	Control Barrier Function
Z-CBF, R-CBF	Zeroing / Reciprocal Control Barrier Function
HOCBF	High-Order Control Barrier Function
Exp-CBF	Exponential Control Barrier Function
CLF	Control Lyapunov Function
QP	Quadratic Program
PH	Port–Hamiltonian
PBC	Passivity-Based Control
IDA–PBC	Interconnection and Damping Assignment PBC
EL	Euler–Lagrange (mechanical) system
DOF	Degree(s) of Freedom
ML	Machine Learning
MLP	Multilayer Perceptron
PINN	Physics-Informed Neural Network
ReLU	Rectified Linear Unit

Chapter 1

General Introduction

Safety-critical control of nonlinear systems demands controllers that guarantee both stability and the invariance of prescribed safe sets. In robotics and cyber-physical systems, these guarantees must coexist with modeling uncertainty, actuator limits, and performance requirements. This thesis advances a unifying, energy-centric route to safety-aware control by combining three pillars: (i) **Control Barrier Functions** (CBFs) for forward invariance, (ii) **Passivity-/Energy-based Control** (PBC) for structure and stability via energy shaping and damping, and (iii) **Machine Learning** (ML) tools, most notably Physics-Informed Neural Networks (PINNs) and model-based learning, to automate hard analytical steps while preserving interpretability and certificates.

The thesis is organized in two layers. First, three *introductory chapters* develop the fundamentals needed to read the main research contributions. Second, three *core research chapters* apply and extend these tools on realistic nonlinear and robotic systems. Two additional chapters collect complementary works that broaden the methodological and application scope.

Chapter 2 (*Control Barrier Functions*) introduces the theory of set invariance for continuous-time systems, classical and exponential CBFs, and their implementation as safety filters. Emphasis is placed on how barrier design reshapes closed-loop energy flows and on practical aspects (relative degree, feasibility, tuning, and performance trade-offs).

Chapter 3 (*Passivity Based Control*) reviews port-Hamiltonian modeling, Interconnection and Damping Assignment Passivity-Based Control (IDA-PBC), and the role of storage, dissipation, and interconnection in shaping closed-loop behavior. The chapter highlights how energetic structure offers both interpretability and modularity and how it interfaces naturally with safety constraints.

Chapter 4 (*Machine Learning for Control*) surveys learning tools that integrate with control certificates: PINNs for physics-constrained approximation, model-based ML for structure-preserving synthesis, and best practices for keeping safety and passivity in the loop (loss design, residual control, and validation).

Chapter 5 [1] **A. Maiani, F. Califano, L. Govoni, and A. Pietrabissa, "Comparative Analysis of Energy Transfers and Performance in Safety-Critical Control using Control Barrier Functions", 2026 IEEE International Conference on Robotics and Automation (ICRA), under review.** This chapter develops a systematic comparison between *energy-based* and *exponential* CBFs.

Beyond feasibility, it makes explicit how each construction shapes energy transfers (storage, dissipation, interconnection) and impacts tracking, transients, and control effort. The study blends theory with simulations on a 3R planar arm and a 7-DoF manipulator, plus hardware experiments, yielding practical guidelines for selecting and tuning CBFs when safety and performance must be co-engineered.

Chapter 6 [2] *A. Di Paola, A. Maiani, and D. Menegatti, "A Physics-Informed Neural Networks Based Method for Interconnection and Damping Assignment Passivity-Based Control", 2025 IEEE Conference on Decision and Control (CDC), accepted for publication.* Addressing the classical bottleneck of IDA-PBC, this chapter proposes a PINN framework that solves the kinetic-energy PDE within a physics-aware loss, leveraging port-Hamiltonian structure. A key result shows that, under mild assumptions on the potential energy, *small PDE residuals plus suitable damping* suffice for asymptotic stabilization, thereby reducing restrictive solvability assumptions while keeping energetic interpretability. Numerical studies demonstrate scalability to cases that are analytically intractable.

Chapter 7[3] *A. Maiani, F. Baldisseri, and A. Pietrabissa, "A Machine Learning Approach to Passivity-Preserving Safety-Critical Control", 2025 Journal of the Franklin Institute, under review.* This chapter unifies CBF-based safety with passivity by learning barrier functions that preserve closed-loop passivity even under actuator saturation. The resulting synthesis ensures forward invariance *and* an energy balance aligned with passivity-based stability. Validation on cart-pole and 2R manipulators shows that enforcing safety need not erode passivity margins, clarifying how safety filters, energy shaping, and damping interplay in practice.

Chapter 8 [4] *F. Baldisseri, D. Menegatti, A. Maiani, A. Giuseppi, and A. Pietrabissa, "Neural MPC for Safety-Critical Biological Systems: an Application to Tumor-Immune Cancer Dynamics", 2025 International Journal of Control, under review.*

The controller learns one-step predictors from data to enable constraint-aware control without an analytical model. In-silico studies on a tumor-immune therapy model show performance comparable to classical MPC with full-model knowledge and to Pontryagin-optimal control, with robustness to parametric uncertainty relevant to clinical constraints.

Chapter 9 [5] *F. Baldisseri, E. Montecchiani, A. Maiani, D. Menegatti, A. Giuseppi, A. Pietrabissa, V. Fogliati, and F. Delli Priscoli, "Behavioural Cloning for Serious Games in Support of Pediatric Neurorehabilitation", 2023 31st IEEE Mediterranean Conference on Control and Automation (MED), 2023.*

An ANN classifier with data augmentation via Behavioural Cloning improves performance and enables autonomous, patient-specific difficulty adjustment, building on the guidelines defined in [6]. Experimental analyses on motor-coordination training illustrate the approach and its potential translational impact.

Chapter 9.5 (*General Conclusions*) synthesizes the main findings, limitations, and future directions. The overarching message is that *energetic structure is a design lever*: by making storage, dissipation, and transfer explicit, we obtain controllers that are **safe**, **stable**, and **interpretable**.

Future work points to robust/stochastic extensions, certificate-aware learning, and scalable implementations for high-DoF systems and interconnected networks.

Through the integration of CBF-based safety, passivity-oriented energy shaping, and learning-assisted computation, the thesis contributes a coherent pathway to safety-critical nonlinear control that marries formal guarantees with practical, computation-ready synthesis.

Chapter 2

Control Barrier Functions

2.1 Control-Affine Systems and Basic Notions

We consider input-affine nonlinear systems of the form

$$\dot{x} = f(x) + g(x)u, \quad (2.1)$$

with state $x \in \mathcal{D} \subseteq \mathbb{R}^n$, input $u \in \mathcal{U} \subseteq \mathbb{R}^m$, and locally Lipschitz $f : \mathcal{D} \rightarrow \mathbb{R}^n$, $g : \mathcal{D} \rightarrow \mathbb{R}^{n \times m}$. Under measurable and locally bounded $u(\cdot)$, (2.1) admits unique Carathéodory solutions on maximal intervals [7].

Lie derivatives. For a continuously differentiable scalar field $h : \mathcal{D} \rightarrow \mathbb{R}$, the Lie derivatives along f and g are

$$L_f h(x) := \nabla h(x)^\top f(x), \quad L_g h(x) := \nabla h(x)^\top g(x) \in \mathbb{R}^{1 \times m}.$$

Higher-order Lie derivatives are defined recursively, e.g., $L_f^2 h := L_f(L_f h)$, and $L_g L_f^k h := \nabla(L_f^k h)^\top g$ [8, Ch. 3].

Definition 1 (Class- $\mathcal{K}$ and extended class- $\mathcal{K}$, [7]). *A function $\alpha : [0, a) \rightarrow [0, \infty)$ is class- $\mathcal{K}$ if it is continuous, strictly increasing, and $\alpha(0) = 0$. A function $\alpha : (-b, a) \rightarrow \mathbb{R}$ is extended class- $\mathcal{K}$ if it is continuous, strictly increasing and $\alpha(0) = 0$.*

Definition 2 (Lyapunov function, [7]). *A continuous differentiable $V : \mathcal{D} \rightarrow \mathbb{R}_{\geq 0}$ is a Lyapunov function for an equilibrium x^* if $V(x^*) = 0$, $V(x) > 0$ for $x \neq x^*$, and $\dot{V}(x) = L_f V(x) + L_g V(x)u(x) \leq 0$ in a neighborhood of x^* .*

Definition 3 (Relative degree, [8]). *Let $b : \mathcal{D} \rightarrow \mathbb{R}$ be differentiable. The relative degree $r \in \mathbb{N}$ of b with respect to (2.1) at x is the smallest integer such that $L_g L_f^{r-1} b(x) \neq 0$ while $L_g L_f^k b(x) = 0$ for $k = 0, \dots, r-2$.*

Safe sets and forward invariance. Given $h : \mathcal{D} \rightarrow \mathbb{R}$, define the *safe set*

$$\mathcal{C} := \{x \in \mathcal{D} : h(x) \geq 0\}, \quad \partial\mathcal{C} := \{x : h(x) = 0\}, \quad \text{Int}(\mathcal{C}) = \{x : h(x) > 0\}. \quad (2.2)$$

A set $\mathcal{C}$ is forward invariant for (2.1) if $x(0) \in \mathcal{C} \Rightarrow x(t) \in \mathcal{C}$ for all $t \geq 0$.

2.2 Control Barrier Functions (CBFs)

CBFs are conditions on h (and the feedback) that guarantee forward invariance of $\mathcal{C}$ for (2.1). We summarize the two classical notions and then extend to higher relative degree.

Zeroing CBFs (Z-CBFs)

Assume $h \in C^1$ and $\nabla h(x) \neq 0$ on $\partial\mathcal{C}$. The function h is a *control barrier function* on $\mathcal{D}$ if there exists an extended class- $\mathcal{K}$ function α such that

$$\sup_{u \in \mathcal{U}} [L_f h(x) + L_g h(x)u] \geq -\alpha(h(x)), \quad \forall x \in \mathcal{D}. \quad (2.3)$$

Under (2.3), any locally Lipschitz feedback $u(x)$ satisfying $L_f h(x) + L_g h(x)u(x) \geq -\alpha(h(x))$ renders $\mathcal{C}$ forward invariant and, moreover, $\mathcal{C}$ is asymptotically stable relative to $\mathcal{D}$ [9, 10]. In practice, Z-CBFs are enforced via a *safety filter* that minimally modifies a nominal control $u_{\text{nom}}(x)$ by solving the *convex* QP

$$\begin{aligned} u^*(x) &= \arg \min_{u \in \mathcal{U}} \|u - u_{\text{nom}}(x)\|^2 \\ \text{s.t. } & L_f h(x) + L_g h(x)u \geq -\alpha(h(x)). \end{aligned} \quad (2.4)$$

When $\mathcal{U} = \mathbb{R}^m$ and $L_g h(x) \neq 0$, the Karush-Kuhn-Tucker system yields a closed-form correction on the active set; see Definition 5 in [9].

Reciprocal CBFs (R-CBFs)

An alternative is to define a *barrier* $B : \text{Int}(\mathcal{C}) \rightarrow \mathbb{R}_{>0}$ that blows up at $\partial\mathcal{C}$ and satisfies (for some class- $\mathcal{K}$ functions $\alpha_1, \alpha_2, \alpha_3$)

$$\frac{1}{\alpha_1(h(x))} \leq B(x) \leq \frac{1}{\alpha_2(h(x))}, \quad (2.5)$$

$$\inf_{u \in \mathcal{U}} [L_f B(x) + L_g B(x)u] < \alpha_3(h(x)), \quad (2.6)$$

$$L_g B(x) = 0 \Rightarrow L_f B(x) < \alpha_3(h(x)), \quad (2.7)$$

for all $x \in \text{Int}(\mathcal{C})$. Then $\mathcal{C}$ is forward invariant [11]. The corresponding safety filter solves

$$\begin{aligned} u^*(x) &= \arg \min_{u \in \mathcal{U}} \|u - u_{\text{nom}}(x)\|^2 \\ \text{s.t. } & L_f B(x) + L_g B(x)u \leq \alpha_3(h(x)). \end{aligned} \quad (2.8)$$

Relative Degree, High-Order CBFs (HOCBFs) and Exponential CBFs

The constraints $h(x) \geq 0$ may have relative degree $r > 1$ w.r.t. (2.1). For instance, when h depends on configuration q only in mechanical systems, $L_g h \equiv 0$ and one must differentiate multiple times before u appears. Two popular treatments are HOCBFs and Exponential CBFs.

High-Order CBFs. Let $h \in C^r$ have relative degree r . Define recursively

$$\psi_0(x) = h(x), \quad \psi_i(x) = \dot{\psi}_{i-1}(x) + \alpha_i(\psi_{i-1}(x)), \quad i = 1, \dots, r-1, \quad (2.9)$$

where each α_i is class- $\mathcal{K}$. A function h is a *high-order control barrier function* if there exists a class- $\mathcal{K}$ function α_r such that the *affine* inequality

$$L_f^r h(x) + L_g L_f^{r-1} h(x) u + S(x) \geq -\alpha_r(\psi_{r-1}(x)) \quad (2.10)$$

holds for all x in the domain, where $S(x)$ collects Lie-derivative terms depending only on f and $\alpha_1, \dots, \alpha_{r-1}$ [12, 13]. Enforcing (2.10) via a QP ensures forward invariance of the set $\{x : \psi_i(x) \geq 0, i = 0, \dots, r-1\}$ [13].

Exponential CBFs (Exp-CBFs). For relative degree r , an Exp-CBF imposes a linear differential inequality of the form

$$h^{(r)}(x) + \sum_{i=0}^{r-1} k_{r-i} h^{(i)}(x) \geq 0, \quad (2.11)$$

with coefficients $k_i > 0$ chosen so that the characteristic polynomial is Hurwitz; this yields *exponential* convergence of $h^{(i)}$ to nonnegativity and ensures safety [14]. Equation (2.11) is affine in u because u enters $h^{(r)}$ via $L_g L_f^{r-1} h(x) u$.

CLF–CBF Quadratic Programs

To *simultaneously* achieve stabilization and safety, a control Lyapunov function (CLF) V can be combined with (H)CBFs. A standard CLF condition is

$$\inf_{u \in \mathcal{U}} [L_f V(x) + L_g V(x) u] \leq -\gamma(V(x)), \quad (2.12)$$

with class- $\mathcal{K}$ γ [7, 9]. A common convex CLF–CBF QP is

$$\begin{aligned} u^*(x) &= \arg \min_{u, \delta} \|u - u_{\text{nom}}(x)\|^2 + p \delta^2 \\ \text{s.t.} \quad & L_f V(x) + L_g V(x) u \leq -\gamma(V(x)) + \delta, \\ & (\text{Z/H/Exp-}) \text{ CBF constraint as in (2.3), (2.10), or (2.11),} \\ & u \in \mathcal{U}, \end{aligned} \quad (2.13)$$

with slack $\delta \geq 0$ penalized by $p > 0$. The slack preserves *feasibility* when CLF and CBF constraints momentarily conflict, while safety constraints are usually kept *hard* [9, 15, 16].

Input Constraints, Feasibility, and the “Safety Index” View

Hard input bounds $u \in \mathcal{U} = \{u : u_{\min} \leq u \leq u_{\max}\}$ (componentwise) are naturally handled as convex constraints in (2.4)–(2.13). Feasibility can be lost if the admissible set $\mathcal{U}$ is too restrictive relative to the CBF condition. Standard remedies include:

- *Softening* the CLF with a slack (as in (2.13)); if necessary, one can introduce a *relaxed* CBF with carefully weighted slack, understanding the safety implications [9, 17].
- *Designing h with relative degree one* (e.g., by including velocity terms), or using HOCBFs/Exp-CBFs when $r > 1$ [13, 14].
- *Safety index*: one may encode safety as $\dot{\phi}(x, u) \leq -\kappa\phi(x)$ for an index ϕ with $\mathcal{C} = \{x : \phi(x) \leq 0\}$; this yields an affine inequality in u analogous to a CBF constraint and can be integrated in QPs with input bounds [18].

Regularity and Well-Posedness

Assume (i) f, g are locally Lipschitz; (ii) the QP objective in (2.4)–(2.13) is strictly convex; (iii) all inequality constraints are affine in u and satisfy a *linear independence constraint qualification* at the optimizer; and (iv) the QP is feasible on a neighborhood of interest. Then the optimizer $u^*(x)$ is *locally Lipschitz* in x (piecewise affine when $\mathcal{U} = \mathbb{R}^m$ and a single constraint is active), which guarantees existence and uniqueness of closed-loop solutions for (2.1) with $u = u^*(x)$ [9, 19]. Robustness of CBF-based safety to small modeling errors can be analyzed following [17, 20].

2.3 Control-Affine Examples

We provide a short, didactic example that is widely used as benchmark: safety critical control of a fully actuated robot manipulator, which will be at the center of the analysis developed in 5.

Fully Actuated Robot Manipulators

Consider a serial, fully actuated n -DOF manipulator with configuration $q \in \mathbb{R}^n$, dynamics

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = u, \quad (2.14)$$

and state $x = (q^\top, \dot{q}^\top)^\top$. In the control-affine form (2.1), one has

$$f(x) = \begin{bmatrix} \dot{q} \\ -M^{-1}(q)(C(q, \dot{q})\dot{q} + G(q)) \end{bmatrix}, \quad g(x) = \begin{bmatrix} 0_{n \times n} \\ M^{-1}(q) \end{bmatrix},$$

which is locally Lipschitz on domains where $M(q)$ is smooth and uniformly positive definite. Standard properties hold: (i) $M(q) = M(q)^\top \succ 0$; (ii) $\dot{M}(q) - 2C(q, \dot{q})$ is skew-symmetric (thus $\dot{q}^\top (\dot{M} - 2C)\dot{q} \equiv 0$); $G(q) = \nabla V(q)$ where $V(q)$ is the potential energy of the system; see [21, 22].

Relative degree in manipulators. If one defines a purely positional safety function, e.g., joint-limit avoidance with

$$h(q) := (q_{\max} - q_i)(q_i - q_{\min}) \quad \text{for joint } i,$$

then $L_g h \equiv 0$ (because h depends only on q) and the relative degree is $r = 2$ w.r.t. u . One can either:

- use an *HOCBF/Exp-CBF* with $r = 2$ to obtain an affine inequality in u (via $L_g L_f h$), or
- design $h(q, \dot{q})$ that includes velocity terms, e.g., $h(q, \dot{q}) = \bar{h}(q) - \rho \|\dot{q}\|$ so that $L_g h \neq 0$ and the relative degree is $r = 1$ (care must be taken to preserve convexity and smoothness).

For example, with $r = 2$ an HOCBF constructed from $\psi_0 = h(q)$, $\psi_1 = \dot{h} + \alpha_1(h)$ yields the affine constraint

$$L_f^2 h(x) + L_g L_f h(x) u + \alpha_2(\psi_1(x)) \geq 0,$$

which can be imposed together with input limits $u \in \mathcal{U}$ in a QP; see [12, 13].

Chapter 3

Passivity Based Control

3.1 Passivity-Based Control and Port–Hamiltonian Systems

This section develops the background on passivity theory and its use in control design, with emphasis on robotics. We introduce passivity and storage functions, formalize Port–Hamiltonian (PH) models and their power balance, connect passivity with LaSalle’s invariance principle, and summarize standard passivity-based controllers for robotic manipulators (energy shaping and damping injection). We then explain *why* partial differential equations (PDEs) arise in Interconnection and Damping Assignment PBC (IDA–PBC) for *underactuated* PH/EL systems and present their main canonical forms and properties.

Passivity, storage functions, and interconnections

Consider a control system

$$\dot{x} = f(x, u), \quad y = h(x, u), \quad x \in \mathcal{D} \subset \mathbb{R}^n, \quad u \in \mathbb{R}^m, \quad y \in \mathbb{R}^m. \quad (3.1)$$

The supply rate is $w(u, y) = u^\top y$, representing exchanged power.

Definition 4 (Passivity, [23]). *The system (3.1) is passive if there exists a locally Lipschitz, nonnegative storage function $S : \mathcal{D} \rightarrow \mathbb{R}_{\geq 0}$ such that for all $t \geq t_0$ and all admissible inputs,*

$$S(x(t)) - S(x(t_0)) \leq \int_{t_0}^t u(\tau)^\top y(\tau) d\tau. \quad (3.2)$$

If the right-hand side is replaced by $\int (u^\top y - \rho \|y\|^2) dt$ for some $\rho > 0$, the system is output strictly passive. More generally, passivity indices $(\nu, \rho) \in \mathbb{R}^2$ capture shortage/excess: $\int (u^\top y - \nu \|u\|^2 - \rho \|y\|^2) dt$.

Differentiating (3.2) yields the pointwise estimate $\dot{S}(x) \leq u^\top y$ almost everywhere.

Interconnection theorems. Passivity is preserved under power-preserving interconnections: (i) *Parallel/series* interconnections of passive systems are passive with storage the sum of storages. (ii) *Feedback* of two passive systems is passive when the loop is power-conjugate ($u_1 = -y_2$, $u_2 = y_1$). Output strict passivity of at least one block implies $\mathcal{L}_2$ -stability of the loop; see [23, 24].

Port–Hamiltonian systems

A finite-dimensional port–Hamiltonian (PH) system with m ports is defined as

$$\dot{x} = (J(x) - R(x))\nabla H(x) + G(x)u, \quad y = G(x)^\top \nabla H(x), \quad (3.3)$$

where $x \in \mathcal{D} \subseteq \mathbb{R}^n$ is the state, $u, y \in \mathbb{R}^m$ are the input and output (port variables), and:

- $H : \mathcal{D} \rightarrow \mathbb{R}_{\geq 0}$ is the *Hamiltonian*, representing the stored energy;
- $J(x) = -J(x)^\top$ is the *interconnection matrix*, defining power-preserving internal energy exchange;
- $R(x) = R(x)^\top \succeq 0$ models dissipation; and
- $G(x)$ maps external ports to the state dynamics.

The time derivative of the Hamiltonian along trajectories of (3.3) is

$$\dot{H}(x) = \nabla H(x)^\top \dot{x} = -\nabla H(x)^\top R(x) \nabla H(x) + u^\top y, \quad (3.4)$$

which expresses the *power balance* between stored, dissipated, and supplied energy. Consequently, every PH system is passive with storage function $S(x) = H(x)$ and supply rate $w(u, y) = u^\top y$ [25–27].

Derivation from Euler–Lagrange mechanics. Consider a mechanical system with configuration $q \in \mathbb{R}^n$, generalized momenta $p = M(q)\dot{q}$, and total energy

$$H(q, p) = \frac{1}{2} p^\top M(q)^{-1} p + V(q),$$

where $M(q) = M(q)^\top \succ 0$ is the inertia matrix and $V(q)$ the potential energy. The Euler–Lagrange (EL) equations with viscous damping $D(q, \dot{q}) \succeq 0$ and actuation $u \in \mathbb{R}^m$ read

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + D(q, \dot{q})\dot{q} + \nabla_q V(q) = B(q)u, \quad (3.5)$$

where $C(q, \dot{q})$ is the Coriolis/centrifugal matrix and $B(q)$ the input mapping. In the state $x = (q^\top, p^\top)^\top$, with $\dot{q} = M(q)^{-1}p$, equation (3.5) becomes

$$\begin{aligned} \dot{q} &= \frac{\partial H}{\partial p}, \\ \dot{p} &= -\frac{\partial H}{\partial q} - D(q, \dot{q}) \frac{\partial H}{\partial p} + B(q)u. \end{aligned}$$

This system can be written compactly as

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}}_J \nabla H(x) - \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & D(q, \dot{q}) \end{bmatrix}}_{R \succeq 0} \nabla H(x) + \underbrace{\begin{bmatrix} 0 \\ B(q) \end{bmatrix}}_{G(x)} u, \quad (3.6)$$

which is a particular case of the general PH structure (3.3). The corresponding port output $y = G(x)^\top \nabla H(x) = B(q)^\top \dot{q}$ represents the generalized velocities weighted by the actuation matrix. Substituting (3.6) into (3.4) yields

$$\dot{H} = -\dot{q}^\top D(q, \dot{q}) \dot{q} + u^\top y,$$

confirming the passivity of mechanical PH systems. Classical robot dynamics of the form (3.5) thus constitute a structured subclass of PH systems [21, 22, 26, 27].

LaSalle’s invariance principle in the PH/PBC context

Consider a closed-loop system that is *passive* with respect to a nonnegative storage function $S : \mathcal{D} \rightarrow \mathbb{R}_{\geq 0}$, and satisfies the dissipation inequality

$$\dot{S}(x) \leq -\sigma(x) \leq 0, \quad (3.7)$$

for some continuous *dissipation function* $\sigma : \mathcal{D} \rightarrow \mathbb{R}_{\geq 0}$.

Precompact trajectories. A trajectory $x(t)$ of the system is said to be *precompact* if its image $\{x(t) : t \geq 0\}$ lies within a compact subset of the state space $\mathcal{D}$. Intuitively, this means that the trajectory remains bounded and does not diverge to infinity as time evolves. This condition ensures that every sequence of states along the trajectory contains a convergent subsequence, an essential requirement for applying LaSalle’s invariance principle.

Detectability of the dissipation set. Define the *dissipation-free set*

$$\mathcal{E} := \{x \in \mathcal{D} : \sigma(x) = 0\}.$$

The set $\mathcal{E}$ is said to be *detectable* if every trajectory that remains in $\mathcal{E}$ necessarily converges to the desired equilibrium point. Equivalently, if the system output associated with dissipation (for instance, the generalized velocity or damping port variable in a port–Hamiltonian model) is identically zero along a trajectory, then the corresponding state must asymptotically approach the equilibrium. This property guarantees that no nontrivial motion can persist with zero dissipation.

LaSalle’s invariance principle. Under these conditions, namely, that trajectories are precompact and that $\mathcal{E}$ is detectable, LaSalle’s invariance principle ensures that every trajectory of the passive system converges to the largest invariant subset of $\mathcal{E}$. In the context of port–Hamiltonian and passivity-based control, the function $\sigma(x)$ typically represents the total physical dissipation of the system (through $R(x)$) together with any additional damping introduced by the controller. Hence, the invariance principle establishes a direct and physically interpretable link between passivity, energy dissipation, and the asymptotic stability of the desired equilibrium [25, 27].

Passivity–based control: energy shaping and damping injection

Passivity–Based Control (PBC) exploits the passivity property of physical systems to achieve stabilization by modifying the energy function and injecting suitable damping. A general feedback law

has the form

$$u = \beta(x) - K(x)y, \quad K(x) = K(x)^\top \succeq 0, \quad (3.8)$$

where $\beta(x)$ represents an *energy-shaping* term and $K(x)y$ provides *damping injection*. In the *Interconnection and Damping Assignment PBC* (IDA–PBC) approach, the objective is to assign to the closed loop a desired PH structure

$$\dot{x} = (J_d(x) - R_d(x)) \nabla H_d(x), \quad (3.9)$$

where $J_d(x) = -J_d(x)^\top$ defines the shaped interconnection, $R_d(x) = R_d(x)^\top \succeq 0$ the desired dissipation, and $H_d : \mathcal{D} \rightarrow \mathbb{R}_{\geq 0}$ the shaped Hamiltonian with a strict minimum at the target equilibrium x^* . When such a feedback law exists, H_d serves as a Lyapunov function for the closed loop, and the dissipation term ensures

$$\dot{H}_d(x) = -\nabla H_d(x)^\top R_d(x) \nabla H_d(x) \leq 0,$$

which implies asymptotic convergence to x^* by LaSalle’s invariance principle [25–27].

3.2 Underactuated PH Systems and Why PDEs Arise in IDA–PBC

Let (3.3) be given with input matrix $G(x) \in \mathbb{R}^{n \times m}$ of rank $m < n$ (underactuation). The IDA–PBC objective (3.9) demands that the feedback exactly *assign* the target vector field using the available inputs:

$$\exists u(x) \text{ s.t. } (J_d(x) - R_d(x)) \nabla H_d(x) = (J(x) - R(x)) \nabla H(x) + G(x) u(x). \quad (3.10)$$

Denote by $G_\perp(x) \in \mathbb{R}^{n \times (n-m)}$ any *left annihilator* of $G(x)$, i.e., $G_\perp(x)^\top G(x) = 0$ and $\text{rank } G_\perp = n - m$. Since the input can only generate directions in $\text{Im } G(x)$, (3.10) is implementable *iff* the difference between the target and the open-loop vector fields has no component along the unactuated directions:

$$G_\perp(x)^\top \left[(J_d(x) - R_d(x)) \nabla H_d(x) - (J(x) - R(x)) \nabla H(x) \right] = 0. \quad (3.11)$$

Equation (3.11) is a *first-order PDE system* for the unknowns (H_d, J_d, R_d) , called the *matching equations* [26, 27]. In the fully actuated case ($m = n$), $G_\perp \equiv 0$ and no PDEs arise: any (H_d, J_d, R_d) is assignable. *Underactuation* ($m < n$) forces the target energy landscape and interconnection/damping to satisfy (3.11), which are nontrivial PDE constraints.

From Matching PDEs to Implementable Feedback Given a solution (H_d, J_d, R_d) of (3.11), the IDA–PBC feedback that realizes (3.9) can be written as

$$u(x) = G(x)^\dagger \left([J_d(x) - R_d(x)] \nabla H_d(x) - [J(x) - R(x)] \nabla H(x) \right) + (I - G^\dagger G) \mu(x), \quad (3.12)$$

where $G^\dagger$ is any right pseudoinverse and μ is an arbitrary internal feedback (it does not affect the port variables). Feasibility requires (3.11); the *free* terms (μ , J_2 , and parts of M_d/V_d) are then tuned for performance. The resulting closed loop is PH with storage H_d and port (u, y) , and damping

assignment ensures $\dot{H}_d \leq 0$ [26, 27].

Mechanical (EL) systems and “simple” targets. For EL systems with $x = (q, p)$ it is standard to look for *simple mechanical* targets

$$H_d(q, p) = \frac{1}{2} p^\top M_d(q)^{-1} p + V_d(q), \quad J_d(x) = \begin{bmatrix} 0 & I \\ -I & J_2(q, p) \end{bmatrix}, \quad R_d(x) = \begin{bmatrix} 0 & 0 \\ 0 & D_d(q) \end{bmatrix}, \quad (3.13)$$

where $M_d(q) \succ 0$ is the *desired inertia*, V_d the *shaped potential*, $D_d \succeq 0$ the *injected damping*, and $J_2 = -J_2^\top$ generates gyroscopic (energy-preserving) forces. Substituting (3.13) into (3.11) splits the matching conditions into two families of PDEs: *potential-energy matching* and *kinetic-energy matching*. These are the PDEs routinely referred to in the IDA–PBC literature for *underactuated* mechanical systems [27–29].

3.3 Applications to Robotic Manipulators

Consider an n –DOF mechanical system in EL form

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + \nabla V(q) = B(q) u, \quad (3.14)$$

where $M(q) = M(q)^\top \succ 0$, $C(q, \dot{q})$ is such that $\dot{M}(q) - 2C(q, \dot{q})$ is skew-symmetric, and $B(q) \in \mathbb{R}^{n \times m}$ has rank $m < n$ (underactuation). Let $B_\perp(q) \in \mathbb{R}^{n \times (n-m)}$ be any full-column-rank left annihilator: $B_\perp(q)^\top B(q) = 0$. In IDA–PBC one seeks a feedback $u = u(q, \dot{q})$ such that the *closed loop* behaves as a target “simple mechanical” system

$$M_d(q) \ddot{q} + C_d(q, \dot{q}) \dot{q} + \nabla V_d(q) + D_d(q) \dot{q} = 0, \quad (3.15)$$

with $M_d(q) = M_d(q)^\top \succ 0$, $D_d(q) = D_d(q)^\top \succeq 0$, and $C_d(q, \dot{q})$ satisfying the same structural identity as C , i.e. $\dot{M}_d(q) - 2C_d(q, \dot{q})$ is skew-symmetric.

Projection of the matching condition. Combining (3.14) and (3.15), a feedback u that realizes (3.15) must satisfy

$$B(q) u = (M(q) - M_d(q)) \ddot{q} + (C(q, \dot{q}) - C_d(q, \dot{q}) - D_d(q)) \dot{q} + (\nabla V(q) - \nabla V_d(q)). \quad (3.16)$$

Since the right-hand side must lie in $\text{Im } B(q)$ for *all* $(q, \dot{q}, \ddot{q})$, the *matching equations* are obtained:

$$\begin{array}{ll} \text{(i)} & B_\perp(q)^\top (M(q) - M_d(q)) = 0, \\ \text{(ii)} & B_\perp(q)^\top (C(q, \dot{q}) - C_d(q, \dot{q}) - D_d(q)) \dot{q} = 0, \quad \forall \dot{q}, \\ \text{(iii)} & B_\perp(q)^\top (\nabla V(q) - \nabla V_d(q)) = 0. \end{array} \quad (3.17)$$

Equation (3.17) is simply the mechanical specialization, via $B_\perp^\top$, of the PH matching condition $G_\perp^\top [(J_d - R_d) \nabla H_d - (J - R) \nabla H] = 0$ written in $(q, \dot{q})$ coordinates.

Chapter 4

Machine Learning

Machine learning provides a set of computational methods for approximating complex input–output relationships from data. It formalizes the problem of learning as the search for a function within a prescribed class that minimizes a measure of prediction error, given finite samples from an unknown distribution. Modern learning approaches combine statistical principles with optimization and high-capacity models, most notably neural networks, to infer mappings that generalize beyond the observed data. In control and robotics, machine learning enables data-driven modeling, adaptation, and policy synthesis in scenarios where first-principles models are incomplete or uncertain. This chapter introduces the mathematical framework underlying supervised learning and empirical risk minimization, followed by the neural architectures used throughout the thesis.

4.1 Machine Learning Setup: Hypotheses, Risks, and Empirical Risk Minimization

We assume familiarity with linear algebra, multivariable calculus, and probability. Let $\mathcal{X} \subseteq \mathbb{R}^{d_x}$ be an input space and $\mathcal{Y}$ an output space (e.g., $\mathbb{R}^{d_y}$ for regression, a finite set for classification). A *hypothesis class* $\mathcal{F}$ is a set of measurable functions $f : \mathcal{X} \rightarrow \mathcal{Y}$, often parameterized by $\theta \in \Theta \subseteq \mathbb{R}^p$, written f_θ .

Given a loss $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_{\geq 0}$ and a data-generating distribution $\mathbb{P}$ over $(X, Y) \in \mathcal{X} \times \mathcal{Y}$, the *population risk* and *empirical risk* are

$$\mathcal{R}(\theta) := \mathbb{E}_{(X,Y) \sim \mathbb{P}}[\ell(f_\theta(X), Y)], \quad \widehat{\mathcal{R}}_N(\theta) := \frac{1}{N} \sum_{i=1}^N \ell(f_\theta(x_i), y_i). \quad (4.1)$$

Empirical Risk Minimization (ERM) selects $\widehat{\theta} \in \arg \min_{\theta \in \Theta} \widehat{\mathcal{R}}_N(\theta)$ and is typically solved with first-order methods.

Losses. Common losses include squared loss $\|f_\theta(x) - y\|^2$ (regression) and cross-entropy loss for classification. In control, multi-term objectives arise naturally (trajectory tracking, constraint penalties, physics residuals).

4.2 Neural Networks: Models and Architectures

Neural networks are parameterized, differentiable function approximators obtained by composing affine maps with fixed nonlinearities (activations). Given inputs $x \in \mathcal{X}$, a network $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ is specified by weights and biases θ arranged in layers; its parameters are learned by minimizing an empirical risk via gradient-based optimization and reverse-mode automatic differentiation (backpropagation) [30, 31]. A central motivation is *expressivity*: sufficiently wide networks with nonpolynomial activations are universal approximators on compact sets, enabling accurate approximation of smooth or piecewise-smooth mappings relevant to modeling, identification, and control [31, 32]. Generalization depends on the interplay between architecture (depth, width, activation), optimization, data, and regularization (e.g., weight decay, early stopping); in control-oriented contexts, architectural and loss-design choices encode inductive biases such as stability, smoothness, or consistency with physics. In the sequel, we briefly recall standard feedforward models and notation used later for physics-informed learning and safety-critical control.

Feedforward Multilayer Perceptrons (MLPs)

An L -layer MLP defines

$$z_0 = x, \quad z_\ell = \phi(W_\ell z_{\ell-1} + b_\ell), \quad \ell = 1, \dots, L-1, \quad (4.2)$$

$$f_\theta(x) = W_L z_{L-1} + b_L, \quad (4.3)$$

with parameters $\theta = \{W_\ell, b_\ell\}_{\ell=1}^L$. The activation ϕ is applied componentwise (e.g., ReLU, tanh, sigmoid). The model is differentiable almost everywhere and trained by gradient-based optimization [30, 31].

4.3 Expressivity: Universal Approximation Theorems

We present (i) a *full proof* of universality for a *sigmoidal activation* on compact sets and (ii) a general statement (Hornik, 1991) with a concise proof sketch.

Universality for Sigmoidal Activation

Let $\sigma : \mathbb{R} \rightarrow (0,1)$ be continuous and sigmoidal (i.e., $\lim_{t \rightarrow -\infty} \sigma(t) = 0$, $\lim_{t \rightarrow +\infty} \sigma(t) = 1$). Consider the single-hidden-layer class

$$\mathcal{S} := \left\{ x \mapsto \sum_{j=1}^m \alpha_j \sigma(w_j^\top x + \beta_j) \mid m \in \mathbb{N}, \alpha_j \in \mathbb{R}, w_j \in \mathbb{R}^{d_x}, \beta_j \in \mathbb{R} \right\}.$$

Theorem 1 (Universal Approximation for Sigmoids). *For any compact $K \subset \mathbb{R}^{d_x}$ and any $f \in C(K)$, the set $\mathcal{S}$ is dense in $C(K)$ under the uniform norm: for all $\varepsilon > 0$ there exists $s \in \mathcal{S}$ such that $\sup_{x \in K} |f(x) - s(x)| < \varepsilon$.*

4.4 Backpropagation and Reverse-Mode Automatic Differentiation

We formalize backpropagation as reverse-mode AD on a computational graph and give a concise proof (within one page) that it computes $\nabla_{\theta}\mathcal{L}$ with cost bounded by a small constant multiple of the forward pass [30, 33].

Computational Graph and Local Jacobians

Let a scalar loss $\mathcal{L} : \mathbb{R}^{n_v} \rightarrow \mathbb{R}$ be computed by a directed acyclic graph (DAG) with nodes $v_1, \dots, v_T$, where each node is $v_k = \psi_k(\text{pa}(v_k))$ with parents $\text{pa}(v_k) \subset \{v_1, \dots, v_{k-1}\}$. Parameters θ appear in some nodes. Define the *adjoint* (reverse sensitivity) $\bar{v}_k := \partial\mathcal{L}/\partial v_k$.

Theorem 2 (Reverse-mode AD = Backprop). *Let $\mathcal{L}$ be computed by a DAG as above. The recursion*

$$\bar{v}_T = 1, \tag{4.4}$$

$$\bar{v}_i = \sum_{k:v_i \in \text{pa}(v_k)} \bar{v}_k \cdot \frac{\partial v_k}{\partial v_i} \quad (\text{vector-Jacobian products}), \tag{4.5}$$

propagated in reverse topological order, yields $\bar{v}_i = \partial\mathcal{L}/\partial v_i$ for all i . Furthermore, for any parameter tensor θ , $\partial\mathcal{L}/\partial\theta$ is obtained by accumulating local terms $\bar{v}_k \cdot \frac{\partial v_k}{\partial\theta}$ at each node where θ appears. The arithmetic cost of this reverse sweep is bounded by a small constant factor of the forward evaluation cost.

Proof. Correctness. Induct on a reverse topological order. Base: the sink node $v_T = \mathcal{L}$ has $\bar{v}_T = 1$. Inductive step: assume $\bar{v}_k = \partial\mathcal{L}/\partial v_k$ is known for all successors v_k of v_i . By the multivariate chain rule,

$$\frac{\partial\mathcal{L}}{\partial v_i} = \sum_{k:v_i \rightarrow v_k} \frac{\partial\mathcal{L}}{\partial v_k} \frac{\partial v_k}{\partial v_i} = \sum_{k:v_i \rightarrow v_k} \bar{v}_k \frac{\partial v_k}{\partial v_i}.$$

Thus the recursion recovers $\bar{v}_i = \partial\mathcal{L}/\partial v_i$ for all i . For parameters θ , the same rule with $v_i \leftarrow \theta$ gives $\partial\mathcal{L}/\partial\theta = \sum_{k:\theta \in \text{pa}(v_k)} \bar{v}_k \partial v_k / \partial\theta$. □

Layerwise Gradients for MLPs (Matrix Form)

For (4.2)–(4.3), let $\mathcal{L} = \ell(f_{\theta}(x), y)$. Define $\delta_L := \nabla_f \ell \in \mathbb{R}^{d_y}$, and for $\ell = L - 1, \dots, 1$

$$\delta_{\ell} := (W_{\ell+1}^{\top} \delta_{\ell+1}) \odot \phi'(a_{\ell}), \quad a_{\ell} := W_{\ell} z_{\ell-1} + b_{\ell}, \tag{4.6}$$

$$\nabla_{W_{\ell}} \mathcal{L} = \delta_{\ell} z_{\ell-1}^{\top}, \quad \nabla_{b_{\ell}} \mathcal{L} = \delta_{\ell}. \tag{4.7}$$

This is the classical backpropagation recursion [30]. For mini-batches, replace outer products by sums over samples.

4.5 Optimization: Practical First-Order Methods

We give standard update rules without convergence theorems.

Gradient descent (GD). $\theta_{t+1} = \theta_t - \eta \nabla_{\theta} \widehat{\mathcal{R}}_N(\theta_t)$.

Mini-batch SGD. With batch $\mathcal{B}_t$, $\theta_{t+1} = \theta_t - \eta_t \frac{1}{|\mathcal{B}_t|} \sum_{i \in \mathcal{B}_t} \nabla_{\theta} \ell(f_{\theta}(x_i), y_i)$. Stochastic approximation foundations go back to [34]; we omit formal statements per scope.

Momentum / Nesterov. $v_{t+1} = \mu v_t + g_t$, $\theta_{t+1} = \theta_t - \eta v_{t+1}$, or Nesterov's look-ahead variant (common defaults $\mu \in [0.8, 0.99]$).

Adam. Biased first/second-moment accumulators $\{m_t, v_t\}$ with bias-correction [35]:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \quad \hat{m}_t = m_t / (1 - \beta_1^t), \quad (4.8)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t \odot g_t, \quad \hat{v}_t = v_t / (1 - \beta_2^t), \quad (4.9)$$

$$\theta_{t+1} = \theta_t - \eta \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon). \quad (4.10)$$

Common defaults: $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$.

Motivation and General Principle

Standard supervised learning minimizes a data-driven loss based on input–output pairs $\{(x_i, y_i)\}_{i=1}^N$. In many scientific and control problems, however, the system behavior is *partly known* from physical laws (e.g., differential equations), and data may be scarce or noisy. *Physics-Informed Neural Networks (PINNs)* incorporate this prior knowledge by enforcing the governing equations within minimum p -norm error loss functions.

Let the physical model be expressed as a differential operator

$$\mathcal{N}[u](t, x) = 0, \quad (4.11)$$

where $u : \mathcal{D} \subseteq \mathbb{R}^{d_t+d_x} \rightarrow \mathbb{R}^{d_y}$ is the unknown field (e.g., position, velocity, temperature). Instead of directly approximating u from data, one defines a neural network $u_{\theta}(t, x)$ and trains it such that: (i) it matches the available data points, and (ii) it satisfies (4.11) in a residual sense at collocation points sampled within $\mathcal{D}$.

PINN loss and training objective

Let $\mathcal{D}_{\text{data}}$ denote the set of observed points and $\mathcal{D}_{\text{phys}}$ a set of collocation points where the physics residual is enforced. The total loss is a weighted combination of data and physics terms:

$$\mathcal{L}(\theta) = \lambda_{\text{data}} \mathcal{L}_{\text{data}}(\theta) + \lambda_{\text{phys}} \mathcal{L}_{\text{phys}}(\theta), \quad (4.12)$$

where

$$\mathcal{L}_{\text{data}}(\theta) = \frac{1}{N_{\text{data}}} \sum_{(t_i, x_i) \in \mathcal{D}_{\text{data}}} \|u_\theta(t_i, x_i) - y_i\|^2, \quad (4.13)$$

$$\mathcal{L}_{\text{phys}}(\theta) = \frac{1}{N_{\text{phys}}} \sum_{(t_j, x_j) \in \mathcal{D}_{\text{phys}}} \|\mathcal{N}[u_\theta](t_j, x_j)\|^2. \quad (4.14)$$

The physics residual $\mathcal{N}[u_\theta]$ and its derivatives with respect to time or state are obtained by *automatic differentiation* (AD). Training proceeds by minimizing $\mathcal{L}(\theta)$ via standard gradient-based methods (e.g., Adam, L-BFGS).

Toy Example: Mass–Spring–Damper System

Consider a one-dimensional mass–spring–damper system with state $x = (q, \dot{q})$, governed by

$$m \ddot{q} + c \dot{q} + k q = 0, \quad (4.15)$$

where $m > 0$ is the mass, $c > 0$ the damping coefficient, and $k > 0$ the stiffness constant. Equation (4.15) can be rewritten as the first-order system

$$\dot{q} = v, \quad \dot{v} = -\frac{k}{m} q - \frac{c}{m} v.$$

PINN formulation. We define a neural network $q_\theta : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ that approximates the displacement trajectory $q(t)$. The physics residual is

$$r_\theta(t) := m \ddot{q}_\theta(t) + c \dot{q}_\theta(t) + k q_\theta(t), \quad (4.16)$$

where $\dot{q}_\theta$ and $\ddot{q}_\theta$ are computed via automatic differentiation of the network output with respect to time.

Given measured samples $\{(t_i, q_i)\}$ (possibly noisy) and a set of collocation points $\{t_j\}$, the PINN loss becomes

$$\mathcal{L}(\theta) = \frac{1}{N_{\text{data}}} \sum_{i=1}^{N_{\text{data}}} |q_\theta(t_i) - q_i|^2 + \frac{\lambda_{\text{phys}}}{N_{\text{phys}}} \sum_{j=1}^{N_{\text{phys}}} |r_\theta(t_j)|^2. \quad (4.17)$$

Minimizing (4.17) enforces both data fidelity and physical consistency with the dynamics (4.15). The resulting network simultaneously interpolates the observed data and satisfies the governing equation in regions without measurements.

Connection with Control and Identification

PINNs naturally extend to controlled systems $\dot{x} = f(x, u)$ by augmenting the residual with input-dependent terms, or by learning parameters of f . In this sense, PINNs bridge system identification and supervised learning: they embed known physics in the hypothesis class while retaining the universal approximation flexibility of neural networks. In the control context, such models can provide differentiable surrogates of physical systems for simulation, policy learning, or constraint enforcement in energy-based designs.

Chapter 5

Comparative Analysis of Energy Transfers and Performance in Safety-Critical Control using Control Barrier Functions

Control barrier functions (CBFs) are used in safety-critical control strategies, implementing a modification of a nominal control action to achieve invariance of a subset of the state space representing safe operating conditions. In this chapter we perform a comparative study involving existing safety-critical CBF designs, including energy-based CBFs and Exponential CBFs. The analysis, performed both theoretically and on a benchmark obstacle avoidance task, provides insights into how these CBFs affect energy transfers and the overall performance of the closed-loop system, highlighting benefits and limitations of each approach. To validate our analysis, we conduct software simulations on a 3R planar robot and a 7-DoF robotic manipulator, complemented by experimental evaluations on a physical robotic platform.

5.1 Introduction

Robotic applications rely on a nominal controller, typically designed to satisfy performance and stability requirements. In practice, performance objectives must be paired with safety specifications, often representing static or dynamic obstacles to be avoided during task execution [10, 36, 37]. This view of safety-critical control applications has led to the concept of *safety filters* [36], computational units that transform a nominal control input into one that i) minimally modifies the controller and ii) ensures safety. A popular technique for implementing safety filters is *control barrier functions* (CBFs) [9, 10], an algorithm which aims to modify a nominal control input to achieve forward invariance of a safe set. In some particular scenarios, safety specifications may conflict with the main task to be executed, leading to deterioration of performance, previously guaranteed by the nominal controller.

CBFs have been proposed for generic systems [13, 14] and robotics [38, 39], often managing positional constraints and generalizing Artificial Potential Fields [40]. Their use cases are diverse, including ensuring joint limits [41] and collision avoidance in multi-agent [42]. Recent research

merges CBFs with energy-based techniques to guarantee safety and closed-loop passivity, including port-Hamiltonian frameworks [15], energy tanks [43, 44], time-varying CBFs [45], and optimization-based controllers [46]. This includes foundational work on passivity-preserving CBFs [47], with extensions to switched systems [48] and human-robot interaction [49].

This chapter presents a systematic comparative study of two principal classes of CBFs for torque-controlled robotic manipulators: Energy-Based CBFs (EB-CBFs) [38, 39] and Exponential CBFs (ECBFs) [14]. The former is tailored to mechanical systems, enabling safety filter design without knowledge of Coriolis and centrifugal terms, while the latter is a general-purpose solution for control-affine systems that requires full model knowledge.

First, we generalize the result of [47] by proving that the EB-CBF from [39] preserves closed-loop passivity. Second, we empirically analyze how these CBFs affect performance and energy transfer in a benchmark obstacle avoidance task. The results reveal a critical trade-off, showing that despite their formal passivity guarantees, EB-CBFs can induce significant task performance degradation compared to ECBFs.

The remainder of the chapter is organized as follows. In Sec. 5.2, the background on robotic manipulator dynamic models and CBFs is reviewed. Sec. 5.3 describes the main contribution of this chapter, i.e., the development of the energetic analysis of each CBF. In Sec. 5.4, numerical simulations are performed on a 3R planar robot to evaluate the proposed approach in a simplified setting. Sec. 5.5 extends this analysis to a more complex scenario, presenting both simulation results and real-world experiments on the 7-DOF KUKA LWR manipulator. Finally, in Sec. 5.6 conclusions are drawn together with future works.

Notation. The set $\mathcal{C}^1$ refers to the set of continuously differentiable functions whose first derivative is continuous. Given a general dynamical system, the set $\mathcal{D}$ and the set $\mathcal{U}$ refer to the *state space* and *input space*, respectively. The expression $E(q, \dot{q})$ represents the total mechanical energy of a system. The function $\mathbb{1}_X(x) = \begin{cases} 1 & \text{if } x \in X \\ 0 & \text{if } x \notin X \end{cases}$ denotes the indicator function.

5.2 Background

We refer to [10], [9], [20], [38] regarding CBF for references that fully cover the presented background and to [21], [22] for preliminaries on modeling and control of robotic manipulators.

5.2.1 Control Barrier Functions

Consider the input-affine nonlinear system:

$$\dot{x} = f(x) + g(x)u, \quad (5.1)$$

where $x \in \mathcal{D} \subseteq \mathbb{R}^n$, $u \in \mathcal{U} \subseteq \mathbb{R}^m$, and f and g are continuously differentiable. A Lipschitz continuous controller ensures existence and uniqueness of solutions.

Definition 5. *The relative degree r of a differentiable function $b : \mathbb{R}^n \rightarrow \mathbb{R}$ with respect to (5.1) is the number of times it must be differentiated along the system dynamics until u explicitly appears.*

Control barrier functions represent a technique to guarantee forward invariance of a set $\mathcal{C}$, normally called *safe set*: the control goal is to design a state feedback $u(x) = k(x)$ for system (1) resulting in the closed-loop system $\dot{x} = f_{cl}(x) = f(x) + g(x)k(x)$ such that

$$\forall x(0) \in \mathcal{C} \implies x(t) \in \mathcal{C} \forall t > 0. \quad (5.2)$$

The safe set $\mathcal{C}$ is defined according to the class of CBF, which can be either *Zeroing* or *Reciprocal*:

Zeroing CBFs (Z-CBF)

The safe set $\mathcal{C}$ is built as the superlevel set of a continuously differentiable function $h : \mathcal{D} \rightarrow \mathbb{R}$, i.e.,

$$\mathcal{C} = \{x \in \mathcal{D} : h(x) \geq 0\}. \quad (5.3)$$

The function $h(x)$ is then a *control barrier function* (CBF) on $\mathcal{D}$ if $\frac{\partial h}{\partial x}(x) \neq 0$, $\forall x \in \partial\mathcal{C}$ and

$$\sup_{u \in \mathcal{U}} [L_f h(x) + L_g h(x)u] \geq -\alpha(h(x)) \quad (5.4)$$

where $L_f h(x) = \frac{\partial h}{\partial x} f(x) \in \mathbb{R}$, $L_g h(x) = \frac{\partial h}{\partial x} g(x) \in \mathbb{R}^{1 \times m}$, for all $x \in \mathcal{D}$ and some *extended class* $\mathcal{K}$ function α ¹.

The following key result connects the existence of such CBF to forward invariance of the corresponding safe set.

Lemma 1. [10]: Let $h(x)$ be a CBF on $\mathcal{D}$ for (1). Any locally Lipschitz controller $u(x) = k(x)$ such that $L_f h(x) + L_g h(x)k(x) \geq -\alpha(h(x))$ provides forward invariance of the safe set $\mathcal{C}$. Additionally, the set $\mathcal{C}$ is asymptotically stable on $\mathcal{D}$.

The way controller synthesis induced by CBFs are implemented is to use them as *safety filters*, transforming a nominal state-feedback control input $u_{\text{nom}}(x)$ into a new state-feedback control input $u^*(x)$ in a minimally invasive fashion in order to guarantee forward invariance of $\mathcal{C}$. In practice, the following Quadratic Program (QP) is solved:

$$\begin{aligned} u^*(x) &= \arg \min_{u \in \mathcal{U}} \|u - u_{\text{nom}}(x)\|^2 \\ \text{s.t. } & L_f h(x) + L_g h(x)u \geq -\alpha(h(x)). \end{aligned} \quad (5.5)$$

The transformation of the nominal control input $u_{\text{nom}}(x)$ in $u^*(x)$ by solving (5.5) is denoted as safety-critical control, or safety-critical filtering. Finally, the closed form solution to (5.5) is presented in the following lemma.

Lemma 2. [9]: Let $h(x)$ be a Z-CBF on a set $\mathcal{D}$ for system (1), and assume $\mathcal{U} = \mathbb{R}^n$ and $L_g h(x) \neq 0$, $\forall x \in \mathcal{D}$. Define $\Psi_Z(x; u) = L_f h(x) + L_g h(x)u + \alpha(h(x))$, where $\alpha(\cdot)$ is a class $\mathcal{K}$ -function, and $E_Z = \{x \in \mathcal{D}, u \in \mathcal{U} \mid \Psi_Z(x; u) < 0\}$, representing the set where the safety-critical filtering is applied. A closed-form solution for (5.5) is given by $u^*(x) = u_{\text{nom}}(x) + u_{\text{safe}}(x)$,

¹A function $\alpha : (-b, a) \rightarrow (-\infty, \infty)$ with $a, b > 0$, which is continuous, strictly increasing, and $\alpha(0) = 0$.

where

$$u_{\text{safe}}(x) = -\mathbb{1}_{E_Z}(x, u_{\text{nom}}) \frac{L_g h(x)^\top}{\|L_g h(x)\|^2} \Psi_Z(x; u_{\text{nom}}) \quad (5.6)$$

Reciprocal CBFs (R-CBFs)

Reciprocal Control Barrier Functions (R-CBFs) were introduced in [11] and are characterized by the fact that they become unbounded at the boundary of the safe set $\mathcal{C}$, hence the term "barrier." Their formal definition is provided as follows.

Given a continuously differentiable function $h : \mathcal{D} \rightarrow \mathbb{R}$, define its superlevel set as $\mathcal{C} \subseteq \mathcal{D}$. The function $B : \text{Int}(\mathcal{C}) \rightarrow \mathbb{R}$ is referred to as an R-CBF defined for the set $\mathcal{C}$ if it satisfies the following conditions for all $x \in \text{Int}(\mathcal{C})$, namely:

$$\frac{1}{\alpha_1(h(x))} \leq B(x) \leq \frac{1}{\alpha_2(h(x))} \quad (5.7)$$

$$\inf_{u \in \mathbb{R}^n} [L_f B(x) + L_g B(x)u] < \alpha_3(h(x)), \quad (5.8)$$

$$L_g B(x) = 0 \implies L_f B(x) < \alpha_3(h(x)) \quad (5.9)$$

where α_1 , α_2 , and α_3 are *class- $\mathcal{K}$* functions[17]. If the R-CBF in (5.7)-(5.9) exists, forward invariance of set $\mathcal{C}$ is guaranteed, as stated in the following theorem.

Theorem 3. [11]: *Given a set $\mathcal{C} \subset \mathcal{D}$, defined as the superlevel set of a continuously differentiable function $h : \mathcal{D} \rightarrow \mathbb{R}$ as in (5.3), if there exists a barrier function $B(x)$ (5.7)-(5.9), then $\mathcal{C}$ is forward invariant.*

Similarly to (5.5), the nominal state-feedback control input $u_{\text{nom}}(x)$ is filtered in a minimally invasive fashion in order to guarantee forward invariance of $\mathcal{C}$. The optimal control action $u^*(x)$ is found solving the following QP[19]:

$$\begin{aligned} u^*(x) &= \arg \min_{u \in \mathcal{U}} \|u - u_{\text{nom}}(x)\|^2 \\ \text{s.t. } & L_f B(x) + L_g B(x)u \leq \alpha_3(h(x)) \end{aligned} \quad (5.10)$$

where $\mathcal{U} = \mathbb{R}^m$. The closed form solution of (5.10) is defined in the following lemma, similarly to the Z-CBF case.

Lemma 3. [11]: *Let $B(x)$ be a R-CBF on $\mathcal{D}$ for (1) and assume $\mathcal{U} = \mathbb{R}^m$ and $L_g B(x) \neq 0$, $\forall x \in \mathcal{D}$. Define $\Psi_R(x; u) = L_f B(x) + L_g B(x)u - \alpha_3(h(x))$. A closed-form solution for (6) is given by $u^*(x) = u_{\text{nom}}(x) + u_{\text{safe}}(x)$, where*

$$u_{\text{safe}}(x) = -\mathbb{1}_{E_R}(x, u_{\text{nom}}) \frac{L_g h(x)^\top}{\|L_g h(x)\|^2} \Psi_R(x; u_{\text{nom}}) \quad (5.11)$$

and $E_R = \{x \in \mathcal{D}, u \in \mathcal{U} \mid \Psi_R(x; u) > 0\}$ represents the set where the safety-critical filtering is applied.

For simplicity, we denote $\mathbb{1}_{E_Z}(x, u_{\text{nom}})$ and $\mathbb{1}_{E_R}(x, u_{\text{nom}})$ as $\mathbb{1}_{E_Z}$ and $\mathbb{1}_{E_R}$. Although both CBFs serve similar purposes, their structures differ: Z-CBFs are defined on the entire domain $\mathcal{D}$, whereas R-CBFs are restricted to the safe set $\mathcal{C}$, where the barrier diverges at the boundary. Consequently, Z-CBFs remain well-defined even outside $\mathcal{C}$, ensuring not only invariance but also attractivity of the safe set (Theorem 1). Their similarities and differences under a common application will be discussed through the proposed energy-based analysis.

Having introduced the general theory of CBFs, we next turn to the case of robotic manipulators. Their mechanical structure can be framed into the input-affine formulation in (5.1) and allows for an energy-based analysis of the safety-critical actions induced by CBFs.

5.2.2 Model of a Fully Actuated Robot Manipulator

A common instance of (5.1) is a serial, fully actuated n -DOF robot manipulator with configuration $q \in Q$ and velocity $\dot{q} \in T_q Q$, where $T_q Q$ is the tangent space at q , containing all possible velocity vectors. Its dynamics are described by

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = u, \quad (5.12)$$

where $M(q)$ is the inertia matrix, $C(q, \dot{q})$ captures Coriolis/centrifugal forces, $G(q)$ is gravity, and u is the actuation.

Defining the state $x = (q^\top, \dot{q}^\top)^\top \in TQ$, where TQ is the tangent bundle of Q (5.12) can be written in affine form (5.1) with $x \in \mathbb{R}^{2n}$. Analysis in this work is performed on these local coordinates. System (5.12) possesses properties central to our analysis, stated next.

Property 1. *The inertia matrix $M(q)$ is symmetric, positive definite, and bounded for any $q \in \mathbb{R}^n$.*

Property 2. *If the matrix $C(q, \dot{q})$ is defined through Christoffel symbols, then $\dot{M}(q) - 2C(q, \dot{q})$ is a skew symmetric matrix. This is equivalent to $\dot{M}(q) = C(q, \dot{q}) + C^\top(q, \dot{q})$.*

Property 3. *The time derivative of kinetic energy $K_e(q, \dot{q}) = \frac{1}{2}\dot{q}^\top M(q)\dot{q}$ along solutions of (5.12) is given by $\dot{K}_e(q, \dot{q}) = \dot{q}^\top [u - G(q)]$.*

5.2.3 Energy Based Analysis of CBFs on Mechanical Systems

In the domain of safety-critical control applied to mechanical systems, the expression for the power injected by the CBF control action u_{safe} in (5.6),(5.11) into system (5.12) can be analyzed by writing the power balance equation for mechanical energy. Consider the general expression of the mechanical energy $E(q, \dot{q}) = K_e(q, \dot{q}) + V(q)$, where $K_e(q, \dot{q})$ is the kinetic energy and $V(q)$ is the potential energy. Using property 3 and $\frac{\partial V}{\partial q} = G(q)$ it is easily seen that the power balance equation takes the following expression:

$$\dot{E} = \dot{K}_e + \dot{V} = \dot{q}^\top u_{\text{nom}} + \dot{q}^\top u_{\text{safe}} \quad (5.13)$$

The last term in the power balance corresponds to the power injected into system (5.12) by the safety-critical control, which can be further analyzed as follows.

Proposition 1. *Consider system (5.12) controlled with either a Z-CBF (5.6) or a R-CBF (5.11). The power injection due to safety-critical control component $\dot{q}^\top u_{\text{safe}}$ results in*

$$P_{h,\alpha} := \dot{q}^\top u_{\text{safe}} = -\mathbb{1}_{E_i} \frac{\dot{q}^\top M^{-1}(q) \frac{\partial h}{\partial \dot{q}}^\top}{\|M^{-1}(q) \frac{\partial h}{\partial \dot{q}}^\top\|^2} \Psi_i. \quad (5.14)$$

where $i = Z, R$ in case of ZCBF and RCBF, respectively.

Proof. Consider the mechanical system (5.12) with state $x = (q^\top, \dot{q}^\top)^\top \in TQ$, rewritten in the input-affine state space model (5.1). The corresponding input vector field is defined as $g(q, \dot{q}) = [0_{n \times n}, M^{-1}(q)]^\top$. The expression for $P_{h,\alpha}$ is obtained substituting the Lie derivative $L_g h(q, \dot{q}) = \frac{\partial h}{\partial \dot{q}} M^{-1}(q)$ in the closed form expressions (5.6)-(5.11), yielding the result. $\square$

A special case is analyzed in the following lemma from [47], when the power injected in the system takes negative values.

Lemma 4. [47]: *Let system (5.12) controlled with a nominal control action $u = u_{\text{nom}}(q, \dot{q})$ result in a closed-loop system stable in the sense of Lyapunov with respect to a desired Lyapunov function $S_{cl}(q, \dot{q})$ of the form:*

$$S_{cl}(q, \dot{q}) = K_e(q, \dot{q}) + V_d(q), \quad (5.15)$$

where $V_d(q) \geq 0$ is a positive definite function of the configuration variables $q \in Q$. If the safety-filtering action induced by a CBF $h(q, \dot{q})$ is such that $P_{h,\alpha} < 0$, then $S_{cl}(q, \dot{q})$ is still a Lyapunov function for closed-loop system subject to the safety-filtering action.

In particular, we focus our attention on the class of mechanical systems controlled by a PD control law with gravity compensation of the form

$$u_{\text{nom}} = -D\dot{q} - Pe + G(q) \quad (5.16)$$

where $P \in \mathbb{R}^n$ and $D \in \mathbb{R}^n$ are positive definite matrices and $e = q - q_d$. The nominal control law (5.16) guarantees that the closed-loop energy function $S_{cl}(q, \dot{q})$ is a Lyapunov Function for system (5.12) with nominal equilibrium $(q_d^\top, 0^\top)^\top$ and $V_d(q) = \frac{1}{2}\|e\|_P^2$, $P > 0$. For this class of systems the energy balance takes the form:

$$\dot{S}_{cl} = -\dot{q}^\top D\dot{q} + P_{h,\alpha}. \quad (5.17)$$

Equation (5.17) can be obtained by substituting expression (5.16) and $\frac{\partial V_d}{\partial q} = e^\top P$ into the following energy balance equation and :

$$\dot{S}_{cl} = \dot{q}^\top [u_{\text{nom}} + u_{\text{safe}} - G(q)] + \frac{\partial V_d}{\partial q} \dot{q}. \quad (5.18)$$

In the following section, we further analyse the CBF from an energetic perspective, comparing the main classes of such control technique.

5.3 CBFs for safety-critical control of Robotic Manipulators

Applying control barrier functions to robotic manipulators poses challenges beyond the general affine case. In mechanical systems, safety constraints are often expressed as functions of configuration variables, $\phi(q) : Q \rightarrow \mathbb{R}$ with $Q \subseteq \mathbb{R}^n$, leading to a safe set of the form:

$$\mathcal{S} = \{q \in Q \mid \phi(q) \geq 0\}. \quad (5.19)$$

However, for second-order dynamics such as torque-controlled manipulators (5.12), a barrier function of the form (5.19) yields relative degree greater than one, i.e. $L_g\phi(q) = 0_{1 \times n}$. As a result, the standard CBF formulation (5.5) cannot be applied directly, since the control input u does not appear in the constraint. To overcome this limitation, several works [14, 39, 40] propose defining barrier functions $h(q, \dot{q}) : TQ \rightarrow \mathbb{R}$ that explicitly depend on both position and velocity, ensuring that $L_g h(q, \dot{q}) \neq 0$ when the safety condition becomes active.

We analyze these CBFs by examining the power injected into the nominal closed-loop system by the safety-critical action. The first two types [38, 39] fall under Stability Preserving CBFs [50], where $P_{h,\alpha} \leq 0$ guarantees stability. By contrast, CBFs such as [14] yield $P_{h,\alpha}$, indefinite in sign, providing no stability guarantee. In what follows, we discuss the resulting trade-offs between task performance and stability.

5.3.1 Energy Based Z-CBFs

An interesting type of Z-CBF which deserves some discussion is the one developed in [38], that works by subtracting the kinetic energy of the system to the function $\phi(q)$, scaled by a positive constant $\beta \in \mathbb{R}^+$:

$$h(q, \dot{q}) = \beta\phi(q) - K_e(q, \dot{q}). \quad (5.20)$$

The CBF (5.20) is such that its superlevel set $\bar{\mathcal{S}}$ can be expressed as

$$\bar{\mathcal{S}} = \{(q, \dot{q}) \in TQ \mid h(q, \dot{q}) \geq 0\}, \quad (5.21)$$

and is always contained in $\mathcal{S}$ (5.19), with $\lim_{\beta \rightarrow \infty} \bar{\mathcal{S}} = \mathcal{S}$. The safety-critical control action (5.6) takes the following expression:

$$u_{\text{safe}}(x) = \mathbb{1}_{E_Z} \frac{\dot{q}}{\|\dot{q}\|^2} \Psi_Z(q, \dot{q}, u_{\text{nom}}). \quad (5.22)$$

Property 4. [47] *The sign of the instantaneous power injection (5.14) obtained by applying CBF (5.20) on system (5.12) is negative semi-definite:*

$$P_{h,\alpha} = \mathbb{1}_{E_Z} \Psi_Z(q, \dot{q}, u_{\text{nom}}) \leq 0, \quad (5.23)$$

CBF Expression	Barrier-Type	Injected power	Stabi Pre- serv- ing	Related works	ID
$\beta\phi(q) - K_e(q, p)$	Z- CBF	$\mathbb{1}_{E_Z}\Psi(q, \dot{q}, u_{\text{nom}})$	Yes	[38], [47], [50]	I
$\frac{1}{\phi(q)} + \frac{K_e(q, \dot{q})}{1+K_e(q, \dot{q})}$	R- CBF	$-\mathbb{1}_{E_R}(1 + K_e)^2\Psi(q, \dot{q}, u_{\text{nom}})$	Yes	[39]	II
$\phi(q) + \gamma\frac{\partial\phi}{\partial q}\dot{q}$	Z- CBF	$-\mathbb{1}_{E_Z}\frac{\dot{q}^\top M^{-1}(q)\frac{\partial\phi}{\partial q}^\top}{\gamma\ M^{-1}(q)\frac{\partial\phi}{\partial q}^\top\ ^2}\Psi(q, \dot{q}, u_{\text{nom}})$	No	[18],[51], [13],[14]	III

Table 5.1: Comparison of CBF formulations, barrier type, injected power, and stability-preserving properties.

where $\Psi_Z(q, \dot{q}, u_{\text{nom}})$ takes the form:

$$\Psi_Z(q, \dot{q}, u_{\text{nom}}) = \dot{q}^\top \left[\beta \frac{\partial\phi}{\partial q}^\top + g(q) - u_{\text{nom}} \right] + \alpha(h(q, \dot{q})). \quad (5.24)$$

Note that due to the non-positive sign of the injected power (5.23), this CBF type preserves the stability of the overall system. The expression in (5.20) is a valid CBF since $\dot{q} = 0$ implies $\Psi_Z(q, \dot{q}, u_{\text{nom}}) = \alpha(h(q, \dot{q})) > 0$, hence it does not activate the safety-critical constraint [38].

5.3.2 Energy Based R-CBF

In this section we calculate the power injected by safety-critical controller induced by the R-CBF presented in [39], defined as:

$$h(q, \dot{q}) = \frac{1}{\phi(q)} + \frac{K_e(q, \dot{q})}{1 + K_e(q, \dot{q})}. \quad (5.25)$$

We present a property for R-CBFs, analogous to Property 4 for Z-CBFs.

Property 5. *The sign of the instantaneous power injection (5.14) obtained by applying CBF (5.25) on system (5.12) is negative semi-definite:*

$$P_{h,\alpha} = -\mathbb{1}_{E_R}(1 + K_e)^2\Psi_R \leq 0. \quad (5.26)$$

where $\Psi_R(q, \dot{q}, u_{\text{nom}})$ takes the form:

$$\Psi_R(q, \dot{q}, u_{\text{nom}}) = -\frac{L_f h(q, \dot{q})}{h(q)^2} + \frac{\dot{q}^\top (u_{\text{nom}} - G(q))}{(1 + K_e(q, \dot{q}))^2} - \alpha(h(q, \dot{q})) \quad (5.27)$$

Proof. The result in (5.26) follows from the relation $\frac{\partial h}{\partial \dot{q}} = \frac{\dot{q}^\top M(q)}{(1+K_e)^2}$, which, when substituted into (5.14), yields the stated expression. Equation (5.27) is obtained by applying proposition (3) to $\dot{h}(q, \dot{q}, u)$. $\square$

5.3.3 Exponential CBFs

Another approach to define Z-CBFs for safety-critical control of mechanical systems is represented by [18], which introduced the concept of a safety index, further investigated in [14] as Exponential CBFs, and generalized to time varying systems in [13] as Higher Order CBFs. In the domain of mechanical systems, a valid form for an Exponential CBF is:

$$h(q, \dot{q}) = \phi(q) + \gamma \frac{\partial \phi}{\partial q} \dot{q}, \quad (5.28)$$

with $\gamma \in \mathbb{R}_{>0}$. Thanks to the differentiability assumption $\phi \in \mathcal{C}^1$ and to the fact that the $\gamma \geq 0$, it holds that the safe set $\bar{\mathcal{C}} = \{q, \dot{q} \in TQ : h(q, \dot{q}) \geq 0\}$ defines a positively invariant set in $\mathcal{S}$ (5.19), i.e., $\bar{\mathcal{C}} \in \mathcal{S}$ [18]. Considering that $\frac{\partial h}{\partial \dot{q}} = \gamma \frac{\partial \phi}{\partial q}$, the power injected in the system (5.14) by the safety-critical filtering action (5.28) takes the following expression:

$$P_{h,\alpha} = -\mathbb{1}_{E_Z} \frac{\dot{q}^\top M^{-1}(q) \frac{\partial \phi}{\partial q}^\top}{\gamma \|M^{-1}(q) \frac{\partial \phi}{\partial q}^\top\|^2} \Psi_Z(q, \dot{q}, u_{\text{nom}}) \quad (5.29)$$

where function Ψ_Z takes the form:

$$\Psi_Z = \alpha \phi(q) + (1 + \gamma \alpha) \frac{\partial \phi}{\partial q} \dot{q} + \gamma \frac{\partial \phi}{\partial q} \ddot{q} + \gamma \dot{q}^\top \frac{\partial^2 \phi}{\partial q^2} \dot{q}. \quad (5.30)$$

Note that the power expression in (5.29) is indefinite in sign and, in general, does not preserve stability. In particular, stability guarantees are lost if simultaneously $\Psi_Z < 0$ and $\dot{q}^\top M^{-1}(q) \frac{\partial \phi}{\partial q}^\top > 0$.

It must be noted that Exponential CBFs may lead to undesired performance as well. In particular, in the following lemma we prove that invariant sets different from the desired equilibrium can arise when using Exp-CBFs on system (5.12), nominally controlled with (5.16).

Lemma 5. Consider system (5.12) under control law (5.16) with the CBF filter (5.28). Let $Q_\phi = \{q \in Q : \phi(q) = 0\}$ and define

$$\Omega = \{q \in Q_\phi, \dot{q} = 0, \lambda \in \mathbb{R} \mid \Psi_Z < 0, Pe = \lambda M^{-1}(q) \frac{\partial \phi}{\partial q}^\top\}. \quad (5.31)$$

Then Ω is invariant.

Proof. For $\mathbb{1}_{E_Z} < 0$, the closed-loop dynamics are

$$M(q)\ddot{q} = C(q, \dot{q})\dot{q} + G(q) + u_{\text{nom}} + u_{\text{safe}}. \quad (5.32)$$

At $(q, \dot{q}) = (\tilde{q}, 0)$ with $\tilde{q} \in \Omega$, $u_{\text{nom}} = -G(q) - Pe$ and

$$u_{\text{safe}} = \mathbb{1}_{E_Z} \frac{M^{-1}(q) \frac{\partial \phi}{\partial q}^\top \frac{\partial \phi}{\partial q} M^{-1}(q)}{\|M^{-1}(q) \frac{\partial \phi}{\partial q}^\top\|^2} Pe.$$

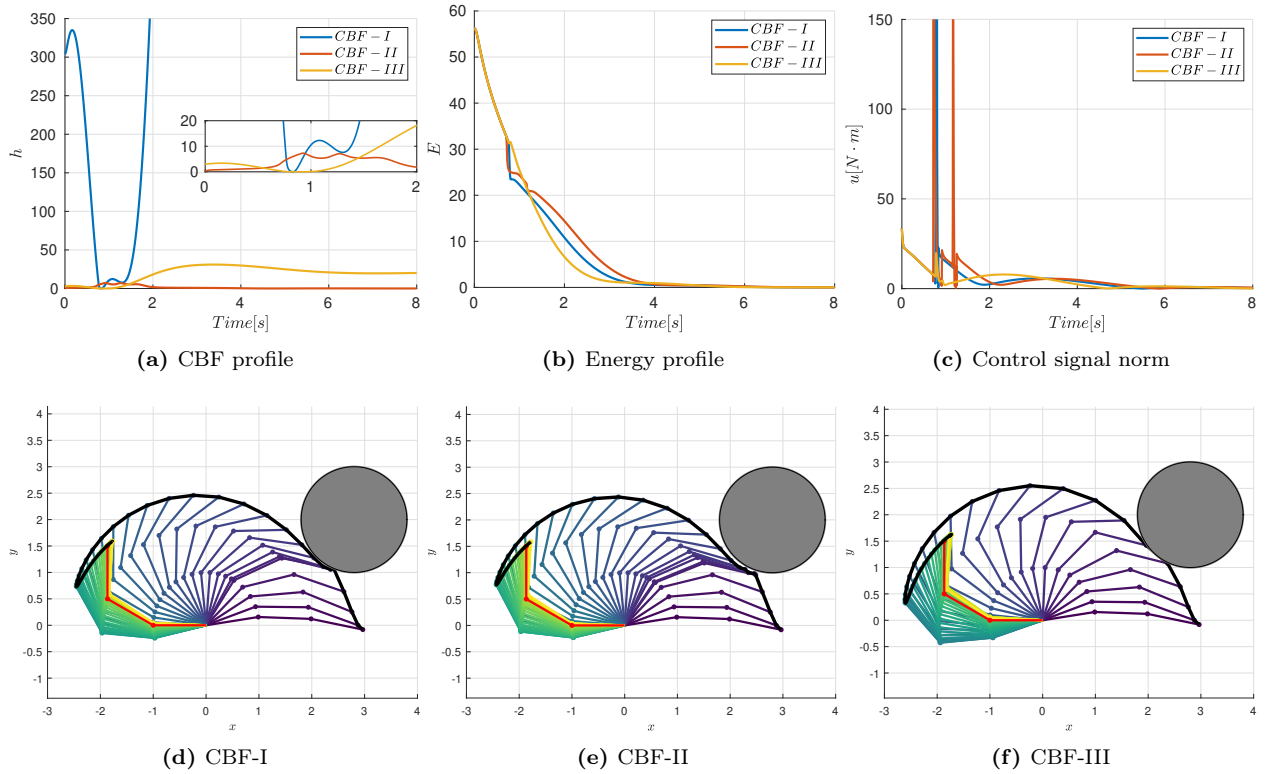


Figure 5.1: 3R simulation results for *Case A*. Top row: overall system quantities. Bottom row: individual barrier functions.

Substituting into (5.32) gives

$$\|M^{-1}(q)\frac{\partial\phi}{\partial q}^\top\|^2 Pe = \left(M^{-1}(q)\frac{\partial\phi}{\partial q}^\top \frac{\partial\phi}{\partial q} M^{-1}(q)\right) Pe. \quad (5.33)$$

Since the matrix is symmetric rank-1, its only nonzero eigenvalue is $\lambda = \|M^{-1}(q)\frac{\partial\phi}{\partial q}^\top\|^2$ with eigenvector $v = M^{-1}(q)\frac{\partial\phi}{\partial q}^\top$. Hence the condition in (5.31) holds and Ω is invariant. $\square$

The previous theorem characterizes conditions under which stationary points may arise, namely when $\dot{q} = 0$ and $\phi(q) = 0$. In this case, the obstacle geometry may cause the proportional control term to balance the action of the safety filter. A full analysis of solutions to (5.31) is beyond the scope of this work, though such results could provide guarantees on the absence of undesired stationary points.

5.4 Simulations: 3R planar robot

In this section a comparative analysis regarding performance and power transfer between different CBFs through simulations on a planar 3R robotic manipulator is performed. A nominal PD control law is designed to move the robot from its initial to a nominal joint configuration. A 2D virtual obstacle, depicted as a grey circle in the plots, is introduced along the robot's trajectory. To account for the obstacle, we define the function $\phi(q)$, used in all the CBF expressions, as follows

$$\phi(q) = [x_c - x_{EE}(q)]^2 + [y_c - y_{EE}(q)]^2 - r^2 \quad (5.34)$$

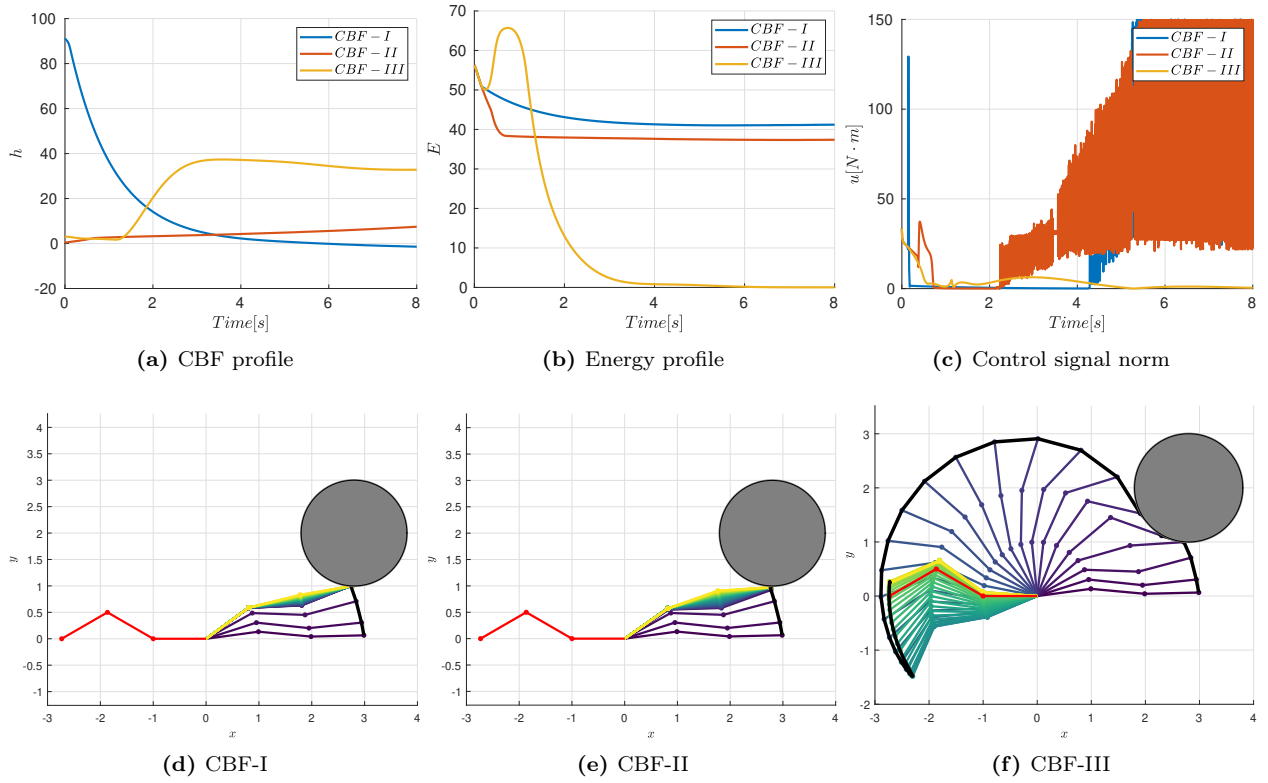


Figure 5.2: 3R simulation results for *Case B*. Top row: overall system quantities. Bottom row: individual barrier functions.

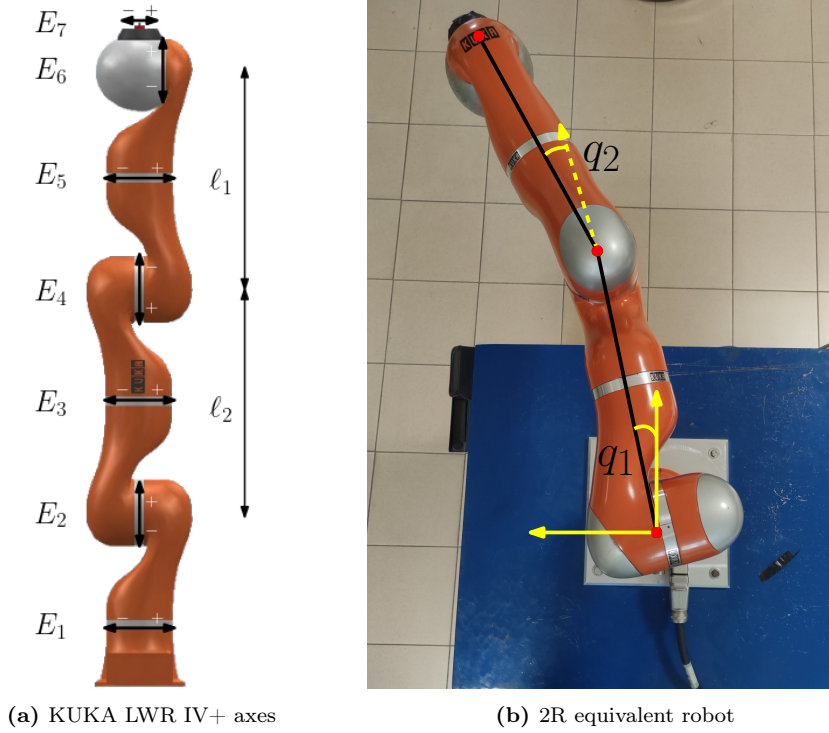
	Parameter	P	D	α	β	γ
Value	<i>Case A</i>	10	10	1	100	100
	<i>Case B</i>	10	10	1	30	1

Table 5.2: Parameters for the nominal controller and the CBFs in software simulations on a 3R planar robot.

where x_c and y_c are the obstacle center coordinates, r is the obstacle radius and $x_{EE}(q)$ and $y_{EE}(q)$ denote the coordinates of the robot manipulator end-effector. Moreover, the desired configuration is depicted in red in the plots.

Two scenarios, *Case A* and *Case B*, illustrate different system behaviors. In *Case A*, the desired joint configuration is $q_d = (\pi, -\pi/6, -\pi/3)^\top$, corresponding to a positive value for the third joint. Conversely, *Case B* is defined with $q_d = (\pi, -\pi/6, \pi/3)^\top$, where the third joint assumes a negative value. The parameter values for the nominal controller and the CBFs used in both cases are summarized in Table 5.2. All software simulations are performed with a time discretization of $T_s = 10^{-5}s$ and the dynamics is integrated using Runge-Kutta-Fehlberg method.

The stability-preserving nature of energy-based CBFs (CBF-I and CBF-II), reported in Fig. 5.1d and Fig. 5.1e, yields high control effort during obstacle avoidance, as highlighted by the control peaks in Fig. 5.1c. On the other hand, the Exponential CBF allows for a larger reconfiguration to overcome the obstacle, as reported in Fig. 5.1f, thus avoiding excessive peaks in the control signal. Moreover, in Fig. 5.1a the CBF profile of the three approaches is reported, where a box highlights the time evolution of the first two seconds to show better that none of the CBF violates the safety requirement.

**Figure 5.3:** KUKA LWR IV+ Robot.

In Fig. 5.2a- 5.2f, the simulation results of *Case B* are reported, where the limitations of energy-based CBFs are visible. Specifically, both CBF-I and CBF-II fail to avoid the obstacle, see Fig. 5.2d and Fig. 5.2e respectively. The task requires a large reconfiguration that violates the stability-preserving condition, permissible under CBF-III, as demonstrated by the peak in the closed-loop energy in Fig. 5.2b. With large reconfigurations, energy-based CBFs perform poorly. Such CBFs can only dissipate power, therefore when the reconfiguration requires an increase of the value of potential function $V_d(q)$, the robot significantly slows down during the task execution. This effect is clearly visible in Fig. 5.2d and Fig. 5.2e where the robot halts in front of the obstacle, unable to complete the regulation task. When $\dot{q} \approx 0$, these CBFs exhibit chattering from numerical errors, as shown in Fig. 5.2c. Conversely, as demonstrated in *Case A*, CBF-III successfully avoids the obstacle and reaches the desired configuration as shown in Fig. 5.2f.

5.5 Simulations and Experiments: KUKA LWR IV+

The experimental setup uses a fully actuated KUKA LWR-IV+ at the Robotics Laboratory, Università di Roma “La Sapienza”. This lightweight, seven-DoF anthropomorphic manipulator is equipped with encoders for joint positions and torque sensors on each joint. The robot is controlled via a KR-C2 unit with the Fast Research Interface (FRI) [52], which provides access to joint positions and torques.

A nominal joint reference q_d is generated by integrating the robot dynamics under the virtual control input u , derived from either CBF-I (5.20), CBF-II (5.25), or CBF-III (5.28). This reference is sent to the FRI, which applies the corresponding torques. Joint positions are measured directly, while velocities are obtained numerically from position data and filtered to reduce noise.

Parameter	P	D	α	β	γ
Value	10	10	10	100	9e-5

Table 5.3: Simulation parameters for the nominal controller and the CBFs in the simulation on the KUKA LWR in the single obstacle scenario.

5.5.1 Simulation: single obstacle scenario

This section presents the simulation results for the KUKA LWR in addressing the collision avoidance through the CBF formalism. The simulation parameters for the nominal controller and the various CBFs are provided in Table 5.3.

To evaluate collision avoidance capabilities, a virtual obstacle modeled as a 3D sphere is placed along the robot's trajectory. To account for the obstacle, the spatial formulation of the function $\phi(q)$, introduced in (5.34), is employed. The resulting expression is given by:

$$\phi(q) = [x_c - x_{EE}(q)]^2 + [y_c - y_{EE}(q)]^2 + [z_c - z_{EE}(q)]^2 - r^2, \quad (5.35)$$

where x_c , y_c , and z_c represent the coordinates of the obstacle center, r is the obstacle radius, and $x_{EE}(q)$, $y_{EE}(q)$, and $z_{EE}(q)$ denote the coordinates of the robot's end-effector as a function of the joint configuration q .

Fig. 5.4 illustrates a comparison of the different control strategies applied to the KUKA LWR IV+. As shown in Fig. 5.4f, CBF-III is the only safety-critical approach that successfully enables the robot to avoid the virtual obstacle. This is confirmed by the region highlighted in the box of Fig. 5.4a, where the safety function satisfies $h(q) > 0$ during the whole simulation. In contrast, CBF-I fails to ensure safety, resulting in a collision with the obstacle, represented by the CBF entering the region where $h(q) < 0$.

CBF-II, owing to its stability-preserving nature, is unable to compute a feasible control input that avoids the obstacle without violating the energy constraints of the closed-loop system. As a result, the robot comes to a halt in front of the obstacle, remaining on the boundary of the safe set, where $h(q) = 0$. Additionally, both CBF-I and CBF-II exhibit a chattering effect in the control signal, as depicted in Fig. 5.4c. This behavior arises from the challenging objectives of maintaining safety while preserving system stability.

Furthermore, it is important to note that the computational complexity of the control law derived using CBF-III may be higher compared to the other two approaches, primarily due to the evaluation of the terms $C(q, \dot{q})$ and $\frac{\partial^2 \phi}{\partial q^2}$. However, these expressions can be symbolically computed offline, enabling efficient numerical evaluation during real-time task execution, not compromising the execution speed of method-III relative to the other two approaches.

Fig. 5.4d- 5.4f provide a visual representation of the KUKA robot, including both its initial and desired configurations. In particular, Fig. 5.4f shows that, under CBF-III, the robot successfully avoids the obstacle by passing over it, an action that yields an increase in the system's closed-loop energy, shown in Fig. 5.4b.

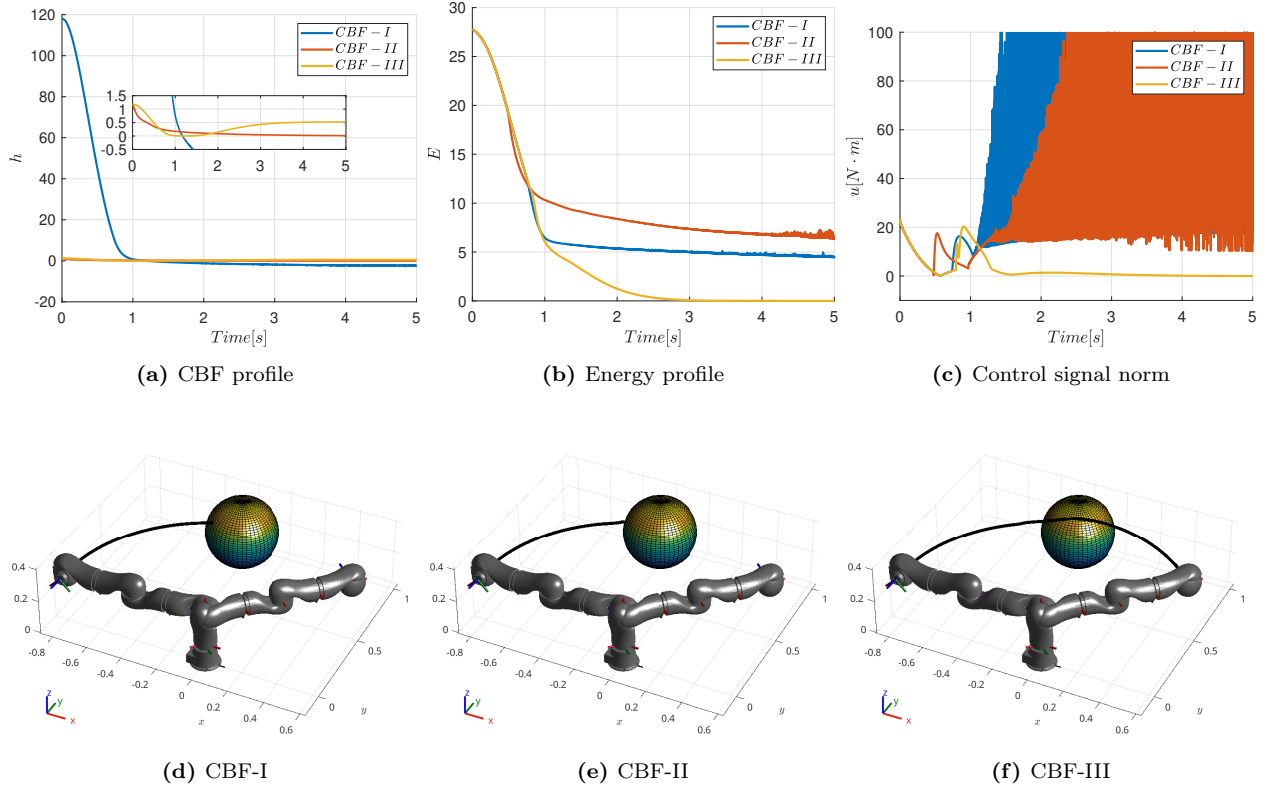


Figure 5.4: KUKA 7-DoF simulation results: single obstacle scenario. Top row: performance metrics. Bottom row: end-effector trajectories for the three CBF formulations.

5.5.2 Simulation: multiple obstacles scenario

This section considers the same simulation setup as before but with a harder collision-avoidance task. Two spherical obstacles of different sizes are placed along the KUKA trajectory, requiring simultaneous handling of multiple safety constraints. Following [53], a single CBF is built using Boolean logic to encode complex specifications. For each obstacle, a function ϕ_i (as in (5.35)) is defined from its center and radius, and the overall safe set $\mathcal{C}$ is obtained by combining $\mathcal{C}_1$ and $\mathcal{C}_2$ through a logical AND, implemented via the min operator:

$$\mathcal{C} = \mathcal{C}_1 \cap \mathcal{C}_2 = \{q \in \mathbb{R}^n : \min_{i=1,2} h_i(q) \geq 0\}. \quad (5.36)$$

However, the use of the min operator may lead to a function that is not continuously differentiable with respect to q , and therefore not a valid CBF. To address this, [53] proposes a smooth approximation of the min function, resulting in the following CBF formulation:

$$h(q) = -\frac{1}{\kappa} \ln \left(e^{-\kappa h_1(q)} + e^{-\kappa h_2(q)} \right), \quad (5.37)$$

where $\kappa > 0$ is a smoothing parameter. In our implementation, we choose $\kappa = 300$.

For clarity, Fig. 5.6 shows simulation results for CBF-III only, as it is the only method that successfully completes the obstacle-avoidance task in all cases. Figs. 5.6d–5.6e depict the robot trajectory around two obstacles of different sizes from different viewpoints. Note that the CBF in (5.37) accounts only for the end-effector position, not the volume of the last link. Fig. 5.6a shows the CBF profile, confirming $h(q) > 0$ throughout the entire simulation and demonstrating effective

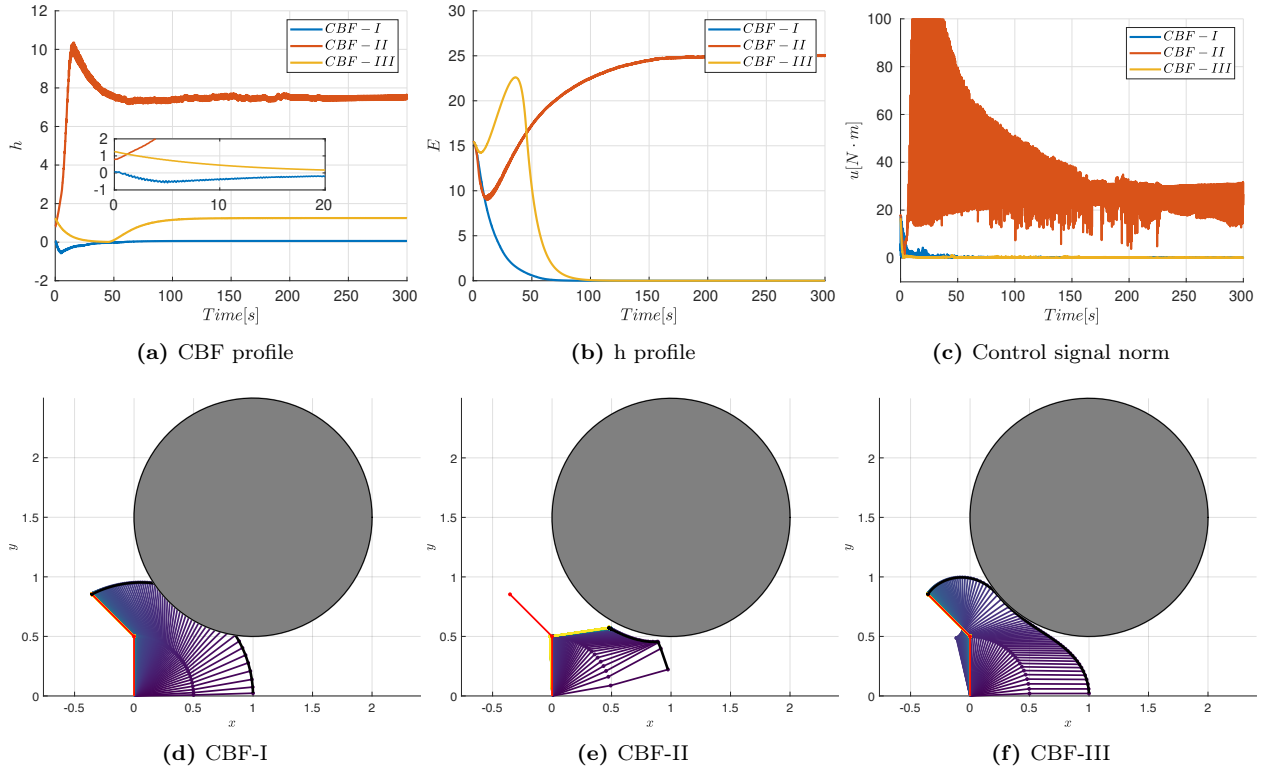


Figure 5.5: KUKA experimental results. Top row: performance metrics. Bottom row: end-effector trajectories for the three CBF formulations.

Parameter	P	D	α	β	γ
Value	10	20	0.8	0.05	0.1

Table 5.4: Parameters for the nominal controller and the CBFs in the experiment.

obstacle avoidance while reaching the desired configuration.

5.5.3 Experimental setup

To provide further insight into the behavior of the proposed control strategies, this section presents an experimental scenario based on the KUKA LWR IV+. While the previous sections employed the full 7-DoF KUKA LWR, here the analysis focuses on an equivalent 2R planar robot that preserves the essential features of the collision avoidance task while simplifying the dynamics.

The axes E_i ($i = 2, 3, 5, 6, 7$) in Fig. 5.3a remain fixed, simplifying the KUKA robot to a 2R planar model (Fig. 5.3b). The controller and CBF parameters are given in Table 5.4.

Fig. 5.5 presents the comparative analysis of the experimental results on the KUKA LWR IV+, when the different CBF approaches are used. Consistent with the simulations on the 3R planar robot in Section 5.4 and on the KUKA LWR in Section 5.5, due to chattering effects, the CBF-I method fails to ensure safety, resulting in $h(q) < 0$ during the time window when the robot *virtually* collides with the obstacle, as reported in Fig. 5.5a. Furthermore, a zoom on the first 20 seconds of simulation is reported to show that all the CBF profiles are positive at $t = 0$, as the robot starts in a safe configuration. Although CBF-II is theoretically proven to preserve stability, as shown in (5.26), the closed-loop energy profile increases while avoiding the obstacle, see Fig. 5.5b. This increase is

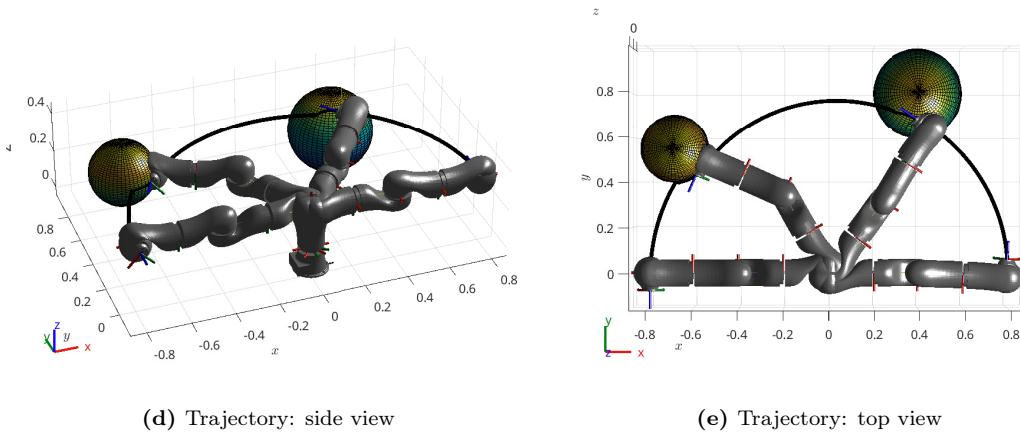
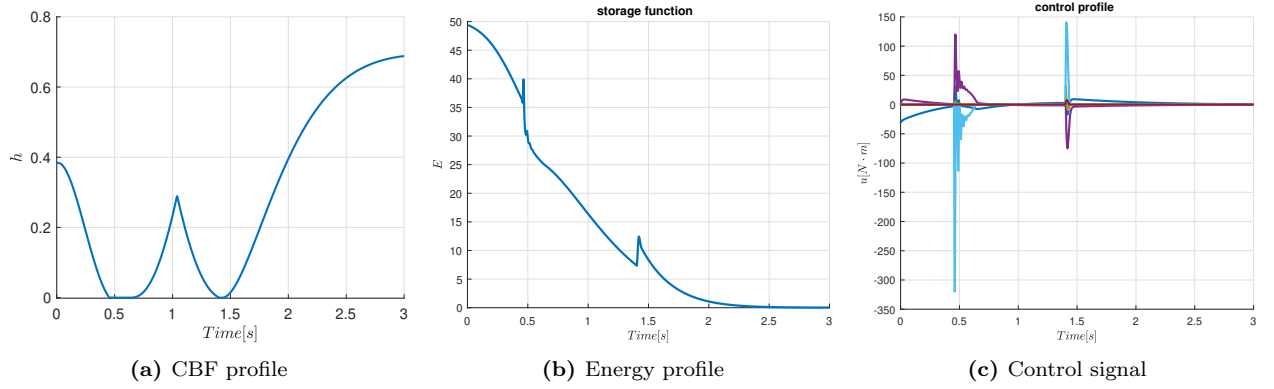


Figure 5.6: KUKA simulation plots: multiple obstacle scenario.

due to the robot ending in a state where $\dot{q} \approx 0$, resulting in an undesirable chattering behaviour in the control action leading to an increase in the potential function $V_d(q)$, as shown in *Case B* of the 3R simulation.

Moreover, similarly to software simulation results, CBF-III is the only technique that successfully achieves safety in avoiding the obstacle, while guiding the robot to its desired configuration. The fact that CBF-III is not stability preserving, allows the robot to fold in a configuration with negative q_2 values to avoid the obstacle, as highlighted by the increase in the total closed-loop energy in Fig. 5.5b. In conclusion, Fig. 5.5 shows results consistent with those in Fig. 5.4 and with *Case B* in Fig. 5.2. The robot controlled with CBF-I violates the safety constraint by entering the virtual obstacle (see Fig. 5.5d), while CBF-II causes the robot to stall in front of the obstacle (see Fig. 5.5e), again due to the inability to increase the closed-loop energy to perform the necessary motion. CBF-III, on the other hand, successfully guides the robot around the obstacle, as shown in Fig. 5.5f.

5.6 Conclusions

This chapter presented a comparative study of Energy-Based and Exponential Control Barrier Function (CBF) designs. Theoretical analysis and practical evaluation in an obstacle avoidance task provided insights into their effect on energy transfer and system performance. The analysis was validated through software simulations on 3R and 6-DoF robotic manipulators, and experiments on a physical 2R robot, confirming the theoretical predictions and real-world applicability of the different CBF designs. Future work will analyze energy-injecting CBFs for achieving desired closed-loop behavior [50] and extend the analysis to time-varying obstacle avoidance [12].

Chapter 6

A Physics-Informed Neural Networks based method for Interconnection and Damping Assignment Passivity-Based Control

The Interconnection and Damping Assignment Passivity-Based Control (IDA-PBC) is an advanced nonlinear control strategy founded on the passivity and stability principles of port-Hamiltonian (PH) systems. This approach provides a powerful tool for controlling nonlinear systems by the assignment of a desired energetic structure to the system, establishing a new equilibrium point and achieving asymptotic stability through energy dissipation. IDA-PBC has proven particularly effective in addressing the stabilization problem of underactuated systems, where the number of control inputs is insufficient to directly control all degrees of freedom. The main challenge in this context lies in solving complex, nonlinear Partial Differential Equations (PDEs) that arise from the desired energy assignment process. This challenge is often mitigated by imposing several assumptions that identify classes of systems for which these equations are solvable. To address these limitations, a methodological approach aimed at providing a more general framework, employing Physics-Informed Neural Networks (PINNs) for solving the nonlinear PDE related to the kinetic energy and the definition of a customized loss function, is proposed. Furthermore, it is demonstrated that, under a simple structure of the potential energy function and with no need of solving the potential energy PDE, stabilization is achieved by a suitable tuning of the damping matrix, provided the kinetic energy PDE is solved up to a small error. The proposed methodology is verified through numerical simulations.

6.1 Introduction

IDA-PBC has been successfully applied to various domains, such as mechanical, electrical, and biological systems [54]. Its foundations are derived from Passivity-Based Control (PBC) theory, which has been extensively studied and which is widely applied, for example in problems of coordination and synchronization of multi-agent systems represented through directed and undirected graphs [55]. This technique relies on the concept of passivity, which refers to the system's ability to store

or dissipate energy, and is strongly connected to Lyapunov stability [7] [24]. IDA-PBC is formulated for systems described in PH form, a class of passive systems that strictly contains Euler-Lagrange (EL) models [25]. Without resorting to linearization or decoupling procedures, that often compromise the native EL structure of the system, IDA-PBC not only ensures that the Closed-Loop (CL) system remains passive and stable but also preserves a meaningful energetic structure. This is guaranteed by the fact that IDA-PBC endows the CL system with a desired PH structure, where the desired energy function serves as a Lyapunov function for the desired equilibrium [27]. When dealing with underactuated systems, IDA-PBC relies on the possibility of solving a set of PDEs also known as "Matching Conditions" that identify the admissible energy functions that can be assigned to the CL. Therefore, if these PDEs can be solved, the original system and the desired CL, along with the passivity and stability properties, are said to match. Authors in [28] proposed to separate the set of nonlinear PDEs associated with the kinetic energy from the linear PDEs related to the potential energy. Solving such PDEs, especially the kinetic one, is, in general, challenging and it represents the main obstacle and limitation of the IDA-PBC methodology in terms of its practical applicability. In [56], a method for reducing the PDEs to a simple set of nonlinear ODEs for a class of mechanical systems is proposed, while in [29] the set of PDEs are explicitly solved for systems with underactuation degree one through a suitable parametrization of the admissible energy functions that can be assigned to the CL. The analytical approach is therefore closely interconnected to the identification of a class of systems for which the PDEs can be solved explicitly, becoming unfeasible for complex nonlinear systems.

The growth and rapid advancement of Artificial Intelligence (AI) and, particularly, Neural Networks (NNs), have significantly impacted the field of automatic control. In recent years, one of the most promising developments in this field has been the emergence of PINNs, combining the computational power of Deep Learning with the physical laws that govern dynamical systems [57]. This approach is particularly valuable for the modeling and control of complex systems, where traditional methods may encounter difficulties due to nonlinearity or high dimensionality. In [58] authors employed PINNs for solving forward and inverse problems involving nonlinear PDEs, while in [59] it is stressed how such data-driven solutions are effective for solving nonlinear PDEs even though limited data are available. Recent developments viewed the construction of Variational PINNs (VPINNs) and demonstrated a reduction of training costs, also improving accuracy compared to PINNs [60]. In [61], authors proposed a data-driven solution for solving the IDA-PBC problem, by estimating through NNs the PDE solutions and then employing linear control techniques to asymptotically stabilize the system.

The present work introduces a novel approach that integrates PINNs with IDA-PBC technique for nonlinear underactuated systems, avoiding to resort to external control inputs or restrictions on the specific class of systems for the stabilization task. After decoupling the kinetic from the potential PDE, a set of PINNs, equipped with a customized loss function, is employed to solve the kinetic PDE by estimating the desired inertia matrix over a reduced state space, defined according to the open-loop energy of the system, improving learning. It is then discussed that given a simple structure for the desired potential energy function, an appropriate design of the damping matrix allows to stabilize the system arbitrarily close to the desired equilibrium, provided that the kinetic PDE is solved up to a small error. This analysis has the advantage of avoiding the search for a solution of the potential PDEs for each desired equilibrium that has to be stabilized.

In the remainder of the chapter, Section 6.2 presents the theoretical foundations of IDA-PBC control and gives an overview of PINNs, Section 6.3 describes the proposed PINNs-based IDA-PBC control framework, 6.4 illustrates the result of numerical simulations to validate the proposed approach over a nonlinear underactuated system, and 6.5 draws the conclusions and outlines future research works.

6.2 Background Theory

6.2.1 IDA-PBC

Let us consider a generic underactuated mechanical system with generalized coordinates $q \in \mathbb{R}^n$ and momenta $p = M\dot{q} \in \mathbb{R}^n$, with inertia $M \in \mathbb{R}^{n \times n}$, expressed in PH form:

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \begin{bmatrix} \nabla_q H(q, p) \\ \nabla_p H(q, p) \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} u, \quad (6.1)$$

with input matrix $G \in \mathbb{R}^{n \times m}$, with $m < n$, and control $u \in \mathbb{R}^m$. The system (6.1) reflects the structural properties of Lagrangian systems, with associated Hamiltonian function derived by the difference of the kinetic and potential energy:

$$H(q, p) = \frac{1}{2} p^\top M^{-1} p + V(q) \in \mathbb{R}. \quad (6.2)$$

IDA-PBC control methodology follows two main steps, the *Energy Shaping* (ES) and *Damping Assignment* (DI).

Energy Shaping

The primary objective of IDA-PBC methodology is to assign a desired PH structure equipped with a new Hamiltonian function (shaping) and a new equilibrium $(q^*, 0)$ to the CL system. Therefore, the CL energy is characterized by a new inertia matrix $M_d = M_d^T \in \mathbb{R}_{>0}^{n \times n}$ and potential function $V_d(q) \in \mathbb{R}$, resulting into:

$$H_d(q, p) = \frac{1}{2} p^\top M_d^{-1} p + V_d(q), \quad (6.3)$$

with the requirement that the (6.3) has an isolated minimum at $(q^*, 0)$. To this end, the following conditions must hold:

$$\begin{cases} V_d(q^*) = 0 \\ \nabla_q V_d(q^*) = 0 \\ \nabla_q^2 V_d(q^*) > 0. \end{cases} \quad (6.4)$$

To endow the CL system with the (6.3), it is customary to refer to the *Target Dynamics* expressed as:

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & M^{-1} M_d \\ -M_d M^{-1} & J_2 \end{bmatrix} \begin{bmatrix} \nabla_q H_d \\ \nabla_p H_d \end{bmatrix}, \quad (6.5)$$

where $J_2 \in \mathbb{R}^{n \times n}$ is a skew-symmetric matrix. The final ES control $u_{ES} \in \mathbb{R}^m$ is obtained by equating the (6.1) to the (6.5). Being the equalities for the unactuated dynamics satisfied by construction, it is obtained that:

$$Gu_{ES} = \nabla_q H(q, p) - M_d M^{-1} \nabla_q H_d(q, p) + J_2 \nabla_p H_d(q, p). \quad (6.6)$$

When dealing with fully actuated systems, G is invertible and, consequently, the ES control law can be uniquely determined by inverting the input matrix. In underactuated systems, however, G has full column rank $\rho(G) = m < n$, meaning that u_{ES} can only influence terms in the span of G . This leads to the following set of constraints that define the admissible energy functions that can be assigned to the CL, expressed in form of nonlinear PDEs:

$$G^\perp [\nabla_q H(q, p) - M_d M^{-1} \nabla_q H_d(q, p) + J_2 \nabla_p H_d(q, p)] = 0, \quad (6.7)$$

where $G^\perp \in \mathbb{R}^{s \times n}$ is the left annihilator of G , with $s = n - m$. By exploiting the degrees of freedom provided by J_2 , if a solution for the unknowns M_d and $V_d(q)$ is obtained, then the ES control is defined as:

$$u_{ES} = G^\# [\nabla_q H(q, p) - M_d M^{-1} \nabla_q H_d(q, p) + J_2 \nabla_p H_d(q, p)], \quad (6.8)$$

where $G^\# = (G^\top G)^{-1} G^\top$ is the left pseudoinverse of G . The terms related to the kinetic energy of the (6.7) are given by:

$$G^\perp [\nabla_q (p^\top M^{-1}(q) p) - M_d(q) M^{-1}(q) \nabla_q (p^\top M_d^{-1}(q) p) + 2J_2 M_d^{-1}(q) p] = 0, \quad (6.9)$$

where it is introduced the notation $M(q)$ and $M_d(q)$, denoting that these matrices are functions of q , to motivate the presence of the first two terms in (6.9). Once the (6.9) is solved and the desired inertia is fixed, then the potential PDE:

$$G^\perp [\nabla_q V(q) - M_d M^{-1} \nabla_q V_d(q)] = 0, \quad (6.10)$$

has to be solved in $V_d(q)$ such that the (6.4) holds. In [28], an equivalent expression for the Kinetic PDE is given by using the fact that, being $S(q)$ a symmetric matrix, $\nabla_q [p^\top S(q) p] = [\nabla_q (S(q) p)]^\top p$. Moreover, the matrix J_2 is reparametrized through a set of free design matrices $U_k(q) = -U_k^\top(q) \in \mathbb{R}^{n \times n}$, $k = 1, \dots, n$ as:

$$2J_2 = \sum_{k=1}^n U_k p_k, \quad (6.11)$$

obtaining that the (6.9) can be equivalently rewritten as:

$$G^\perp \{ [\nabla_q (M_{(\cdot, k)}^{-1}(q))]^\top - M_d M^{-1} [\nabla_q (M_{d(\cdot, k)}^{-1}(q))]^\top + U_k M_d^{-1} \} = 0, \quad (6.12)$$

allowing to eliminate the dependence on p , yielding a set of k PDEs.

Damping Assignment

Once the ES stage is completed, an additional input $v \in \mathbb{R}^m$ is used for asymptotically stabilize the new equilibrium $(q^*, 0)$. By computing the derivative of the energy along system's trajectories, and recalling the fact that the skew-symmetric structure of the (6.5) inside a quadratic term vanishes, it is obtained that:

$$\dot{H}_d(q, p) = L_g H_d(q, p) v = \left[\nabla_q H_d(q, p) \nabla_p H_d(q, p) \right]^\top \begin{bmatrix} 0 \\ G \end{bmatrix} v, \quad (6.13)$$

where $L_g H_d(q, p)$ denotes the $H_d(q, p)$ Lie derivative operator applied to g . Therefore, by PBC theory, the system is passive and lossless with respect to the passive output $y_p = G^\top \nabla_p H_d(q, p)$, according to the Kalman-Yakubovich-Popov Lemma. The control that makes $H_d(q, p)$ negative semi-definite, acting as a Lyapunov function for the system and, thus, asymptotically stabilizing $(q^*, 0)$, is known as DI control, defined as:

$$u_{DI} = -K_v y_p = -K_v G^\top \nabla_p H_d(q, p), \quad (6.14)$$

with symmetric gain matrix $K_v \in \mathbb{R}_{>0}^{m \times m}$.

6.2.2 PINNs

PINNs offer a data-driven framework for solving PDEs by embedding physical laws directly into the learning process. Let us consider a general PDE:

$$\mathcal{N}[u(\mathbf{x})] = 0, \quad \mathbf{x} \in \Omega, \quad (6.15)$$

with boundary conditions:

$$\mathcal{B}[u(\mathbf{x})] = g(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega, \quad (6.16)$$

where $\mathcal{N}$ is a differential operator governing the system dynamics, $\mathcal{B}$ represents the boundary operator encoding constraints and where Ω is the domain.

PINNs approximate a solution $u_\theta(\mathbf{x})$ by minimizing the following loss function:

$$\mathcal{L}(\theta) = \lambda_r \|\mathcal{N}[u_\theta(\mathbf{x})]\|_\Omega^2 + \lambda_b \|\mathcal{B}[u_\theta(\mathbf{x})] - g(\mathbf{x})\|_{\partial\Omega}^2 + \lambda_d \|u_\theta(\mathbf{x}_i) - u_i\|^2, \quad (6.17)$$

where $\lambda_r, \lambda_b, \lambda_d$ are regularization coefficients balancing the contributions of the residual loss, boundary loss and an optional loss, with $\{(\mathbf{x}_i, u_i)\}$ representing the dataset. The employment of Automatic Differentiation (AD) to compute derivatives appearing in $\mathcal{N}[u_\theta(\mathbf{x})]$ allows to eliminate discretization errors and ensuring a continuous approximation.

6.3 Main Result

Let us recall the (6.10) and consider the term:

$$s(q) = G^\perp \nabla_q V(q) \in \mathbb{R}^{s \times 1}. \quad (6.18)$$

The above function is only determined by the open-loop potential energy and, through a straightforward steady-state analysis, it can be deduced that $s(q)$ defines the set of admissible equilibria, defined as:

$$\mathcal{Z}_r^* = \{q_r^* = \Pi_r q^* \in \mathbb{R}^l \subseteq \mathbb{R}^n \mid s(q_r^*) = 0\}, \quad (6.19)$$

where $\Pi_r : \mathbb{R}^n \rightarrow \mathbb{R}^l$ is a projection operator introduced to denote that $s(q)$ may only depend on some components q_r of the state. By the (6.18), the PDE in (6.10) can be expressed as:

$$\gamma^\top(q) \nabla_q V_d(q) = s(q_r), \quad (6.20)$$

where $\gamma(q) = M^{-1}(q) M_d^{-1}(q) (G^\perp)^\top \in \mathbb{R}^{n \times s}$. The (6.20) can be written, respectively, as the sum of the heterogeneous and homogeneous characteristics as:

$$\gamma^\top(q) \nabla_q (\tilde{V}_d(q) + \bar{V}_d(q)) = s(q_r). \quad (6.21)$$

The original potential PDE is, thus, transformed in the following set of linear PDEs:

$$\begin{aligned} \gamma^\top(q) \nabla_q \tilde{V}_d(q) &= s(q_r), \\ \gamma^\top(q) \nabla_q \bar{V}_d(q) &= 0, \end{aligned} \quad (6.22)$$

with desired potential energy $V_d(q) = \tilde{V}_d(q) + \bar{V}_d(q)$.

Let us assume $l = 1$, underactuation degree $s = 1$ and constant inertia M . Then, it is reasonable to assume that also M_d is only function of q_r [29] and the first equation in (6.22) can be solved as:

$$\gamma_r(q_r) \nabla_{q_r} \tilde{V}_d(q_r) = s(q_r) \Rightarrow \nabla_{q_r} \tilde{V}_d(q_r) = \frac{s(q_r)}{\gamma_r(q_r)}, \quad (6.23)$$

where, if $\gamma_r(q_r^*) \neq 0 \Rightarrow \nabla_{q_r} \tilde{V}_d(q_r^*) = 0$. Notice that this is a scalar solution because of the assumptions made above and $\nabla_q \tilde{V}_d(q)$ will be a 0-vector except for the r -th term. Those assumptions can be relaxed by defining:

$$\nabla_{q_{\bar{r}}} \tilde{V}_d(q_{\bar{r}}) = (\gamma^\top(q_{\bar{r}}))^\# s(q_{\bar{r}}), \quad (6.24)$$

where $(\gamma^\top(q_{\bar{r}}))^\#$ denotes the left pseudoinverse of $\gamma^\top(q_{\bar{r}})$. In this case, the admissible equilibria $q_{\bar{r}} \in \mathbb{R}^{\bar{l}}$, with $1 \leq \bar{l} \leq n$, are defined by the set $\mathcal{Z}_{\bar{r}}^*$. To match conditions (6.4) for a generic underactuated system of degree $s \geq 1$, remains to guarantee that:

$$\nabla_{q_{\bar{r}}}^2 \tilde{V}_d(q_{\bar{r}}^*) > 0. \quad (6.25)$$

Being $\gamma(q_{\bar{r}})$ a function of $M_d(q_{\bar{r}})$, it is proposed to embed the (6.25) inside the Kinetic PDE optimisation problem:

Equation(6.12)

subject to $M_d = M_d^\top > 0$,

$$U_k = -U_k^\top,$$

$$\nabla_{q_{\bar{r}}}^2 \tilde{V}_d(q_{\bar{r}}^*) > 0. \quad (6.26)$$

The above nonlinear optimization problem does not admit a closed form solution, so it is proposed to solve it for the candidate solutions $M_d(q_{\bar{r}})$ and the set of matrices $U_k(q_{\bar{r}})$ by leveraging the universal approximation theorem of NNs over a sampled state space $\mathcal{X}$. To estimate a symmetric, positive definite matrix, it is proposed to use Cholesky decomposition, directly defining the PINN $R^{\theta_R} : \mathbb{R}^{\bar{l}} \rightarrow \mathbb{R}^{\frac{n(n-1)}{2}}$ and arranging the PINN's output vector into a lower triangular matrix $\bar{R}^{\theta_R}(q_{\bar{r}}) \in \mathbb{R}^{n \times n}$. The inverse of the desired inertia is then obtained as:

$$M_d^{\theta_R^{-1}}(q_{\bar{r}}) = \bar{R}^{\theta_R}(q_{\bar{r}}) \bar{R}^{\theta_R \top}(q_{\bar{r}}) + \epsilon_{M_d} I_n, \quad (6.27)$$

where ϵ_{M_d} is a small positive number. The design choice of directly estimating the inverse of the desired Inertia arises from the fact that the inverse of a symmetric and positive definite matrix is again a symmetric and positive definite matrix. This choice is additionally motivated by the (6.12), in which the gradient operator appears directly on the inverse of M_d . The latter has been computed as:

$$\begin{aligned} \nabla_{q_{\bar{r}}} M_d^{\theta_R^{-1}}(q_{\bar{r}}) &= \nabla_{q_{\bar{r}}} \left[\bar{R}^{\theta_R}(q_{\bar{r}}) \bar{R}^{\theta_R \top}(q_{\bar{r}}) \right] = \\ &= \left(\nabla_{q_{\bar{r}}} \bar{R}^{\theta_R}(q_{\bar{r}}) \right) \bar{R}^{\theta_R \top}(q_{\bar{r}}) + \bar{R}^{\theta_R}(q_{\bar{r}}) \left(\nabla_{q_{\bar{r}}} \bar{R}^{\theta_R \top}(q_{\bar{r}}) \right). \end{aligned} \quad (6.28)$$

Similarly, a second PINN $\mathcal{U}^{\theta_U} : \mathbb{R}^{\bar{l}} \rightarrow \mathbb{R}^{n^2}$ is employed to approximate the set of skew-symmetric matrices:

$$U_k^{\theta_U}(q_{\bar{r}}) = \bar{U}^{\theta_U^k}(q_{\bar{r}}) - \bar{U}^{\theta_U^k \top}(q_{\bar{r}}), \quad (6.29)$$

where the PINN's output $\mathcal{U}^{\theta_U}$ is arranged into $\bar{U}^{\theta_U^k}(q_{\bar{r}}) \in \mathbb{R}^{n \times n}$ and where θ_U^k are the set of weights and biases related to the k -th matrix U_k . Let $\theta = \{\theta_R, \theta_U\} = \{\theta_R, \theta_U^1, \dots, \theta_U^n\}$, the learning problem aimed at solving the (6.12) is defined, through the design of a customized loss $\mathcal{L}_\theta$, as:

$$\min_{\theta} \mathcal{L}_\theta = \sum_{q_{\bar{r}} \in \mathcal{Q}_{\bar{\nabla}}} \mathcal{L}_\theta = \sum_{q_{\bar{r}} \in \mathcal{Q}_{\bar{\nabla}}} \left(\left(\sum_{k=1}^n \mathcal{L}_\theta'^k \right) + \alpha \mathcal{L}_\theta'' \right), \quad (6.30)$$

where $\mathcal{Q}_{\bar{\nabla}}$ is the state obtained from the sampling over $\mathcal{X}$ and $\alpha \in \mathbb{R}_{>0}$ is a regularization parameter. The loss function $\mathcal{L}_\theta$ is composed by the sum of n terms defined as:

$$\mathcal{L}_\theta'^k = \|G^\perp \{ [\nabla_q (M_{(\cdot, k)}^{-1}(q))]^\top - M_d^{\theta_R}(q_{\bar{r}}) M^{-1}(q) [\nabla_q (M_{d(\cdot, k)}^{\theta_R^{-1}}(q_{\bar{r}}))]^\top + U_k^{\theta_U^k}(q_{\bar{r}}) M_d^{\theta_R^{-1}}(q_{\bar{r}})] \|^2, \quad (6.31)$$

and a term aimed at ensuring the positivity of the Hessian of the heterogeneous solution:

$$\mathcal{L}_\theta'' = \max(0, -\nabla_{q_{\bar{r}}}^2 \tilde{V}_d^{\theta_R}(q_{\bar{r}}^*) + \epsilon_L), \quad (6.32)$$

where $\epsilon_L > 0$, is a small positive constant ensuring the Hessian to be strictly positive, with θ_R denoting dependence of the latter on $M_d^{\theta_R}$. It is emphasized that, depending on the open-loop characteristics of the considered system, the dimensionality of the NNs' input and, consequently, the sample space is significantly reduced, enhancing the network's accuracy. Once the desired inertia $M_d^{\theta_R}(q_{\bar{r}})$ has been learned, the ES stage of IDA-PBC problem is concluded if the following linear

optimisation problem in the unknown $\bar{V}_d(q)$ is solved:

$$\begin{aligned} G^\perp M_d^{\theta_R}(q_r) M^{-1}(q) \nabla_q \bar{V}_d(q) &= 0, \\ \text{subject to } q^* &= \operatorname{argmin}_q \bar{V}_d(q). \end{aligned} \quad (6.33)$$

In what follows it is shown how the search for a suitable structure for $\bar{V}_d(q)$ that solves the (6.33) can be avoided if the Kinetic PDE is solved up to a small error $\epsilon^\perp \approx 0$. Let us define the simplest expression for $\bar{V}_d(q)$ such that it has an isolated and globally defined minimum at q^* :

$$\bar{V}_d(q) = \frac{1}{2}(q - q^*)^\top P(q - q^*), \quad (6.34)$$

with P a constant diagonal matrix such that $P \in \mathbb{R}_{>0}^{n \times n}$, with arbitrary positive entries $p_{ij}, \forall i = j = 1, \dots, n$. Let us define the full-rank matrix:

$$\bar{G} = \begin{bmatrix} G^\perp \\ G^\top \end{bmatrix} \in \mathbb{R}^{n \times n}. \quad (6.35)$$

The (6.1) can be equivalently rewritten, premultiplying by $\bar{G}$, as:

$$\begin{bmatrix} \bar{G}\dot{q} \\ \bar{G}\dot{p} \end{bmatrix} = \begin{bmatrix} 0 & \bar{G} \\ -\bar{G} & 0 \end{bmatrix} \begin{bmatrix} \nabla_q H(q, p) \\ \nabla_p H(q, p) \end{bmatrix} + \begin{bmatrix} 0 \\ \bar{G}G \end{bmatrix} u. \quad (6.36)$$

The Matching Conditions are now defined as:

$$\begin{bmatrix} 0 & \bar{G} \\ -\bar{G} & 0 \end{bmatrix} \begin{bmatrix} \nabla_q H(q, p) \\ \nabla_p H(q, p) \end{bmatrix} + \begin{bmatrix} 0 \\ \bar{G}G \end{bmatrix} u_{ES} = \begin{bmatrix} 0 & \bar{G}M^{-1}M_d \\ -\bar{G}M_dM^{-1} & \bar{G}J_2 \end{bmatrix} \begin{bmatrix} \nabla_q H_d(q, p) \\ \nabla_p H_d(q, p) \end{bmatrix}. \quad (6.37)$$

Since the q dynamics is trivially satisfied, it is obtained that:

$$\bar{G}Gu_{ES} = \bar{G}A(q, p) = \begin{bmatrix} G^\perp \\ G^\top \end{bmatrix} A(q, p) = \begin{bmatrix} a^\perp \\ a^\top \end{bmatrix}, \quad (6.38)$$

where $A(q, p)$ is defined as the term inside the brackets of the (6.7), so that $a^\perp$ is exactly the PDE, while $a^\top$ defines the terms that can be assigned to the CL system, living in the space of the input matrix G . By simple calculations, recalling that $G^\perp G = 0$, it is shown that:

$$\bar{G}Gu_{ES} = \begin{bmatrix} 0 \\ G^\top G \end{bmatrix} u_{ES} = \begin{bmatrix} 0 \\ G^\top A(q, p) \end{bmatrix} = \begin{bmatrix} 0 \\ a^\top \end{bmatrix}. \quad (6.39)$$

Let us assume that the PDE is not solved to zero, but has error $\epsilon^\perp = \epsilon_M^\perp + \epsilon_V^\perp = \|a^\perp\| = \|a_M^\perp\| + \|a_V^\perp\|$, sum of the contribution of the kinetic and the potential PDE errors. Therefore, it is shown that:

$$\bar{G}Gu_{ES} + \begin{bmatrix} \epsilon^\perp \\ 0 \end{bmatrix} = \bar{G}A(q, p) \Rightarrow Gu_{ES} = A(q, p) - \bar{G}^{-1} \begin{bmatrix} \epsilon^\perp \\ 0 \end{bmatrix}. \quad (6.40)$$

By substituting the ES control expression in the (6.36) and by defining u_{DI} as in (6.14), the CL in presence of an error on the PDE has the form:

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & M^{-1}M_d \\ -M_dM^{-1} & J_2 - GK_vG^\top \end{bmatrix} \begin{bmatrix} \nabla_q H_d(q, p) \\ \nabla_p H_d(q, p) \end{bmatrix} + \begin{bmatrix} 0 \\ -\tilde{a}^\perp \end{bmatrix} \quad (6.41)$$

where $\tilde{a}^\perp = \bar{G}^{-1} \begin{bmatrix} \epsilon^\perp \\ 0 \end{bmatrix} \in \mathbb{R}^{n \times 1}$. By computing the derivative of $H_d(q, p)$ over the CL trajectories, it is obtained that:

$$\dot{H}_d(q, p) = -\nabla_p^\top H_d(q, p)GK_vG^\top \nabla_p H_d(q, p) - \nabla_p^\top H_d(q, p)\tilde{a}^\perp. \quad (6.42)$$

Being the gain K_v a symmetric and positive definite matrix, it is possible to bound the (6.42) as:

$$\dot{H}_d(q, p) \leq -\lambda_{\min}^K \|G^\top \nabla_p H_d(q, p)\|^2 + \|\nabla_p^\top H_d(q, p)\tilde{a}^\perp\|, \quad (6.43)$$

with $\lambda_{\min}^K = \lambda_{\min}(K_v)$. Let us consider the (6.41) and assume that $\epsilon^\perp = \epsilon_M^\perp$, denoting that the error on the total PDE is reduced to the error on the Kinetic PDE. Furthermore, if the Kinetic PDE has been solved, $\epsilon^\perp \approx 0$. With this assumption, by Lemma (6) (see the Appendix), it is guaranteed that the momenta dynamics in (6.41), starting from p_0 , are such that $\|p(t)\| \leq \epsilon$, $\forall t \geq 0$, being $\tilde{a}_M^\perp(q, p)$ bounded on $\mathcal{X}$ due to the PINN approximation of $M_d^{\theta_R}$ and $J_2^{\theta_V}$.

By explicitly writing the gradient of the (6.34):

$$\nabla_q \bar{V}_d(q) = P(q - q^*), \quad (6.44)$$

the (6.33) is always satisfied when $q = q^*$. Furthermore, by the (6.43), if $\lambda_{\min}^K$ is properly tuned to ensure that $\dot{H}_d(q, p)$ remains negative semi-definite $\forall t \geq 0$ and the (6.34) is set through an arbitrary P , the desired Hamiltonian decreases up to a small value around 0, implying $q \rightarrow q^*$, $\epsilon_V^\perp \rightarrow 0$, concluding that the stability analysis for small perturbations can be extended also to the q dynamics when trajectories are sufficiently near q^* . In that case, locally, it is possible to treat $\tilde{a}^\perp$ as a small and bounded perturbation, despite the potential PDE has not been solved. This allows to conclude that by finding a gain matrix K_v^* , the trajectories of the (6.41) starting from (q_0, p_0) , will approach a region $B_{\epsilon_{\bar{b}}}$ of radius $\epsilon_{\bar{b}}$ around $(q^*, 0)$ in which:

$$\|(q, p)\| \leq \epsilon_{\bar{b}}, \forall t \geq 0. \quad (6.45)$$

In the next section, the proposed methodology is applied through numerical simulations to a simple example taken from [29].

6.4 Simulations

Let us consider the Vertical Take-Off and Landing (VTOL) moving on the (x, y) plane, described by the following set of equations:

$$\begin{cases} \ddot{x} = -\sin\theta u_1 + \cos\theta u_2 \\ \ddot{y} = \cos\theta u_1 + \sin\theta u_2 - g \\ \ddot{\theta} = u_2, \end{cases} \quad (6.46)$$

where θ is the roll angle, g is the gravity acceleration and u_1, u_2 are the control actions. The system (6.46) admits the following PH representation:

$$\begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \begin{bmatrix} \nabla_q H(q, p) \\ \nabla_p H(q, p) \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} u, \quad (6.47)$$

with $q = \begin{bmatrix} x & y & \theta \end{bmatrix}^\top \in \mathbb{R}^3$, $p = \dot{q} \in \mathbb{R}^3$, being $M = I_3$ the tridimensional identity matrix, and $u = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^\top \in \mathbb{R}^2$. The Hamiltonian function is given by the sum of the kinetic and potential energy as:

$$H(q, p) = \frac{1}{2} p^\top p + g q_2, \quad (6.48)$$

while the input matrix is:

$$G(q_3) = \begin{bmatrix} -\sin q_3 & \cos q_3 \\ \cos q_3 & \sin q_3 \\ 0 & 1 \end{bmatrix}. \quad (6.49)$$

The system has underactuation degree $s = 1$ and the left annihilator of G is obtained as:

$$G^\perp = \begin{bmatrix} \cos q_3 & \sin q_3 & -1 \end{bmatrix}. \quad (6.50)$$

By the definition of (6.18), the admissible equilibria are obtained by computing:

$$s(q)|_{q=q^*} = s(q_3)|_{q_3=q_3^*} = g \sin(q_3)|_{q_3=q_3^*} = 0, \quad (6.51)$$

and, according to this, it is proposed to stabilize the roll angle to $\theta^* = 0$ [rad].

Due to the constant M and having $G^\perp(q_3)$, it is possible to estimate $M_d^{\theta_R-1}(q_3)$, implying that $\gamma(q_3)$. Therefore, the heterogeneous solution of the potential PDE is obtained as:

$$\nabla_{q_3} \tilde{V}_d(q_3) = \frac{s(q_3)}{\gamma_3(q_3)} \Rightarrow \tilde{V}_d(q_3) = \int_0^{q_3} \frac{s(\tau)}{\gamma_3(\tau)} d\tau. \quad (6.52)$$

By simple computations, the constraint of having a positive definite Hessian at q_3^* reduces to:

$$\begin{aligned} \gamma_3(q_3^*) \frac{\partial s}{\partial q_3}(q_3^*) &= g\gamma_3(q_3^*) > 0 \Rightarrow \\ \Rightarrow \gamma_3(q_3^*) &= M_{d(1,3)}(q_3^*) - M_{d(3,3)}(q_3^*) > 0. \end{aligned} \quad (6.53)$$

The PINNs have been trained against the loss function (6.30), according to (6.31), (6.32), where:

$$\mathcal{L}_\theta'' = \max(0, M_{d(3,3)}^{\theta_R}(q_3) - M_{d(1,3)}^{\theta_R}(q_3) + \epsilon_L). \quad (6.54)$$

Table (6.1) reports the architecture and the hyperparameters of R^{θ_R} and $\mathcal{U}^{\theta_U}$. The dataset has been generated by uniformly sampling the q_3 state space $\mathcal{X} = [-\pi/2, \pi/2]$ and adding to each sample a uniform noise $\mathcal{N} \sim \text{Uniform}(-0.05, 0.05)$ to enhance the generalization capabilities of the networks. Training parameters included $n_{\text{epochs}} = 100$, batch size = 50, a dataset of dimension $d_{\text{dataset}} = 200$, $\epsilon_{M_d} = 0.1$, $\alpha = 0.1$ and $\epsilon_L = 1 \times 10^{-6}$.

Fig. (6.1) depicts the behavior of the loss function during training and gradients of R^{θ_R} have been obtained by the Pytorch AD [33].

Simulation has been runned over a time interval of 10[s] and dynamic equations have been solved using the fourth-order Runge-Kutta method with time step $dt = 0.001$. The symmetric and positive definite matrix P has been fixed as:

$$P = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 15 \end{bmatrix}, \quad (6.55)$$

while the gain matrix K_v^* has been tuned as:

$$K_v^* = \begin{bmatrix} k_{v11}^* & 0 \\ 0 & k_{v22}^* \end{bmatrix}, \quad (6.56)$$

where $k_{v11}^* = k_{v22}^* = \lambda_{\min}^{K^*} = 18.75$. The initial state has been set to $q_0 = \begin{bmatrix} 0 & 2 & 0.5 \end{bmatrix}^\top$ and $p_0 = 0_3$, while the desired equilibrium at the origin. Fig. (6.2a) reports the q dynamics, showing the trajectories' convergence at $t \approx 6$ [s], while Fig. (6.2b) depicts the momenta. Fig. (6.2c) illustrates the energetic behaviour of the CL system in a first interval $t \in [0, 5]$, in which (in blue) the desired energy $H_d^{\theta_R}(q, p)$ is dissipated, as expected from passive systems, while (in red) the time derivative of the desired Hamiltonian is shown to be negative semi-definite. Fig. (6.2d) focuses on the desired energy at steady-state, highlighting the Lyapunov stability result. Fig. (6.2e) shows the control resulting from the IDA-PBC law in which, at the equilibrium, u_1 correctly balances gravity while u_2 keeps the angle $\theta = 0$. The PDEs (6.9), (6.10) are reported in Fig. (6.2f), (6.2g) where, respectively, the evolution over a first interval $t \in [0, 5]$ and a second interval $t \in [5, 10]$ are shown. Results demonstrate the effectiveness of the proposed solution with the potential PDE having an initial error $\epsilon_V^\perp \approx -28$ and converging to $\|a_V^\perp\| \leq 0.05$. This value, for the previous discussion, identifies the

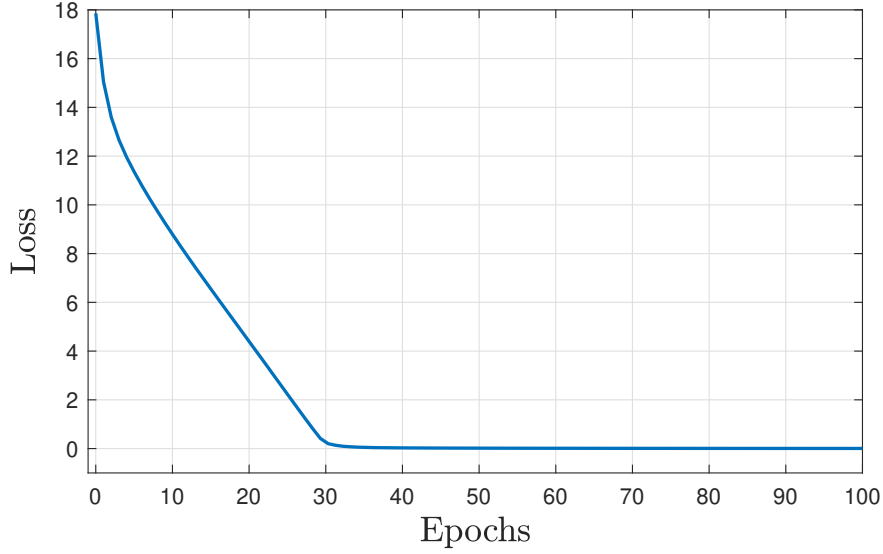


Figure 6.1: Loss evolution over 100 epochs.

steady-state trajectories error in presence of a small bounded error $\|a_M^\perp\| = \epsilon_M^\perp \approx 2 \times 10^{-10}$ on $\mathcal{X}$. In Fig. (6.2h), a different configuration with a new desired equilibrium $\bar{q}^* = \begin{bmatrix} -0.5 & 1 & 0 \end{bmatrix}^\top$ has been reported. Trajectories converge in about 4[s] with same initial conditions, same matrix P and $k_{v11}^* = k_{v22}^* = 12.1$.

It is stressed how the presented methodology allows to stabilize the system to a new desired equilibrium point over $\mathcal{X}$ by simply tuning a different K_v , keeping the same potential energy function and same kinetic PDE solution.

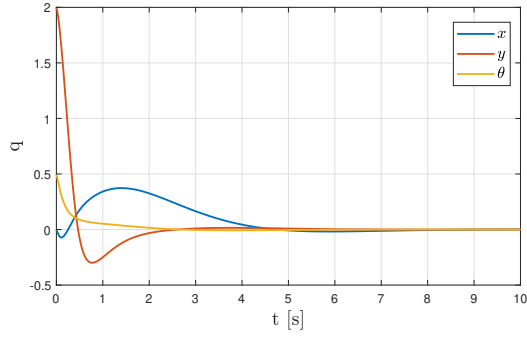
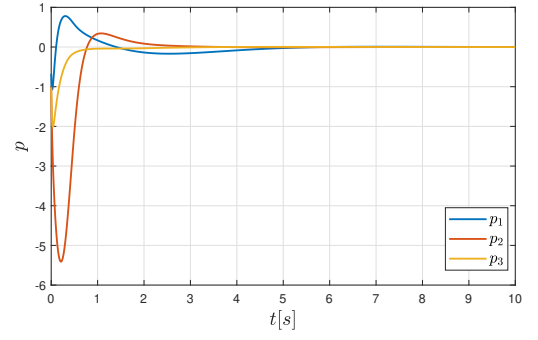
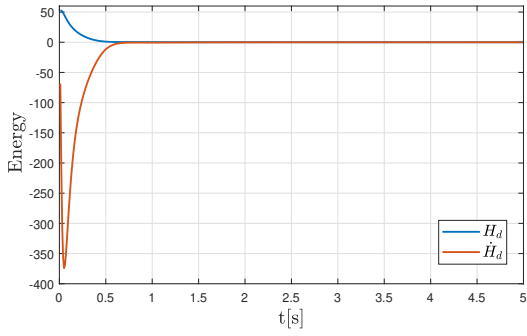
Hyperparameter	R^{θ_R}	$\mathcal{U}^{\theta_U}$
Learning Rate	0.0001	0.0001
Optimizer	Adam	Adam
Weights Initialization	Xavier	Xavier
Layers dimensions	[1, 16, 16, 16, 6]	[1, 16, 16, 16, 9]
Hidden Layers Act. Functions	ELU	ELU
Output Layer Act. Functions	Softplus	Linear

Table 6.1: NNs hyperparameters used in the training process.

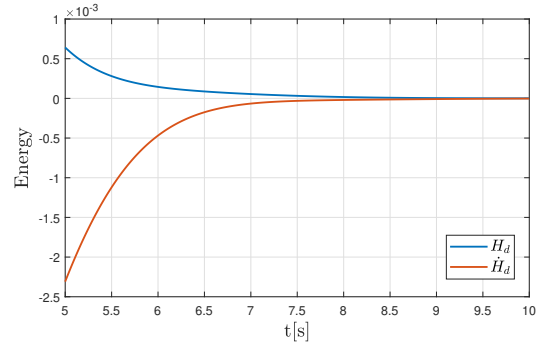
6.5 Conclusions

Within the context of IDA-PBC, this chapter proposes a PINNs-based approach to stabilize a generic nonlinear underactuated system expressed in PH form to a desired equilibrium. This study integrates PINNs for solving the kinetic PDE, employing a customized loss function to a stability analysis, avoiding to solve the potential PDE. This has the advantage of avoiding the need to repeatedly solve the latter each time a new desired equilibrium has to be stabilized. It is verified

that, via a suitable choice of the damping matrix and in presence of a small neural error in solving the kinetic PDE, the system can be stabilized arbitrarily closed to any desired equilibrium within the sampling space. The proposed approach is then validated on a nonlinear underactuated VTOL mechanism system, showing how, once having fixed a simple structure for the desired potential energy, the tuning of the damping matrix allows to stabilize the system by dissipating energy. Future research work is aimed at enhancing the robustness of such methodology potentially employing NNs for the optimal estimation of the damping matrix, instead of resorting to empirical tuning, along with a more detailed investigation of the role of P in the system's response performance.

(a) Positions over $t \in [0, 10]$.(b) Momenta over $t \in [0, 10]$.

(c) Energy in first 5 s.



(d) Energy in last 5 s.

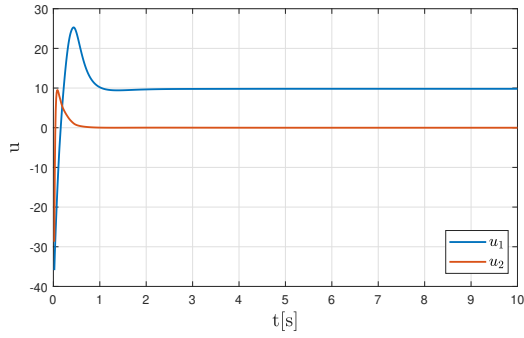
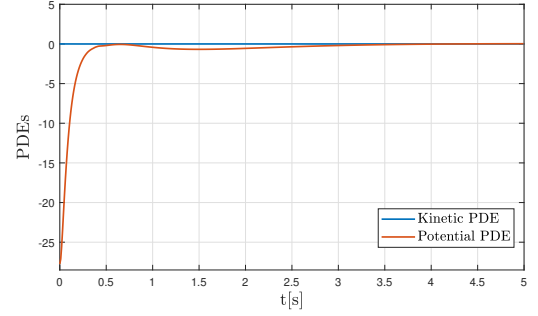
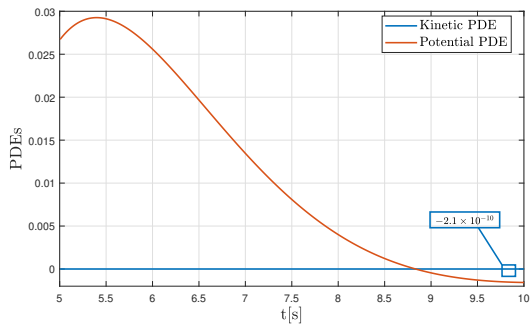
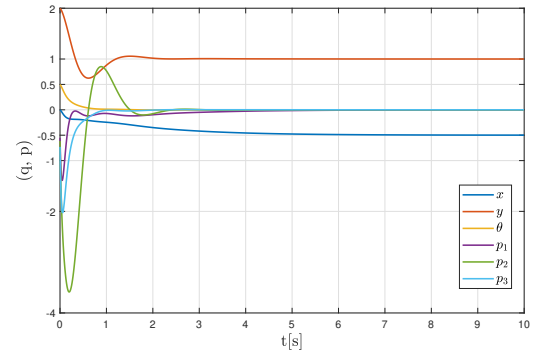
(e) Control over $t \in [0, 10]$.(f) PDEs over $t \in [0, 5]$.(g) PDEs over $t \in [5, 10]$.(h) Convergence to $(\bar{q}^*, 0)$.

Figure 6.2: Simulation results of the learned IDA-PBC framework: system trajectories, energetic analysis, and PDE residuals over time.

Chapter 7

A Machine Learning Approach to Passivity-Preserving Safety-Critical Control

Passivity-Based Control enables to constructively synthesize controllers with guarantees on the stability of the overall system. Although safety-critical control based on Control Barrier Functions is able to synthesize controllers that effectively cope with state constraints, passivity and input bounds are not addressed in general. This chapter 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.

7.1 Introduction

This chapter 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[62]. PBC has several practical advantages:

- Passive systems tolerate modeling errors, uncertainties, and even time delays better than non-passive ones [63].
- PBC methods such as PBC-IDA [64] 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 [65] or spacecraft attitude control in presence of uncertainties [66]. 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 chapter is organized as follows: Section 7.2 presents related works in the context of PBC and CBF, and reports the proposed innovations; Section 7.3 provides background on PBC and CBFs; Section 7.4 presents the formulation for the ML-based optimization aimed at finding the desired CBF; Section 7.5 validates the proposed approach by numerical simulations; Section 7.6 draws the conclusions and outlines future research directions.

7.2 Related works and proposed innovations

The concept of passivity-preserving safety-critical control, specifically tailored to mechanical systems, was introduced by [47], which puts in light the passivity-preserving behavior of the CBFs proposed in [38]. The latter are not derived based on an optimality criterion nor address input constraints. Conversely, inspired by works such as [51], where optimal CBFs are found through ML-based optimization, this chapter 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 [67], 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 [68]. 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 [69, 70]: 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 [71, 72]. 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 [73], voltage-source converters [74], and port-Hamiltonian systems [75] 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 [76] 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 minimum energy threshold; a safety filtering control action guarantees that the tank never depletes, hence preserving passivity. In [77], 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 [76] and [77] 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 [78]. The work in [79] provides a data-driven approach to the synthesis of provably safe controllers and Lyapunov functions. In [51], an actor-critic method is defined to synthesize CBFs under input saturation. In [80], 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 [79], (ii) guaranteeing safety under input saturation [80], or (iii) synthesizing Robust Control Lyapunov Barrier Functions without input bounds [78].

To the best of the authors' knowledge, while existing works have separately addressed input saturation (e.g., [51, 80]) and safety constraints (e.g., [78, 81]), this chapter 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., [76, 77]), 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.

7.3 Preliminaries

Consider a nonlinear input-affine system of the form:

$$\dot{x} = F(x, u) = f(x) + g(x)u, \quad (7.1)$$

with relative degree r , where $x \in X \subseteq \mathbb{R}^n$, $u \in U \subseteq \mathbb{R}^m$, with the functions $F : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^n$, $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $g : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ continuously differentiable. In addition, f and the columns of g are smooth vector fields (i.e., C^∞). The system (7.1) is equipped with an output function $c : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^m$ such that $y = c(x, u)$ is the measured system output, with the same dimension of the input, and c is a smooth mapping.

7.3.1 Passivity and Passivity-Based Control

Definition 1 (Passivity [82]): *the system 7.1 is passive if there exists a positive definite, continuously differentiable storage function $V : \mathbb{R}^n \rightarrow \mathbb{R}$ such that, for all x and u ,*

$$\dot{V} = \frac{\partial V}{\partial x} F(x, u) \leq u^T y. \quad (7.2)$$

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

Definition 2 (Passivity-Based Control (PBC)): *the PBC objective for system (7.1) is to find a state feedback law:*

$$u(x) = u_{des}(x) + \nu, \quad (7.3)$$

where ν 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 (ν, y) , such that the closed-loop system is in the form:

$$\dot{x} = f_{cl}(x) + g(x)\nu, \quad (7.4a)$$

$$y = g(x)^T \frac{\partial V_{cl}(x)}{\partial x}, \quad (7.4b)$$

where $u_{des}(x)$ is a generic passive state-feedback control law and a typical ν takes the form $\nu = -Ky$, with K positive definite [62]. 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.

7.3.2 Passivity Preserving Control Barrier Functions

Considering system (7.1), 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:

$$x(t) \in S \quad \forall t > 0, \forall x(0) \in S. \quad (7.5)$$

The safe set S is defined as the superlevel set of a continuously differentiable function $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$, 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 $\sup_{u \in U} [L_f \phi(x) + L_g \phi(x)u] \geq 0$, for all $x \in U$.

Theorem 1 [81]: *let $\phi(x) \in C^1$ be a CBF on X for the system (7.1) 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:

$$\begin{aligned} u^*(x) &= \arg \min_{u \in U} \|u(x) - u_{des}(x)\|^2, \\ \text{s.t.} \quad &L_f \phi(x) + L_g \phi(x)u(x) \geq 0, \end{aligned} \quad (7.6)$$

where $u_{des}(x)$ is the first term of a PBC controller (7.3) satisfying Def. 2. QP 7.6 admits the closed form solution of the form:

$$u^*(x) = u_{des}(x) + u_{sc}(x), \quad (7.7)$$

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

$$u_{sc}(x) = \begin{cases} -\frac{L_g\phi(x)^T}{\|L_g\phi(x)\|^2}\psi(x, u_{des}) & \text{if } \psi(x, u_{des}) < 0 \\ 0 & \text{if } \psi(x, u_{des}) \geq 0 \end{cases}, \quad (7.8)$$

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

Theorem 2 [47]: *let $u(x) = u^*(x) + \nu$, where $u^*(x)$ is the safety-critical controller (7.7) induced by a CBF $\phi(x)$. The resultant closed-loop system (7.4) is passive with respect to $V_{cl}(x)$ and the input-output pair (ν, y) if and only if $L_g\phi(x) \neq 0$ and the following inequality holds when $\psi(x; u_{des}) < 0$:*

$$-\frac{L_gV_{cl}(x)L_g\phi(x)^T}{\|L_g\phi(x)\|^2}\psi(x, u_{des}(x)) \leq d_p(x), \quad (7.9)$$

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).

7.4 ML-based Approach for Passivity-Preserving Safety-critical Control

In this Section, the closed loop system (7.4) is considered with the control law (7.3). The term ν of (7.3) 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.

7.4.1 Neural SC-PBC

The methodology presented in this work is inspired by [51], 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:

$$\rho(x) = \left[\prod_{i=1}^{r-1} \left(1 + \gamma_i \frac{d}{dt} \right) \right] h(x), \quad (7.10)$$

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 [18]. In [51], the input constraints are tackled by subtracting a semi-positive definite term to the CBF (7.10), obtaining the modified neural CBF (NCBF):

$$\phi_\theta(x) = \rho(x) - K_\theta(x), \quad (7.11)$$

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 [18], 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 [32], 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 equations (7.7), (7.8):

Definition 3: *the neural safety-critical control $u_\theta(x)$ is defined as:*

$$u_\theta(x) = u_{des}(x) + u_{sc,\theta}(x), \quad (7.12)$$

with:

$$u_{sc,\theta}(x) = \begin{cases} -\frac{L_g \phi_\theta(x)^T}{\|L_g \phi_\theta(x)\|^2} \psi_\theta(x, u_{des}) & \text{if } \psi_\theta(x, u_{des}) < 0 \\ 0 & \text{if } \psi_\theta(x, u_{des}) \geq 0 \end{cases}, \quad (7.13)$$

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 (7.12)-(7.13) satisfies both the passivity condition (7.9) 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:*

$$\min_{\theta} \quad \mathcal{L}_\theta \quad (7.14a)$$

$$s.t. \quad A_\theta(x) \psi_\theta(x, u_{des}(x)) \leq d_p(x), \quad (7.14b)$$

$$u_\theta(x) \in U, \quad \forall x \in \partial S_{\phi_\theta}, \quad (7.14c)$$

where (7.14b) enforces the passivity condition (7.9) with:

$$A_\theta(x) = -\frac{L_g V_{cl}(x) L_g \phi_\theta(x)^T}{\|L_g \phi_\theta(x)\|^2}. \quad (7.15)$$

If the OP (7.14) 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. (7.12) 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 (7.12) does not exist.

Figure 7.1 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, $X_\theta^P := \{x \in X \mid (7.14b)\}$, which is the region where safety critical control can be enforced passively, and $X_\theta^U := \{x \in X \mid (7.14c)\}$, where the input bounds constraints hold.

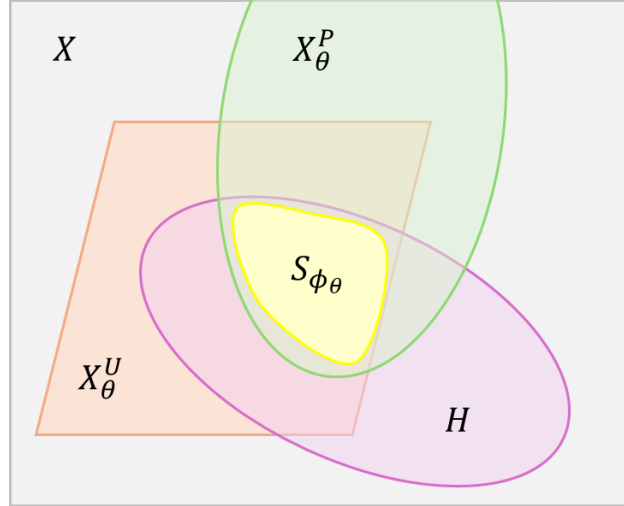


Figure 7.1: Graphical depiction of problem (7.14)

7.4.2 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 (7.14) 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 (7.14) 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 (7.14a), 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 [51] as a *regularization term* and takes the form:

$$\mathcal{L}_\theta^r = \int_X (1 - \sigma(\phi_\theta(x))) dx, \quad (7.16)$$

where $\sigma(\cdot) : \mathbb{R} \rightarrow (0, 1)$ 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 $x \in S_{\phi_\theta}^\circ$, and $\sigma(\phi_\theta(x)) < 0.5$ if $x \notin S_{\phi_\theta}$. Since $\sigma(\cdot)$ is monotonically increasing, minimizing (7.16) 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 (7.14) are cast as two additional cost functions. The passivity constraint (7.14b) can be transformed into a cost index which becomes positive when (7.14b) is violated, i.e.:

$$\mathcal{L}_\theta^p(x) = \max\{0, A_\theta(x)\psi_\theta(x, u_{des}(x)) - d_p(x)\}, \quad (7.17)$$

(similarly to the Lyapunov-based loss function in [79]). The input constraint (7.14c) can be cast

into the following cost index:

$$\mathcal{L}_\theta^u(x) = \inf_{u \in U} [d(u_\theta(x), u)], \quad (7.18)$$

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:

$$\min_{\theta} \left\{ \mathcal{L}_\theta^r + \max_{x \in S_{\phi_\theta}} [\lambda^p \mathcal{L}_\theta^p(x) + \lambda^u \mathcal{L}_\theta^u(x)] \right\}, \quad (7.19)$$

where λ^p and λ^u are weights, with sufficiently large values to let constraints (7.14b)-(7.14c) be verified: in this case, the max operator in (7.19) is such that the minimization seeks to drive the second and third terms to 0 for all $x \in S_{\phi_\theta}$.

7.4.3 Data Sampling, Critic Function and Loss Function

The first term of the cost function (7.19) 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 [51], ∂S_{ϕ_θ} is approximated by means of Monte Carlo Sampling (MCS). The term in (7.16) 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 (7.19). 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 (7.19)) 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:

$$\psi_\theta(x, u_{des}(x)) \leq 0. \quad (7.20)$$

2) Considering that the function $1 - \sigma(\phi_\theta(x))$ takes values mostly close to 0 in $S_{\phi_\theta}^o$, 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 (7.19) is negligible. Therefore, the region where MCS is performed is chosen close to the boundary ∂S_{ϕ_θ} , in the set:

$$S_{\phi_\theta}^\epsilon = \{x \in X : 0 \leq \phi_\theta(x) \leq \epsilon\}, \quad (7.21)$$

where ϵ is a positive parameter so that $S_{\phi_\theta}^\epsilon \supset \partial S_{\phi_\theta}$. Moreover, since the probability of sampling data points exactly on the boundary ∂S_{ϕ_θ} is 0, the same points in $S_{\phi_\theta}^\epsilon$ will be used for the evaluation of the last two terms of (7.19).

Thus, the set where the samples are collected is

$$\tilde{S}_{\phi_\theta}^\epsilon = \{x \in S_{\phi_\theta}^\epsilon | h(x) \geq 0, \psi_\theta(x, u_{des}(x)) \leq 0\}. \quad (7.22)$$

MCS-based optimization for high dimensional control problems would require a method to uniformly sample on $\tilde{S}_{\phi_\theta}^\epsilon$; however, the use of such techniques goes beyond the scope of this work, and

the simpler sampling strategy used in [51] is adopted.

Let N be the number of points in $\tilde{S}_{\phi_\theta}^\varepsilon$ 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 (7.19), whereas the evaluation of the the other two terms is performed over a subset of k points in D_θ . Let the so-called *critic function* be defined as:

$$C_\theta(x) = \lambda^p \mathcal{L}_\theta^p(x) + \lambda^u \mathcal{L}_\theta^u(x). \quad (7.23)$$

The selected k points, collected in the set $\text{Top}_k(C_\theta, D_\theta)$, 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 (7.19) and on the described sampling strategy. In practice, problem (7.19) is solved through SGD on the following loss function:

$$L(\theta) = \frac{1}{|D_\theta|} \sum_{x \in D_\theta} R_\theta(x) + \sum_{x \in \text{Top}_k(C_\theta, D_\theta)} C_\theta(x), \quad (7.24)$$

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

7.4.4 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 $\tilde{S}_{\phi_\theta}^\varepsilon$, the number M and the threshold ξ used for the algorithm termination criterion must be selected.

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 (7.22)
 - 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: $D^{train} \leftarrow \{x \in D^{train} | x \in (7.22)\}$
 - 8: **end for**
 - 9: Find D_θ as a set of N samples in (7.22)
 - 10: Evaluate the loss function $L(\theta, D_\theta)$ (eq. (7.24))
 - 11: Exit if $L(\theta, D_\theta) < \xi$ for M iterations over i .
 - 12: **end for**
-

At each iteration i , the training set D^{train} is updated by sampling N states where equations (7.21)-(7.20) are met, and the parameters 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 equations (7.21)-(7.20) 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 equations (7.21)-(7.20) in the test set D_θ , and the set $\text{Top}_k(C_\theta, D_\theta)$ is found by evaluating the states in D_θ with the critic function (7.23). Finally, the obtained parametrization θ is tested over D_θ by means of the cost function $L(\theta)$ defined in (7.24): the algorithm terminates

if $L(\theta) < \xi$ for M iterations.

The set $\tilde{S}_{\phi_\theta}^\epsilon$ (7.22) (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 [83]. 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 (7.22), 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 (7.19) 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 [53], is to define the function $h(x)$ in (7.10) 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 $\mathcal{L}_\theta^r$, which drives $\phi_\theta(x)$ to expand S_{ϕ_θ} , (ii) the Lagrange multipliers λ^p, λ^u in (7.19), which force $\mathcal{L}_\theta^p$ and $\mathcal{L}_\theta^u$ to zero upon convergence, and (iii) the sampling strategy focusing on $\tilde{S}_{\phi_\theta}^\epsilon$, 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 7.5 show consistent convergence to feasible PP-CBFs that satisfy all constraints even for non-convex problems, with $\mathcal{L}_\theta^p$ and $\mathcal{L}_\theta^u$ decaying to near-zero.

7.5 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 (7.11), learned through algorithm 1 takes the form:

$$\phi_\theta(q, p, u(q, p)) = \dot{h}(q, p, u(q, p)) + \gamma h(q) - K_\theta(q, p), \quad (7.25)$$

where $h(q)$ describes the geometrical constraints for the system. For what concerns the data sampling presented in 7.4.4, 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 (7.24) are $\lambda^u = \lambda^p = 10^2$, and a weight of 10^{-2} is used for standard L2 regularization. The chosen optimizer is Adam [84]. The exit condition is specified by $M = 100$ and $\xi = 1e^{-2}$.

Four types of CBFs are compared: (i) the CBF defined by (7.10), called Limit Blind CBF (LB-CBF), which does not take into account input bounds and passivity; (ii) the NCBF (7.25), taking into account input bounds but not passivity, as in [51]; (iii) the proposed PP-NCBF, still given by (7.25) but taking into account both input bounds and passivity; (iv) the PP-CBF developed in

[47], 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 $K(q, p) : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^+$ 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 $\alpha \rightarrow \infty$. Moreover, as in [38], the forward invariance condition in (7.6) is relaxed into $L_f\phi(q, p) + L_g\phi(q, p)u_{\text{des}}(q, p) > -\gamma_K\phi$, where γ_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 [47]. Passivity inequality (7.9) is satisfied in PP-CBF by guaranteeing that the left side of the inequality (7.9) is negative independently of $d_p(x)$. Conversely, PP-NCBF allows the left hand side of (7.9) 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 $T_s = 5e^{-4}$.

7.5.1 Cart-pole system

The cart-pole system presented in [47] is considered, with state space $q = [x \ \beta]^T$, 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 [47], the cart-pole system and the nominal control do not contain dissipative terms. The passive control action is $u_{\text{des},1}(x) = -k(x - x_d)$, with $k = 1$, and $u_{\text{des},2}(\beta) = 0$. 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 $\gamma_K = 10$ 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 γ_K is selected to ensure the zeroing CBF condition $L_f\phi(q, p) + L_g\phi(q, p)u_{\text{des}}(q, p) > -\gamma_K\phi$ is satisfied without excessive conservatism. This aligns with the theoretical requirement that γ_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 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 $V_d(q) = \mathcal{H}(q, \dot{q}) + \frac{1}{2}k(x - x_d)^2$, 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. 7.2, are initialized from $(x, \beta, \dot{x}, \dot{\beta}) = (0, 0, 3.5, 0)$, i.e., with the

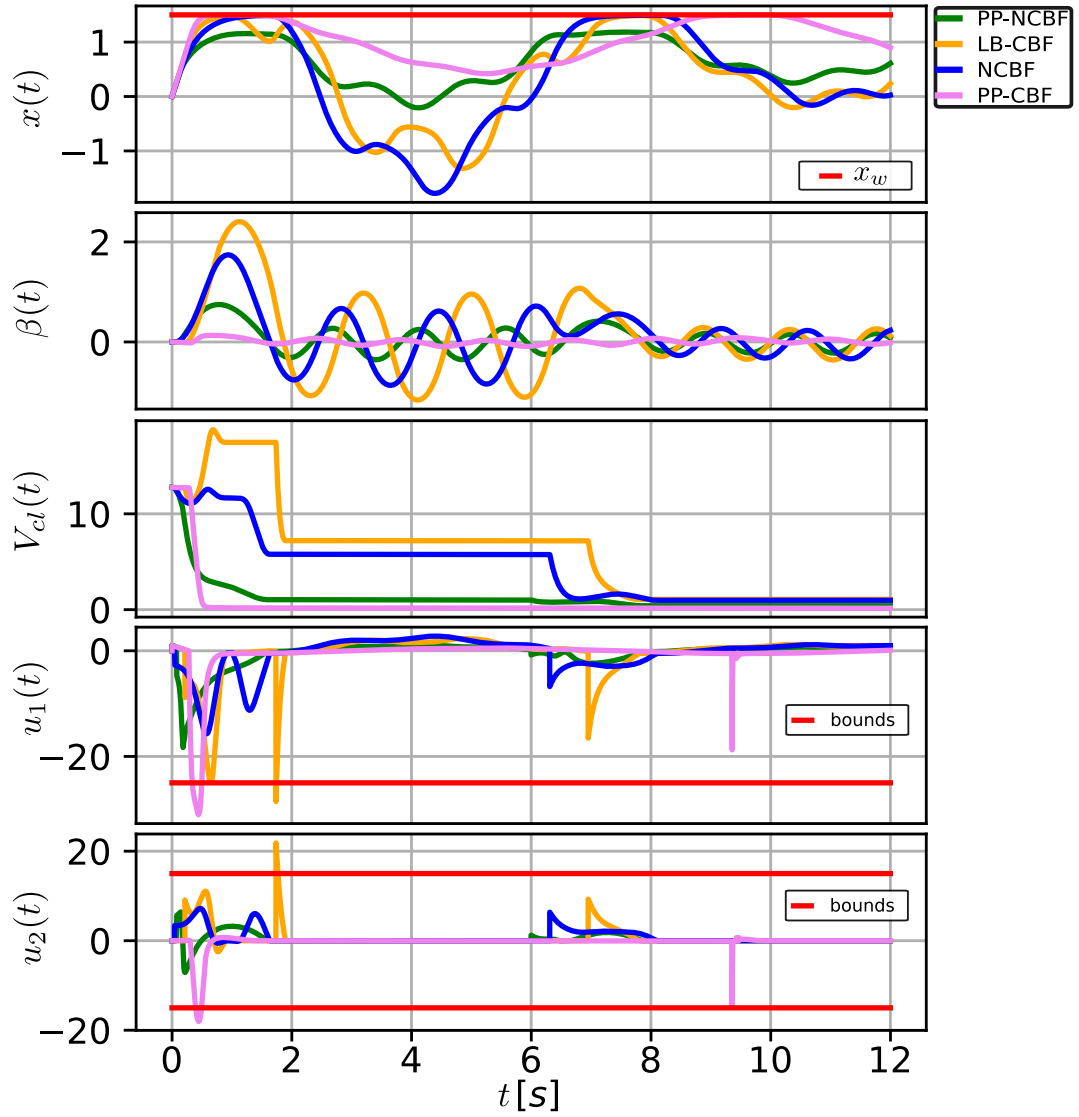


Figure 7.2: Cart-pole simulations: a) position $x(t)$; b) pendulum angle; c) storage function $V_{cl}(t)$; d) control signal $u_1(t)$; e) control signal $u_2(t)$.

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 ($V_d(t)$ 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 the safety-critical control is activated and shows the fastest energy dissipation rate ($V_d(t)$ 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 $S_{\phi_\theta}^\epsilon$ of the boundary ∂S_{ϕ_θ}) and keeps a safe distance from the wall.

7.5.2 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 $K_p = 1$, $K_d = 6$. 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 $r = 0.2\text{m}$; 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 $\gamma_\kappa = 1$ and $\alpha = 100$ for the PP-CBF. The desired storage function is $V_d(q) = K(q, \dot{q}) + \frac{1}{2}K_p\|q - q_d\|^2$, 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 [47], 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 (7.21).

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. 7.3. Fig. 7.4 shows the corresponding storage function plot, showing that PP-CBF and PP-NCBF preserve passivity ($V_d(t)$ 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. 7.3. 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 (7.24), whereas PP-CBF does not perform any optimization.

A visualization of the learned PP-NCBF is shown in Fig. 7.5, representing the safe set in the

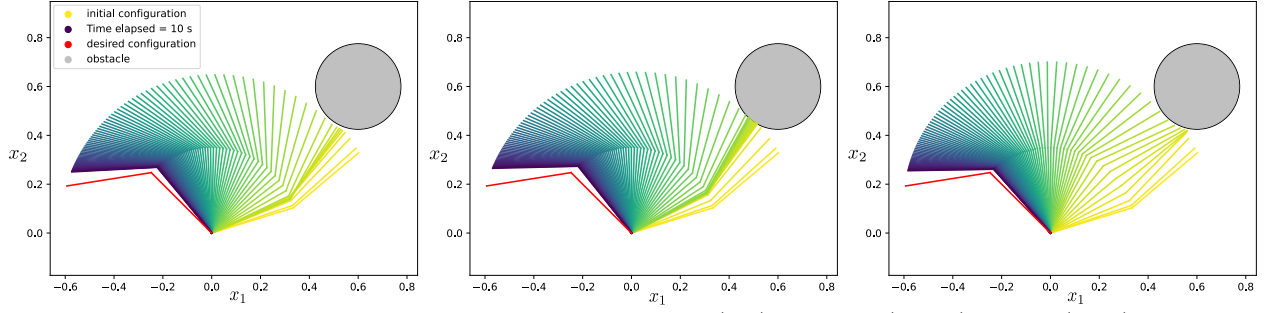


Figure 7.3: Simulations on the 2R robot: PP-NCBF (left); PP-CBF (centre); NCBF (right).

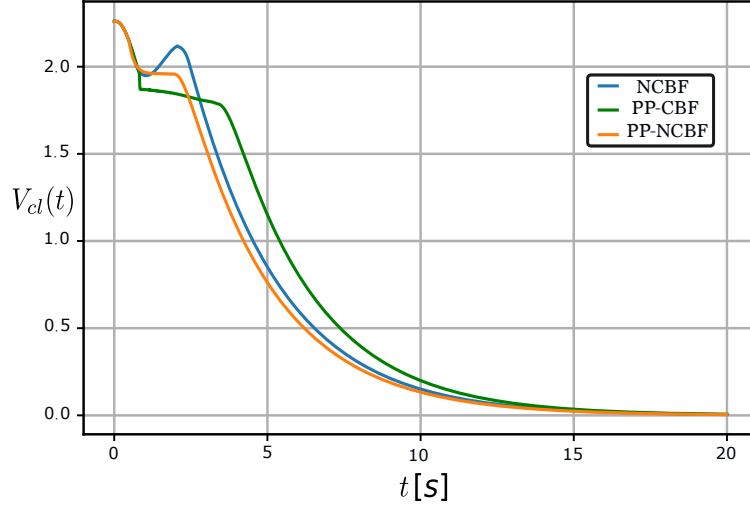


Figure 7.4: Simulations on the 2R robot: storage function $V_{cl}(t)$.

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.

7.5.3 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 (7.24) is independent on $|D_\theta|$ and depends only on the quantity k in the Top_k operation. The regularization term in (7.24) 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

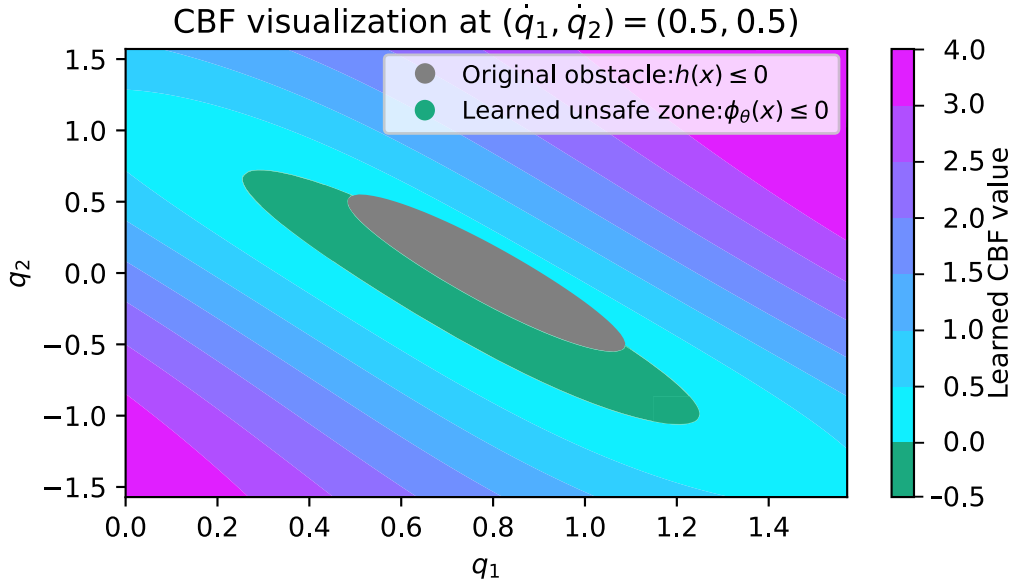


Figure 7.5: Comparison of original (grey) and learned (green) unsafe sets.

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 7.1, 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. 7.6 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.

Table 7.1: 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

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

7.5.4 Safe set comparison with Nonlinear MPC

Following the methodology proposed in [51], we perform a comparative analysis between the safe set derived from a learned Control Barrier Function (CBF) and that obtained using Nonlinear

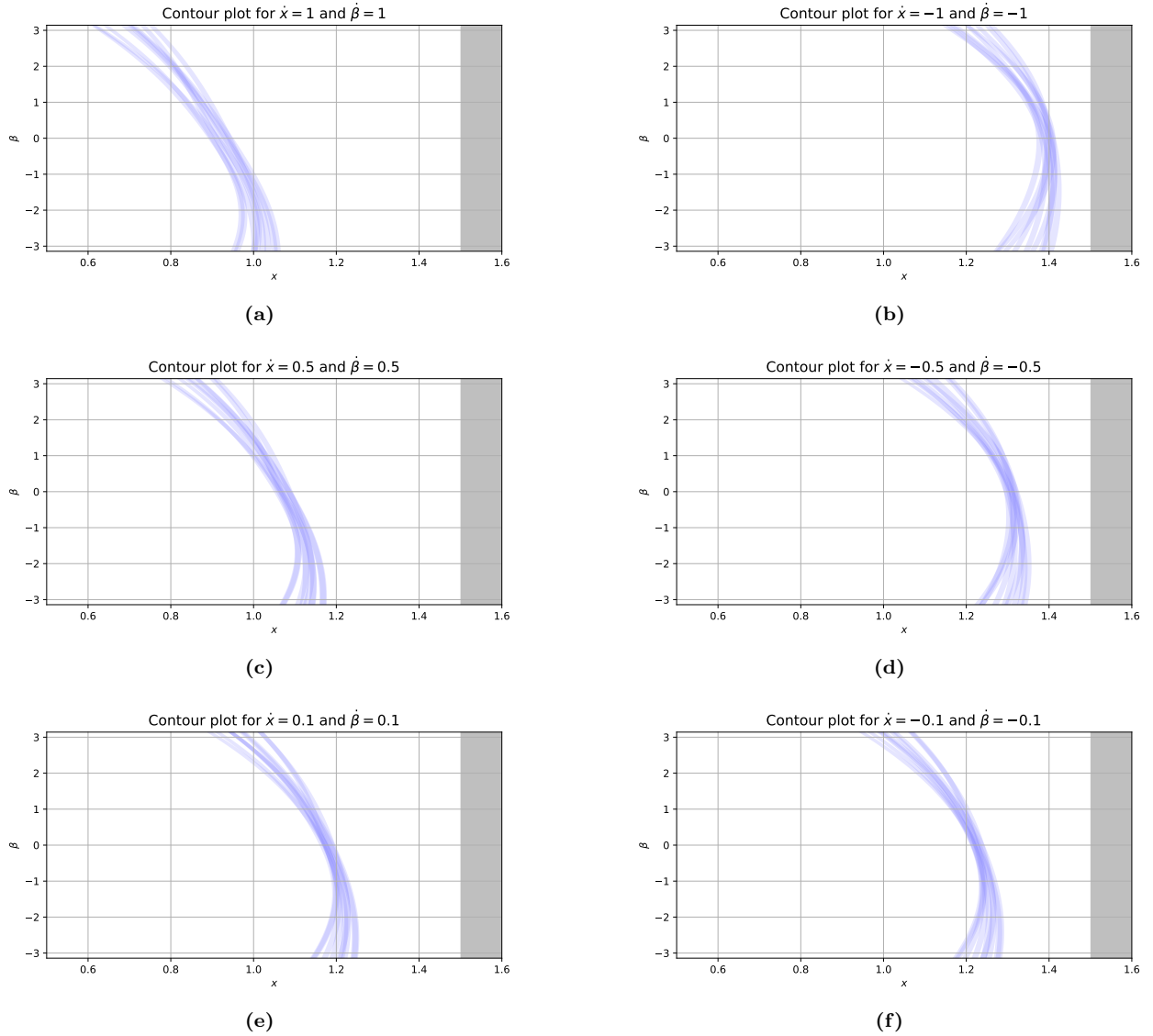


Figure 7.6: Contour plots for the CBF obtained after training on the cart-pole system. The plots are obtained fixing the values for $\dot{x}, \dot{\beta}$ as specified in the title of each figure and plotting in blue the region of the space (x, β) where the output of the CBF function has values in $[-0.05, 0.05]$. 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.

Model Predictive Control (NMPC)[85]. 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 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. 7.6.

To estimate the safe set under nonlinear model predictive control (NMPC), we evaluate control

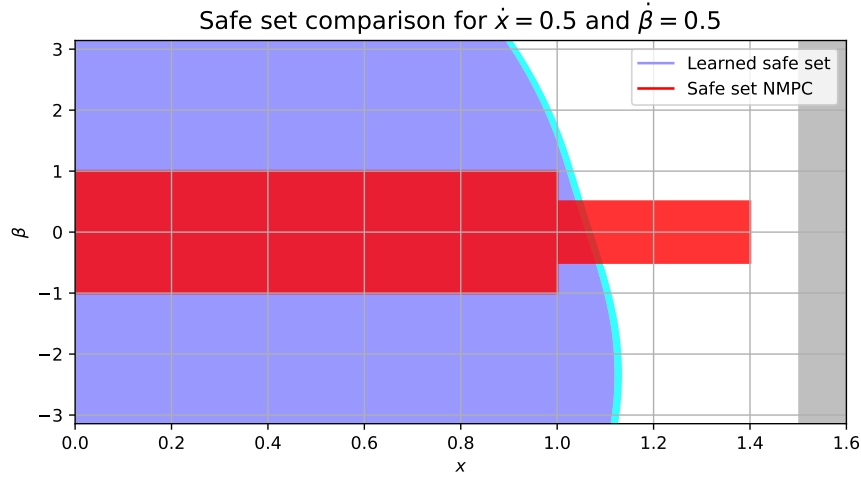


Figure 7.7: NMPC safe set obtained at $(\dot{x}, \dot{\beta}) = (0.5, 0.5)$

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. 7.7, the empirically derived safe set under NMPC represents approximately 60% of the safe set obtained from the CBF algorithm, as shown in Fig. 7.6c. 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.

7.6 Conclusions and future works

This chapter 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 $\tilde{S}_{\phi_\theta}^\varepsilon$ in equation (7.22) 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 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 equation (7.9) was used as an hard constraint in the ML-based optimization problem, and the Lagrangian relaxation in (7.19) 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 [86]. In addition, application of robustness in the sense of [78] will be investigated in order to preserve passivity under model uncertainty.

Chapter 8

Other works: Neural Model Predictive Control of Safety-Critical Biological Systems: an Application to Tumor-Immune Cancer Dynamics

Model Predictive Control (MPC) is a control strategy that effectively handles state constraints in safety-critical applications. However, it relies on the availability of an accurate model of the process to be controlled. In practical scenarios, deriving a model based on physical equations can be challenging. Often, it is difficult to obtain all necessary parameters, or the complexity of the model can hinder real-time control optimization. This work addresses such a challenge by proposing a neural MPC controller that learns an approximation of system dynamics from data, enabling constraint-aware control without prior model knowledge. The proposed methodology is applied to a particular safety-critical system, namely a model for cancer control through chemotherapy and immunotherapy in the presence of a clinical constraint that is crucial for patient's health. The formulated control strategy is validated through in-silico simulations, exhibiting performances comparable to the case of standard MPC with complete model knowledge and to the case of Optimal Control using Pontryagin's Maximum Principle. It also exhibits robustness with respect to uncertainty on a pivotal parameter of the tumor-immune treatment model.

8.1 Introduction and Related Works

8.1.1 Problem Statement

Model Predictive Control is a control methodology capable of combining performance optimization with closed-loop stability and satisfaction of safety-critical constraints on states and inputs [87]. Since it employs a dynamic model to predict the future behaviour of the controlled system, the effectiveness of the control strategy is directly dependent on the accuracy of the model. Therefore, the implementation of MPC traditionally requires a thorough understanding of the physical phenomena governing the system's dynamics, and then the development of a mathematical representation that accurately captures such dynamics. Moreover, the problem of identifying suitable

system equations is coupled with the problem of parameter estimation, based on the observations of different instances of the physical systems, e.g., in the context of medical applications, based on the observation of patients with diverse characteristics [88].

The quality of a model is closely tied to its complexity, as models with nonlinear functions and a high number of parameters can more precisely capture intricate system dynamics. However, since controllers must solve optimization problems in real time, balancing model complexity with accuracy is essential. Moreover, in the context of biological systems it is often not possible to derive a model of the process [89]. This has driven interest in Neural Networks (NNs) for MPC, as they can provide complex yet computationally efficient models, and make use of past clinical data when no explicit model is available. Thanks to their nonlinear structure and large number of parameters, NNs proved to be accurate approximation tools [90].

This work sets the foundation for addressing the challenge of controlling systems with unknown or unmodeled dynamics in the presence of safety-critical constraints. To synthesize a suitable controller, a data-driven approach is employed to identify a neural representation of such dynamics.

8.1.2 Related Works and Proposed Innovations

MPC can effectively handle linear, nonlinear and multi-variable systems in the presence of constraints, disturbances and model uncertainties. Recently, NNs have been used in the framework of MPC to approximate or even augment the system dynamics, thus using the NN as the MPC prediction model. In [91] the authors used a simple NN to approximate the dynamics of a nonlinear chemical reactor without assuming any particular structure for the system dynamics; the authors of [92] compare different deep NN architectures, namely various subclasses of recurrent neural networks, to predict the behaviour of an underwater vehicle; in [93], a multi-layer feedforward neural network is used to approximate the Duffing equation [94], that describes the behaviour of damped and driven oscillators; in the context of biological systems, authors of [95] use a nonlinear autoregressive NN with external input to approximate the dynamics of the artificial pancreas to control type 1 diabetes in rats; the authors of [96] apply a Radial Basis Function NN to model spatial cellular uptake in tissue engineering.

A limitation of most of the mentioned works relies in the fact that they primarily focus on system identification rather than direct control synthesis. While these studies successfully demonstrate the ability of neural networks to approximate complex system dynamics, they do not fully integrate the identified models into a real-time control framework. Moreover, many approaches lack a systematic way to incorporate safety constraints, which is crucial in applications such as biomedical control in the healthcare context. Additionally, the robustness of these data-driven models to uncertainties and variations in system parameters is often not analyzed, limiting their applicability in real-world scenarios where patient-specific or environment-specific adaptations are required. NNs can also be integrated in MPC control schemes to derive control actions straightforwardly using Lyapunov-based analysis [97]. While such approach provides theoretical guarantees, often requires explicit knowledge of a suitable Lyapunov candidate function or rely on restrictive structural assumptions about the system dynamics. In applications involving complex or poorly understood biological processes, such assumptions are difficult to satisfy. Therefore, this work adopts a more flexible data-driven modeling strategy that focuses on robustness and constraint satisfaction without presupposing a known stability certificate.

The main contributions of this work are summarized as follows:

- Data-driven modeling of biological systems, and in particular of tumor growth dynamics, is well documented in the literature [98–100]. However, to the best of the authors' knowledge, this study is the first to propose a data-driven control methodology in such relevant context;
- The proposed data-driven MPC controller enables the exact solution of the control problem. In the considered use-case using the modified Stepanova tumor growth model, it achieves similar control performance of model-based control compared to academic literature [101];
- The proposed data-driven MPC controller proves to be robust with respect to variations of a critical parameter of the tumor-immune system characterizing the treatment needs of specific patients;
- Contrary to other methodologies, e.g., [101, 102], the use of the proposed data-driven MPC enables the explicit formulation of safety-critical clinical requirements in terms of optimization constraints.

The remainder of the chapter is organized as follows. The fundamentals on neural MPC are described in Section 8.2; Section 8.3.3 illustrates the derivation of the proposed controller and the tumor-immune cancer dynamics system to be controlled; Section 8.4 validates the proposed controller by means of in-silico simulations; finally, Section 8.5 draws conclusions and outlines future works.

8.2 Background on Neural MPC

Neural MPC is a data-driven control method that uses a neural network as prediction model [103]. Considering a generic discrete-time system with state $x \in \mathcal{X} \subseteq \mathbb{R}^n$ and control $u \in \mathcal{U} \subseteq \mathbb{R}^p$, an MPC controller solves the following optimization problem at any time instant $t = 0, 1, \dots$, with initial condition x_t :

$$\min_{u_t, \dots, u_{t+K-1} \in \mathcal{U}} J(\hat{x}_t, \dots, \hat{x}_{t+K}, u_t, \dots, u_{t+K-1}) \quad (8.1a)$$

$$\hat{x}_{t+k} = f(\hat{x}_{t+k-1}, u_{t+k-1}), \quad (8.1b)$$

$$k = 1, \dots, K \quad (8.1c)$$

$$\hat{x}_{t+k} \in \mathcal{X}, \quad k = 1, \dots, K \quad (8.1d)$$

$$u_{t+k-1} \in \mathcal{U}, \quad k = 1, \dots, K, \quad (8.1e)$$

$$\hat{x}_t = x_t, \quad (8.1f)$$

where K is the prediction horizon, x_t is the measured state at time t , $\hat{x}_{t+k}$ is the vector of future predicted state at time $t+k$, u_{t+k-1} is the future control action at time $t+k-1$, $f : \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R}^n$ is the prediction model representing the dynamics of the controlled system along the prediction horizon. The first control action u_t is fed to the actual system at time t , and then the MPC problem (8.1) is solved again at step $t+1$ (*receding horizon*), starting from the new state measure x_{t+1} to account for the differences with the state $\hat{x}_{t+1}$ predicted at time t due to disturbances and model inaccuracies.

In case of data-driven MPC, a neural network model is used to perform predictions by modeling the one-step dynamics, i.e., $\hat{x}_{t+1} = h(x_t, u_t)$, thus replacing the function f to describe the system dynamics. The K -step dynamics can then be modeled by concatenating K one-step models (e.g., with $K = 2$, $\hat{x}_{t+2} = h(\hat{x}_{t+1}, u_{t+1}) = h(h(x_t, u_t), u_{t+1})$). The advantage lies in the training phase, which is simpler for such a one-step predictor than for a single neural network predicting all K steps, as usual in data-driven MPC ([104, 105]). Moreover, if a feed-forward neural network with linear and Rectified Linear Unit (ReLU) [106] activation functions is used for the one-step predictor, the work in [107] shows that the MPC problem (8.1) can be cast as a mixed integer quadratic program, whose exact solution can be efficiently found with standard solvers. A feed-forward NN is a parametric function approximator composed of layers of interconnected neurons. Each neuron computes a weighted sum of its inputs followed by a nonlinear transformation, known as an activation function. In a feed-forward architecture, connections only flow in one direction, from inputs to outputs, without recurrent loops. One common activation function is the Rectified Linear Unit (ReLU), defined as $\text{ReLU}(z) = \max(0, z)$, which introduces nonlinearity while being computationally efficient.

8.3 Neural MPC Control of Cancer Dynamics

8.3.1 Tumor-Immune Cancer Dynamics

The modified Stepanova model introduced in [108] effectively describes the tumor-immune interactions in the treatment of a cancer patient, and is used in this work to schedule chemotherapy and immunotherapy injection profiles. Chemotherapy targets and kills rapidly dividing cells, including both cancerous and immune cells, while immunotherapy works by stimulating immune cell growth: this immune boost helps the body to more effectively attack and control tumor development. In what follows, for clarity of notation, subscripts are reserved for discrete time indices, while numeric subscripts in parentheses denote state and control components.

Consider the system dynamics $f(x, u)$: the state is $x = [x_{(1)}, x_{(2)}]^T$ with $x_{(1)}, x_{(2)} \in \mathbb{R}_{\geq 0}$ representing the number of tumor cells and the immune cell density respectively; the control actions are $u_{(1)}, u_{(2)} \in [0, 1]$, representing the administration of chemo- and immunotherapy respectively as fractions of maximum possible dose, thereby enforcing biological toxicity constraints to ensure treatment safety; $f : \mathbb{R}_{\geq 0}^2 \times [0, 1]^2 \rightarrow \mathbb{R}_{\geq 0}^2$, is expressed by the following equations:

$$\begin{cases} \dot{x}_{(1)} = \mu_c x_{(1)} - \frac{\mu_c}{x_\infty} x_{(1)}^2 - \gamma x_{(1)} x_{(2)} - \kappa_x x_{(1)} u_{(1)}, \\ \dot{x}_{(2)} = \mu_I (x_{(1)} - \beta x_{(1)}^2) x_{(2)} - \delta x_{(2)} + \alpha + \\ \quad + \kappa_y x_{(2)} u_{(2)} - \eta u_{(1)} x_{(2)} \end{cases} \quad (8.2)$$

Since both therapies have negative side effects, it is necessary to limit their administration.

An interpretation of the various terms in the equations of the system's dynamics is illustrated in the following.

Equation 1 - tumor cell dynamics:

- $\mu_c x_{(1)}$, intrinsic tumor growth: tumor cells grow exponentially at rate μ_c when not treated. This is the natural growth without immune system or therapy interference;

- $-\frac{\mu_c}{x_\infty}x_{(1)}^2$, saturation / logistic growth limitation: models limited resources (e.g., oxygen, nutrients, space) slowing tumor growth as $x_{(1)}$ approaches the carrying capacity x_∞ . Without this term, tumor growth would be unrealistically unbounded;
- $-\gamma x_{(1)}x_{(2)}$, immune-mediated tumor killing: interaction between tumor cells ($x_{(1)}$) and immune cells ($x_{(2)}$). Larger immune population yields higher tumor cell death rate;
- $-\kappa_x x_{(1)}u_{(1)}$, chemotherapy-induced tumor cell death: chemotherapy dose $u_{(1)}$ kills tumor cells at rate proportional to κ_x .

Equation 2 - immune cell dynamics:

- $\mu_I(x_{(1)} - \beta x_{(1)}^2)x_{(2)}$, immune cell activation by tumor presence: for small to moderate tumor sizes, $x_{(1)}$ stimulates immune proliferation, like an antigen trigger). The $-\beta x_{(1)}^2$ term accounts for immune suppression at large tumor sizes (e.g., due to tumor-secreted immunosuppressive factors);
- $-\delta x_{(2)}$, natural immune cell decay: immune cells die naturally or are removed from circulation at rate δ ;
- $+\alpha$, baseline immune recruitment: represents constant production of immune cells by the body;
- $+\kappa_y x_{(2)}u_{(2)}$, immunotherapy-induced immune boost: immunotherapy dose $u_{(2)}$ increases immune cell proliferation proportionally to κ_y ;
- $-\eta u_{(1)}x_{(2)}$, chemotherapy-induced immune suppression: chemotherapy not only attacks tumor cells but also harms immune cells. The strength of this negative effect is proportional to the parameter η .

A safety-critical constraint on the immune cell density for the tumor treatment is that:

$$x_{(2)} > 0.25, \quad (8.3)$$

to prevent immune system collapse, that could lead to uncontrolled tumor proliferation [101]. This constraint reflects the negative side-effects of chemotherapy, which can severely reduce immune cell populations. If $x_{(2)}$ drops too low, the immune system can no longer contribute meaningfully to controlling tumor growth. This safety-critical consideration is essential for the clinical viability of the proposed control strategy.

The model parameters are defined in Table 8.1, that also reports the values used in the in-silico simulations of Section 8.4. Such specific parameter set used for the model was selected to produce distinct benign and malignant equilibria that are consistent with clinically observed tumor-immune dynamics [101], with the possibility of both long-term tumor control and uncontrolled growth, depending on the initial state and applied therapy.

Table 8.1: Parameters of the Modified Stepanova Model

Parameter	Definition	Numerical value
μ_c	Tumor growth rate	$0.5599 \cdot 10^7$ cells/-day
μ_I	Tumor stimulated proliferation rate	0.00484 day^{-1}
α	Rate of immune cells influx	0.1181 day^{-1}
β	Inverse threshold	0.00264
γ	Interaction rate	$1 \cdot 10^7$ cells/day
η	Interaction rate between chemotherapy and immune cells	$[0, 2]$
δ	Immune cells decay rate	0.37451 day^{-1}
κ_x	Chemotherapeutic killing parameter	$1 \cdot 10^7$ cells/day
κ_y	Immunotherapy injection parameter	$1 \cdot 10^7$ cells/day
x_∞	Fixed carrying capacity	780×10^6 cells

8.3.2 Background on System Controllability Analysis

Following [101], the controllability of the system described in (8.2) is examined below. The system can be expressed in the form:

$$\begin{cases} \dot{x} = f(x(t), u(t)) = f(x) + g_1(x)u_{(1)}(t) + g_2(x)u_{(2)}(t) \\ y = h(x(t)), \quad x_0 = (x_{(1),0}, x_{(2),0}) \end{cases}, \quad (8.4)$$

where:

$$g_1 = \begin{bmatrix} -k_x x_{(1)} \\ -\eta x_{(2)} \end{bmatrix}, \quad g_2 = \begin{bmatrix} 0 \\ k_y x_{(2)} \end{bmatrix}. \quad (8.5)$$

To assess controllability, the nonlinear controllability rank condition is applied [8]: in a neighborhood M of x_0 , if $\dim\langle \text{ad}_f | R(f) \rangle = n$, where n is the dimension of x , then the system satisfies the controllability rank condition. Here $\langle \text{ad}_f | R(f) \rangle$ is the controllability distribution where ad_f denotes the adjoint operator, and $R(f)$ is the distribution spanned by $f(\cdot, u)$. The controllability matrix is constructed as:

$$C = (g_1, g_2, [g_1, g_2], \dots), \quad (8.6)$$

where $[g_1, g_2]$ is the Lie bracket [8]:

$$[g_1, g_2] \equiv \frac{\partial g_2}{\partial x} g_1 - \frac{\partial g_1}{\partial x} g_2. \quad (8.7)$$

If C has full rank n at x_0 , the system is locally controllable around that point. Since the system has $n = 2$ state variables, the nonlinear controllability rank condition requires that the controllability matrix C has full rank $n = 2$ at the point of interest.

Evaluating the first two columns of C yields:

$$C = (g_1, g_2) = \begin{bmatrix} k_x x_{(1)} & 0 \\ -\eta x_{(2)} & k_y x_{(2)} \end{bmatrix}. \quad (8.8)$$

These vectors are linearly independent, ensuring full rank. Substituting the equilibrium points, that are visible in Fig. 8.1 and described in Section 8.4, gives the following full-rank matrices for the healthy and unhealthy equilibria, respectively:

$$C_1 = \begin{bmatrix} 3.5 \times 10^8 & 0 \\ -0.53 & 5.3 \times 10^6 \end{bmatrix}, \quad (8.9)$$

$$C_2 = \begin{bmatrix} 7.4 \times 10^8 & 0 \\ -0.032 & 3.2 \times 10^5 \end{bmatrix}. \quad (8.10)$$

Since both C_1 and C_2 have rank 2, the system is locally controllable around each equilibrium point. Therefore, the combined administration of chemotherapy and immunotherapy can, in principle, influence both tumor and immune cell populations so as to steer the system toward the desired

healthy equilibrium state.

8.3.3 Neural MPC Control Methodology

To develop a data-driven controller, it is assumed that a dataset D of n_D input–output pairs $\{(x^i, u^i), y^i\}$ is available, as an instance from past clinical data; x^i denotes the k -th sample of the system state, u^i the control action value, and y^i the labeled output. As specified in Section 8.4, in this chapter we will consider synthetic data and compute y^i by sampling the dynamics equations (8.2).

The set D is the training set of the neural network h , whose prediction will be denoted by $\hat{y}^i = h(x^i, u^i)$ and validated against the labeled values y^i . A feed-forward neural network with ReLu and linear activation functions was selected, since it resulted adequate for representing the system dynamics in (8.2), as discussed in Section 8.4; moreover, the work in [107] shows that this type of network yields a system representation that is particularly well-suited for integration into the MPC problem.

With reference to the MPC control problem in (8.1), as explained in Section 8.2, the neural network constitutes the prediction model and is used as the neural counterpart of the function f in constraints (8.1b). The predicted states $\hat{x}_k$ are computed by concatenating K one-step predictors: constraints (8.1b) are written as:

$$\hat{x}_{t+k} = h(x_{t+k-1}, u_t), k = 1, \dots, K, \quad (8.11)$$

meaning that the k -th predicted state is computed as:

$$\hat{x}_{t+k} = h(\dots(h(h(x_t, u_t), u_{t+1}), \dots, u_{t+k-1}), k = 1, \dots, K. \quad (8.12)$$

Thanks to the selected feed-forward network, constraints (8.1b) can be cast as mixed integer linear constraints, as shown in [107], with the introduction of additional slack optimization variables.

8.4 Simulations and Results

The Neural MPC was developed based on the model (8.2), as the MPC prediction model was trained on data generated by (8.2). Moreover, (8.2) is used as the controlled system in the simulations – i.e., it is regarded as a real patient.

The simulations were carried out on a machine learning machine equipped with an Intel Core i9 9900k processor with 128 GB of RAM and a Nvidia GeForce RTX 3090 as GPU.

A feed-forward neural network was trained on $n_D = 10^3$ input-output pairs $\{(x^i, u^i), y^i\}$, generated through random excitation of the system, constituting the data set D . In this chapter, we consider synthetic data and compute y^i by sampling the output of equations (8.2) as $y^i = f_\Delta(x^i, u^i)$, where $\Delta = 0.1s$ is the sampling time and f_Δ returns the output of equations (8.2), starting from $x_0 = x^i$ and with constant control action $u_t = u^i$ in the time interval $t \in [0, \Delta]$, obtaining $y^i = x(\Delta)$.

The NN is structured as follows: the input layer comprises four neurons, corresponding to the input variables $x_{(1)}, x_{(2)}, u_{(1)}, u_{(2)}$; three hidden layers with 64, 32, 16 neurons respectively were utilized, with ReLU as activation function, providing non-linearity to the model and enhancing its

capability to learn complex relationships within the data; the output layer consists of two neurons corresponding to the predicted state variables $y_{(1)}$, $y_{(2)}$. The NN was trained using a backpropagation algorithm with Levenberg-Marquardt optimization [109]. The dataset was partitioned into training (80%) and testing (20%) subsets to evaluate model generalization. Mean Squared Error (MSE) was used as the loss function to measure prediction accuracy during the NN training, since it heavily penalizes larger errors due to the squaring of the residuals. This property helps the NN focus on reducing large deviations, and stabilizes gradient-based optimization, leading to smoother convergence during training.

The effectiveness of the NN modeling performances is assessed on the test set demonstrating a strong correlation between true and predicted values. Fig. 8.1 illustrates the phase portrait of the uncontrolled system, showing that the NN approximation of system dynamics effectively replicates the true system dynamics. The accuracy of the neural approximation is quantified using the Mean Absolute Error (MAE) between the true system dynamics and the states predicted by the NN. MAE is used since it provides a more interpretable and robust measure of overall prediction accuracy, less sensitive to outliers than MSE, that was used during the NN training. Let $x_t^{(i)}$ represent the state of the true dynamics at time $t = 1, \dots, T$, and $\hat{x}_t^{(i)}$ denote the state predicted by the NN given initial conditions $(x_{(1),0}^{(i)}, x_{(2),0}^{(i)})$, $i = 1, \dots, N$, error metrics are defined as:

$$\begin{aligned} \text{MAE}_{x_{(1)}} &= \frac{1}{N \cdot K} \sum_{i=1}^N \sum_{k=1}^K \left| x_{(1),t+k}^{(i)} - \hat{x}_{(1),t+k}^{(i)} \right|, \\ \text{MAE}_{x_{(2)}} &= \frac{1}{N \cdot K} \sum_{i=1}^N \sum_{k=1}^K \left| x_{(2),t+k}^{(i)} - \hat{x}_{(2),t+k}^{(i)} \right|. \end{aligned} \quad (8.13)$$

The MAE is computed as follows:

1. Simulate trajectories for N initial conditions using both the true dynamics and the NN model;
2. Store all states $x_t^{(i)}$ and $\hat{x}_t^{(i)}$ for $i = 1, \dots, N$ and $t = 1, \dots, T$;
3. Compute absolute differences between true and NN-predicted values for each state variable at every time step;
4. Average such absolute errors across all initial conditions and time steps to obtain $\text{MAE}_{x_{(1)}}$ and $\text{MAE}_{x_{(2)}}$;

The results corresponding to the NN approximation plotted in Figure 8.1 are $\text{MAE}_{x_{(1)}} = 3.6324$ and $\text{MAE}_{x_{(2)}} = 0.0113$. The significance of such results can be assessed by comparing them with respect to the dynamic ranges of the corresponding state variables in the considered simulation.

For $x_{(1)}$, the dynamic range is given by $\Delta x_{(1)} = 740 - 10 = 730$ [101]. The value $\text{MAE}_{x_{(1)}} = 3.6324$ is therefore $\frac{\text{MAE}_{x_{(1)}}}{\Delta x_{(1)}} \times 100 = \frac{3.6324}{730} \times 100 \approx 0.49\%$ of $\Delta x_{(1)}$. This implies that the model accurately captures the critical dynamics that distinguish healthy and unhealthy states. Similarly, for $x_{(2)}$, the dynamic range is $\Delta x_{(2)} = 1.18 - 0.03 = 1.15$ [101]. The value $\text{MAE}_{x_{(2)}} = 0.0113$ is then $\frac{\text{MAE}_{x_{(2)}}}{\Delta x_{(2)}} \times 100 = \frac{0.0113}{1.15} \times 100 \approx 0.98\%$ of $\Delta x_{(2)}$. This indicates that the NN closely tracks $x_{(2)}$, which is especially important since maintaining $x_{(2)} > 0.25$ is a crucial constraint.

The considered cost function (8.1a) is defined as:

$$J = \sum_{k=1,\dots,K} (\|\hat{x}_{t+k}\|_2^Q + \|u_{t+k-1}\|_2^R), \quad (8.14)$$

where the weighted 2-norm with vector ξ and weighting matrix P is defined as $\|\xi\|_2^P = \xi^T P \xi$, and $R, Q \in \mathbb{R}^{2 \times 2}$ are non-singular matrices expressing the trade-off between performances and control cost.

To use constraints (8.1d) to enforce the requirement $x_{(2)} > 0.25$, it is sufficient to define $X = \mathbb{R}_{\geq 0} \times [0.25, \infty]$, whereas, for the control set, $U = [0, 1]^2$. Note that constraints (8.1d) and (8.1e) are linear ones.

In conclusion, a mixed integer quadratic programming problem is defined as the minimization of the quadratic cost function (8.14) subject to the mixed integer linear constraints (8.11), the linear constraints (8.1d) and (8.1e), with initial conditions (8.1f). Thus, the resulting optimization problem is the following.

$$\min_{u_t, \dots, u_{t+K-1} \in \mathcal{U}} \sum_{k=1}^K (\|\hat{x}_{t+k}\|_Q^2 + \|u_{t+k-1}\|_R^2) \quad (8.15a)$$

$$\text{s.t.} \quad \hat{x}_{t+k} = h(\hat{x}_{t+k-1}, u_{t+k-1}), \quad (8.15b)$$

$$k = 1, \dots, K, \quad (8.15c)$$

$$\hat{x}_{t+k} \in \mathcal{X} := \mathbb{R}_{\geq 0} \times [0.25, \infty), \quad (8.15d)$$

$$k = 1, \dots, K, \quad (8.15e)$$

$$u_{t+k-1} \in \mathcal{U} := [0, 1]^2, \quad (8.15f)$$

$$k = 1, \dots, K, \quad (8.15g)$$

$$\hat{x}_t = x_t, \quad (8.15h)$$

This problem can be efficiently solved by standard solvers, namely it is tractable for the instance sizes considered in this MPC control case.

Define:

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad R = \begin{bmatrix} 10^6 & 0 \\ 0 & 10^6 \end{bmatrix}, \quad (8.16)$$

where Q is such that the variable $x_{(2)}$ is not evaluated in the cost function, since the number of immune cells must not be minimized; R assigns a strong penalty to high or rapid dosing of chemo- and immunotherapy. This choice promotes minimum necessary dosing and limits toxicity.

Equation (8.14) represents the K -step counterpart of the cost function defined in [101]. The non-linear optimization (8.1) with f given by (8.2) was solved numerically using *fmincon* by MATLAB [110].

As shown in Fig. 8.1, the uncontrolled system ($u_{(1)} = u_{(2)} = 0$) has a benign equilibrium in which the tumor persists in a microscopic state. The healthy equilibrium is $(x_{(1)}, x_{(2)}) = (34.98, 0.53)$. There also exists an unhealthy equilibrium at $(x_{(1)}, x_{(2)}) = (735.9, 0.032)$. The objective of the

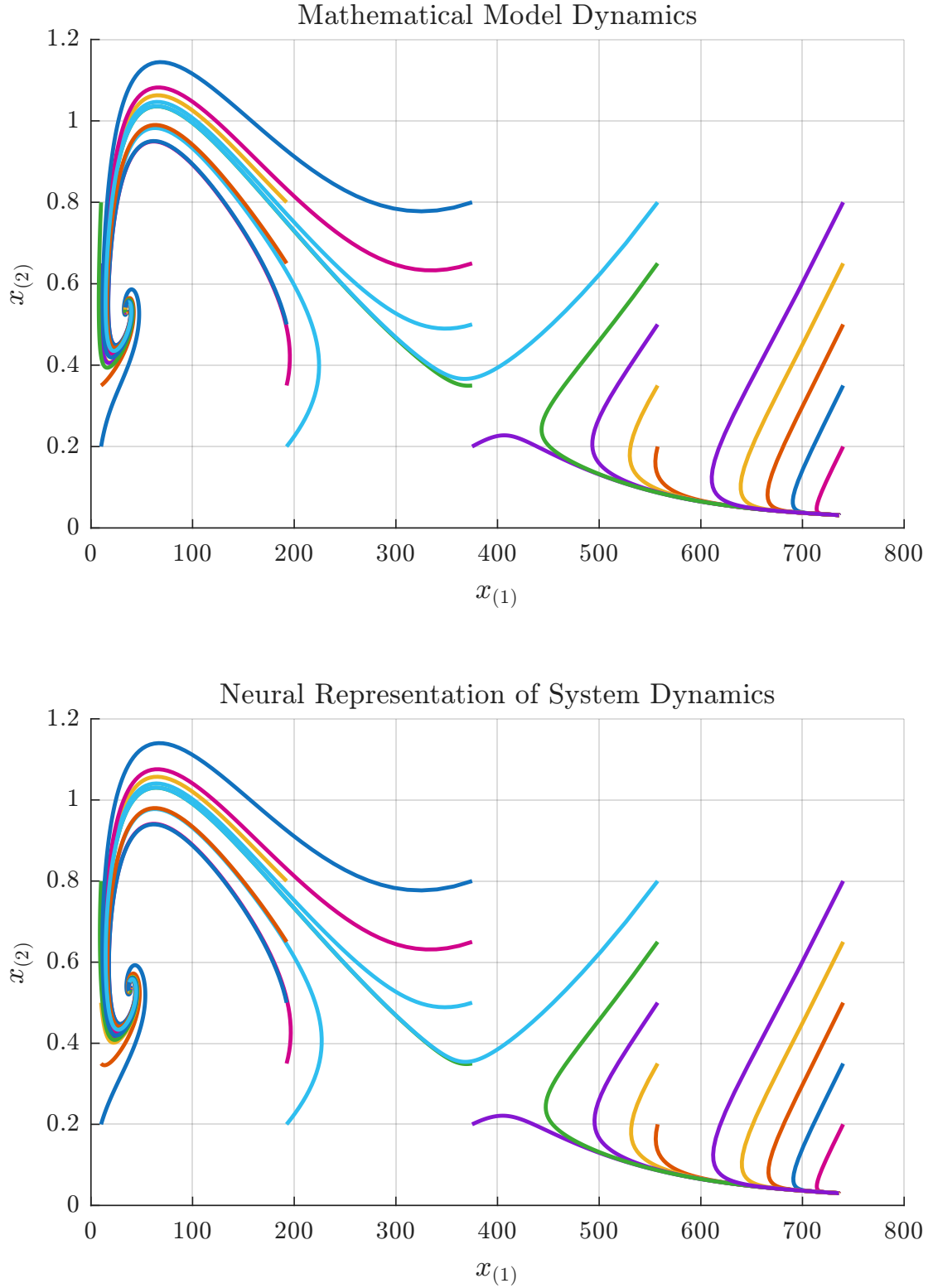


Figure 8.1: Phase portrait of uncontrolled system ($u_{(1)} = u_{(2)} = 0$) in the case of mathematical model dynamics (top) and in the case of the NN representation of such dynamics (bottom). 25 trajectories lasting 350 time steps each are plotted. Note that the plot exhibits a benign equilibrium in which the tumor persists in a microscopic state at $(x_{(1)}, x_{(2)}) = (34.98, 0.53)$ while there also exists an unhealthy equilibrium at $(x_{(1)}, x_{(2)}) = (735.9, 0.032)$.

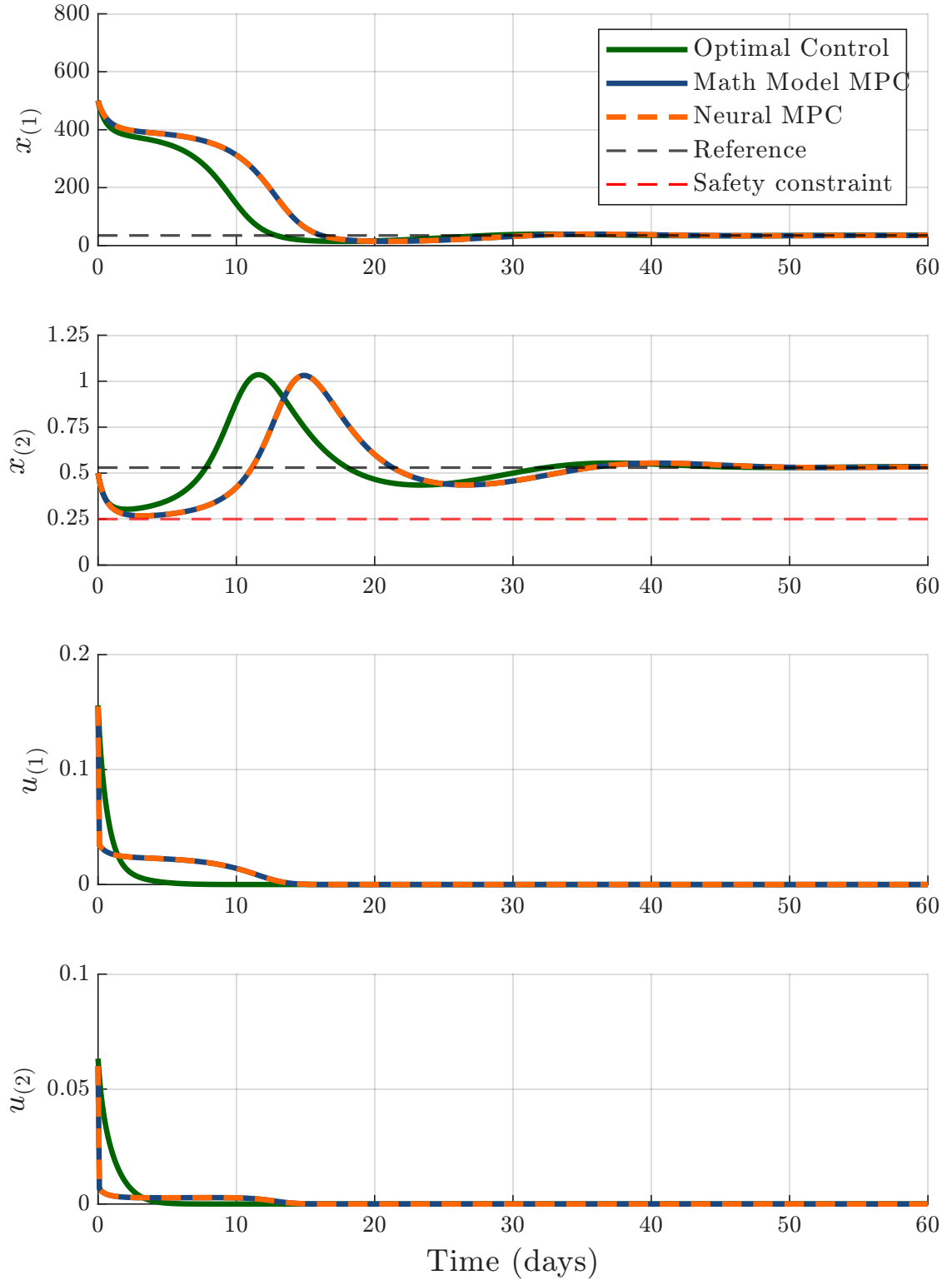


Figure 8.2: Comparison of the evolution of states and control efforts in the case of Optimal Control, MPC with perfect knowledge of the system mathematical model, and Neural MPC. In all of the three cases, the target reference (dashed black line) is reached and the safety constraint (dashed red line $x_{(2)} = 0.25$) is satisfied.

considered control problem is to stabilize the system to the healthy equilibrium while satisfying the constraint $x_{(2)} > 0.25$. The initial condition $(x_{(1),0}, x_{(2),0}) = (500, 0.5)$ is selected to represent a clinically relevant scenario, since it lies outside the basin of attraction of the healthy equilibrium and would converge to the malignant equilibrium in the absence of treatment, thereby necessitating active control. Biologically, this corresponds to an intermediate-to-late stage tumor with partial immune competence, a situation in which spontaneous remission is unlikely and therapeutic intervention is still viable. Therefore, this choice of initial condition allows for a meaningful evaluation of the controller's ability to steer the system toward the healthy equilibrium while respecting the constraint

The performances of the proposed control methodology are evaluated in comparison to the optimal control strategy developed in [101], that uses the Pontryagin's Maximum Principle [111]. The Hamiltonian is as follows:

$$\begin{aligned}
H = & q_1 x_{(1)}^2 + q_2 x_{(2)}^2 + r_1 u_{(1)}^2 + r_2 u_{(2)}^2 \\
& + \lambda_1 \left(\mu_C x_{(1)} - \frac{\mu_C}{x_\infty} x_{(1)}^2 - \gamma x_{(1)} x_{(2)} - k_x x_{(1)} u_{(1)} \right) \\
& + \lambda_2 \left(\mu_1 x_{(1)} x_{(2)} - \mu_1 \beta x_{(1)}^2 x_{(2)} - \delta x_{(2)} + \alpha \right. \\
& \left. + k_y x_{(2)} u_{(2)} - \eta u_{(1)} x_{(2)} \right).
\end{aligned} \tag{8.17}$$

The corresponding adjoint equations are:

$$\begin{aligned}
\dot{\lambda}_1 = & -\frac{\partial H}{\partial x_{(1)}} \\
= & -2q_1 x_{(1)} + \lambda_1 \left(-\mu_C + 2\frac{\mu_C}{x_\infty} x_{(1)} + \gamma x_{(2)} + k_x u_{(1)} \right) \\
& + \lambda_2 \left(-\mu_1 x_{(2)} + 2\mu_1 \beta x_{(1)} x_{(2)} \right).
\end{aligned} \tag{8.18}$$

$$\begin{aligned}
\dot{\lambda}_2 = & -\frac{\partial H}{\partial x_{(2)}} = -2q_2 x_{(2)} + \lambda_1 \gamma x_{(1)} \\
& + \lambda_2 \left(-\mu_1 x_{(1)} + \mu_1 \beta x_{(1)}^2 + \delta - k_y u_{(2)} + \eta u_{(1)} \right)
\end{aligned} \tag{8.19}$$

The optimal control laws are obtained by imposing conditions $\frac{\partial H}{\partial u_{(i)}} = 0$ for $i = 1, 2$, yielding the following solutions:

$$u_{(1)} = \frac{\lambda_1 k_x x_{(1)} + \lambda_2 \eta x_{(2)}}{2r_1}, \quad u_{(2)} = -\frac{\lambda_2 k_y x_{(2)}}{2r_2}. \tag{8.20}$$

The results of the comparison are plotted in Fig. 8.2: the Neural MPC (that uses the NN as prediction model) with prediction horizon $K = 10$, effectively approximates the solution of the Mathematical Model MPC, in which system (8.2) is known and used as prediction model. Also, the figure shows that their performances are comparable to the ones obtained by the optimal controller (8.19), (8.20).

The durations of chemotherapy and immunotherapy span several days, which is consistent with clinically observed administration schedules for certain cancer types [101]. In practice, chemotherapeutic agents are often delivered in cycles lasting from hours to multiple days, followed by rest

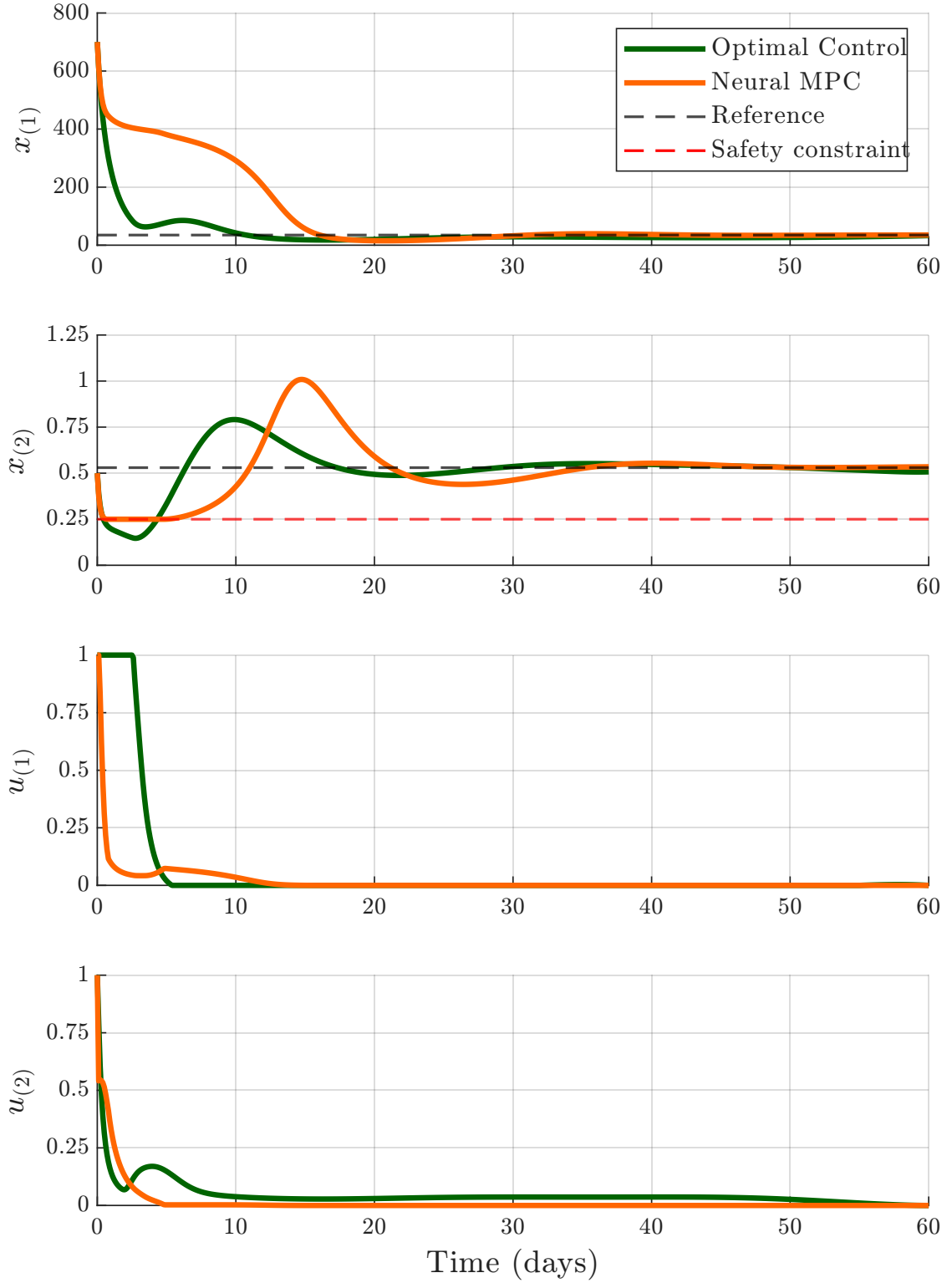


Figure 8.3: Simulation with initial conditions $(x_{(1),0}, x_{(2),0}) = (700, 0.5)$: Optimal Control violates the safety constraint (dashed red line), while the proposed Neural MPC controller does not.

periods to mitigate toxicity, while immunotherapeutic interventions are administered either as short-term infusions or as repeated daily doses over a similar timescale. The specific duration and dosing profiles adopted in this work were chosen to capture these clinically plausible time windows while enabling a tractable simulation framework for the MPC optimization problem.

The simulation in Fig. 8.2 shows that a combination of chemo- and immunotherapy effectively moves the system from its malignant initial conditions to its healthy equilibrium. These results are reasonable when considering the ultimate goal of attacking the tumor cells while also supporting the immune system.

In this setup, the safety constraint is satisfied also in all three cases, but there are initial conditions from which the optimal controller violates the safety constraint, due to the lack of explicit constraint handling. As an instance, by setting initial conditions corresponding to a higher number of tumor cells $(x_{(1),0}, x_{(2),0}) = (700, 0.5)$, Fig. 8.3 shows that, contrary to the proposed Neural MPC, the optimal controller violates the constraint. Note that the control effort is saturated to 1 since $u_{(i)} = 1$, $i = 1, 2$ represents the maximum administrable therapy.

Finally, the proposed Neural MPC control methodology is evaluated in terms of robustness with respect to uncertainty on the parameter η which, as discussed in Sec. 8.3.3, represents the interaction between chemo- and immunotherapy and is a pivotal parameter that significantly differentiates patients in how they respond to chemo- and immunotherapy. The predictions of the MPC algorithm are performed leveraging the NN model derived from data of a patient with nominal $\eta = 1$, while the controlled system corresponds to a patient with different values of η as follows: $\eta = [0, 0.5, 1, 1.5, 2]$. Fig. 8.4 shows that the controller steers the system to the healthy equilibrium in all cases while still satisfying the safety constraint on $x_{(2)}$, despite discrepancies between the model dynamics (trained with $\eta = 1$) and the true plant dynamics (with different values of η in the range $[0, 2]$). Such results suggest that the Neural MPC framework can adapt to patient-specific variations in η without requiring retraining. This is particularly relevant in oncology, where inter-patient heterogeneity often complicates treatment personalization.

8.5 Conclusions and Future Works

This work proposes a neural MPC control scheme for safety-critical biological systems; leveraging a neural network as prediction model effectively representing the dynamics of the system, a controller showing constraint satisfaction is derived within an MPC framework.

In-silico simulations show control performances that are comparable to the ones obtained through Pontryagin's Maximum Principle, while also highlighting the inherent robustness capabilities of the proposed approach in the presence of uncertainties on a critical parameter describing the diverse characteristics of the specific patients. The proposed method's generalizability suggests it could also be applied to other safety-critical biological systems beyond the presented case study.

Future research work aims at fine tuning the proposed control scheme integrating real-time data from clinical settings, incorporating adaptive learning mechanisms to update the neural model in real-time and to improve performance in dynamically changing environments, and extending the proposed approach to cases where tumor state measurements are delayed or noisy.

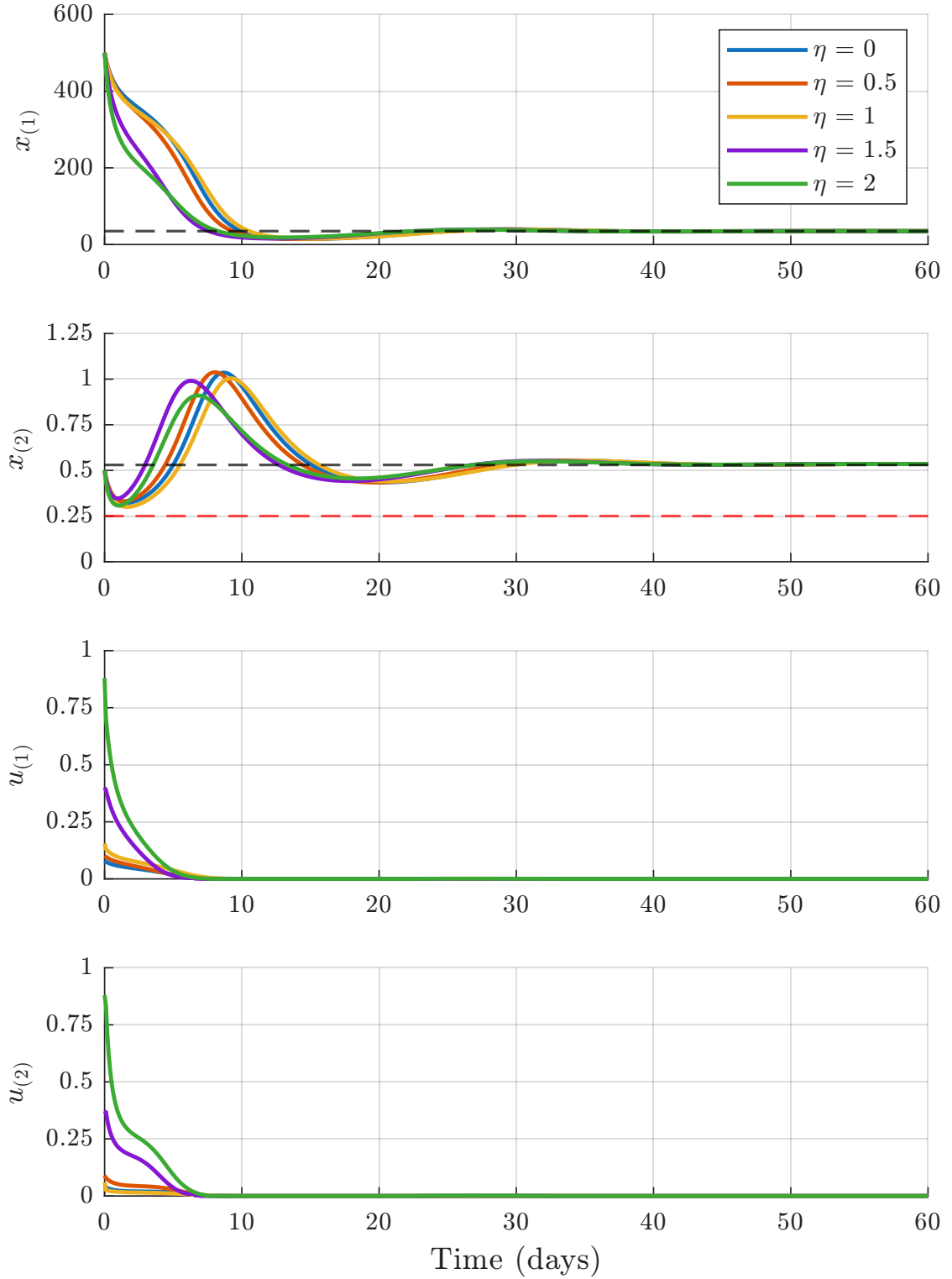


Figure 8.4: Robustness of the proposed Neural MPC control strategy under uncertainty on η , with $\eta = [0, 0.5, 1, 1.5, 2]$. The dashed black and red lines represent the reference and safety constraint respectively, as in previous figures.

Chapter 9

Other works: Behavioural Cloning for Serious Games in Support of Pediatric Neurorehabilitation

Behavioural Cloning is a Machine Learning method concerning how a machine attempts to autonomously mimic the actions of a human, or in general a complex controller, performing a given task. This work innovatively exploits Behavioural Cloning in support of Pediatric Neurorehabilitation. In particular, an Artificial Neural Network Classifier has been implemented to autonomously adapt the difficulty, through a set of tunable parameters, of a Serious Game that was specifically developed to stimulate some relevant cognitive capabilities of the patient. Data augmentation via Behavioural Cloning allows such autonomous difficulty adaptation system to improve its classification performances and, thus, to enforce a control logic that, in turn, improves the effectiveness of the cognitive training. The system is validated through an experimental assessment on a Serious Game that trains motor coordination: experimental results of children gameplay are analyzed and discussed.

9.1 Introduction

One of the main limitations of Neural Networks (NN) in automatic control applications relies in the fact that the training phase requires large amounts of data. However, in practical applications, the problem of data paucity is often experienced, as in the context of neurorehabilitation [112, 113]. To overcome such issue, synthetic data can be generated through Behavioural Cloning (BC), that is a sub-class of Machine Learning concerning how a machine learns to perform a task by attempting to imitate a human, or in general a complex controller, performing it [114]. BC is then a type of supervised learning where the training data consists of recorded samples of the performed actions of a human/controller in certain situations/states.

This study aims at implementing BC to tackle the problem of data paucity in a specific application in the neurorehabilitation domain. An app for mobile devices was implemented consisting of an ad-hoc Serious Game, i.e., a game designed for educational and/or rehabilitation purposes [115]. The game is based on a musical rehabilitation technique, named Rhythmic Auditory Stimulation, that supports recovery and training of motor coordination by means of synchronizing movements

with the rhythmic accents of music [116]. The work was developed in three phases. In the first phase, a beta-version of the app was released, in order to collect data, where the difficulty of the game is adapted by following conventional static threshold-based rules, i.e., the difficulty of the next match is set based on the score of the last match via simple comparison to predefined thresholds. Then, in the second phase, data augmentation via BC was performed on the collected gameplay dataset; the resulting augmented dataset was used to train a NN Classifier for autonomous difficulty adaptation. Finally, in the third step, a new version of the app was released, where the trained NN is exploited to automatically deploy control actions online aimed at adapting difficulty based on the player's performances in order to achieve a stronger cognitive training.

The main contributions of this work are outlined in the following:

- a Serious Game for pediatric neurorehabilitation was developed from scratch under the guidance of pediatric doctors and psychologists;
- a NN architecture was implemented for developing an autonomous difficulty adaptation system;
- a BC method for data augmentation was designed to enable an effective training of the NN.

The proposed system was validated through analysis and discussion of experimental results.

The document is organized as follows: section 9.2 reports some related literature works on the use of BC for data augmentation in video games; section 9.3 introduces the originally conceived rehabilitation Serious Game, presents the NN architecture for autonomous difficulty adaptation via suitable control actions and describes the BC implementation; section 9.4 analyzes and discusses the experimental validation of the proposed system; finally, section 9.5 draws conclusions and suggests some future works.

9.2 Related works

BC can be applied in the context of video games so as to train a computer to play the game by mimicking human gameplay samples.

BC may outperform other techniques such as Reinforcement Learning (RL) [117], that also has been successfully applied to generate excellent synthetic players [118], even if it often requires unpractical training time [119]. Indeed, the main advantage of BC is the ease of data collection, whereas RL requires the costly creation of a learning environment for training the agent. On the other hand, BC results more sensitive to the quality of the collected data with respect to RL solutions.

Several additional alternative methods exist, working on demonstration data provided by a human. These include Generative Adversarial Imitation Learning [120], Batch-Constrained Q-Learning [121], and Inverse Reinforcement Learning [122]. Some of these methods have been successfully used to achieve better performance than simpler BC algorithms, at the price of a more complex implementation and data collection. This work focuses on the BC approach, since it is a proof of concept, disposing of little data, for a following project that will be developed for the actual validation of the system with pediatric patients in an hospital, disposing of more data. Thus, at this stage the main objective is not to fully optimize performances, but to preliminarily show the

efficacy of the proposed system. Further developments shall be considered in the future when more elaborate games will be involved and larger data collection campaigns will be conceived.

In the context of games and Serious Games, in [119], the authors demonstrated that BC in a real-time strategy game was able to match in-game rating of 84% of human players. In [123] it is shown that a BC agent is able to reach 67% of human players' score for item collection tasks, but cannot mimic adequately navigation tasks. In [124], the author conducted an experiment, in which expert human players were required to watch recorded gameplays and to guess whether they had been played by a human or by a BC agent: 48% of the experts found the BC agent to be human-like.

In scenarios where data scarcity is particularly severe, Dataset Aggregation [125] is an iterative algorithm that can improve the performance of BC agents. Firstly, an agent is trained with a supervised learning method based on human demonstration data. Then, the agent collects more samples by imitating the human behaviour and each of such collected samples is labelled by a human expert in order to validate or disconfirm the convenience of integrating such new samples. This procedure is repeated iteratively. Manual labelling is a laborious process compared to the standard BC procedure to collect training data, and therefore such method is not a good choice for this work, since it can be assumed that doctors in the rehabilitation context would hardly have time to perform labelling.

9.3 Adaptive Serious Game Design based on Neural Networks and Behavioural Cloning

For this study, a Serious Game for supporting Pediatric Rehabilitation, named *RhythmRun* [126], has been originally conceived and implemented by the authors employing Unity Game Engine. The game design is based on Rhythmic Auditory Stimulation [116], a music rehabilitation technique used for supporting the rehabilitation of movements that are naturally rhythmic, such as gait. It consists of presenting a series of auditory stimuli at a fixed cadence, and movements have to be synchronized to such rhythmic cadence. The auditory stimuli can be either a basic sequence of rhythmic sounds or a piece of music with clear temporal pattern elements. It has been shown that some patients who are not able to perform a movement voluntarily, may be able to perform the same movement as a response to an auditory stimulus [127]. Indeed, musical activity has proven to be a powerful stimulus for brain plasticity, that is the ability of neuronal networks to evolve through growth and reorganization [128]. The game is intended for supporting the rehabilitation of children with motor impairments, such as paresis, which is often caused by stroke, by training timing functions and enhancing motor planning and execution, since it requires spatial and temporal control. The investigation of music rehabilitation game techniques, that constitutes the basis of the game design, can be found in the previous work [126].

A side-scrolling game is proposed: the player shall dodge some obstacles, whose presence is anticipated by the rhythmic structure of the backing music, i.e., the piece of music that is played in the background of the game. Thus, the child must understand the temporal pattern of the backing music in order to predict and dodge the obstacles from above or from below.

The player controls the position of a virtual character by touching commands on the screen. The character can jump or slide and is constrained to move in the horizontal direction with a fixed speed. The action is viewed from a side-view camera angle. The difficulty depends on two



Figure 9.1: Gameplay screenshot of the game RhythmRun. The player shall execute movements according to the rhythmical structure of the piece of music, which is made evident also by some obstacles to be dodged.

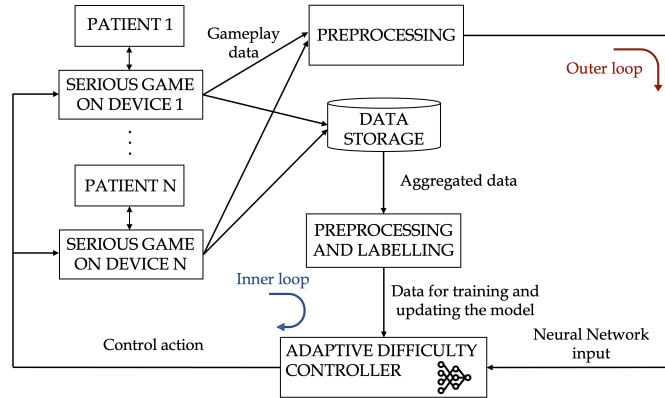


Figure 9.2: Neural Network architecture. The outer loop (denoted in red) represents the online autonomous difficulty adaptation, whereas the inner loop (denoted in blue) represents the training and updating of the Neural Network Classifier model.

parameters, namely the speed and complexity of the music piece, and consequently on the number and the interchange of the obstacles. Fig. 9.1 shows a gameplay screenshot.

In the following section 9.3.1, it is introduced the proposed system for autonomous difficulty adaptation, that automatically modifies the game parameters in real-time based on the player's cognitive capabilities, through a Neural Network Classifier, denoted in the following as NNC [126]. Section 9.3.2 describes the BC implementation used for data augmentation, through a second distinct Neural Network for Behavioural Cloning, denoted in the following as NNBC.

9.3.1 Adaptive Serious Game via Neural Networks

The objective of the rehabilitation game is to keep the player engaged throughout the entire game duration, providing an enjoyable and yet challenging experience. Indeed, the creation of an adequate level of challenge is not a trivial task when the game is proposed to players having varying capabilities. This is a notably important aspect to consider in a clinical setting where the players are children of various ages and bear different specific pathologies, thus having different cognitive capabilities. To tackle such problem, this chapter proposes to relay on a NNC architecture to automatically adapt the game difficulty level.

Fig. 9.2 shows a high-level block scheme of the proposed NNC architecture. The outer loop (denoted in red) has the task of adapting online the difficulty level of the game, whereas the inner loop (denoted in blue) represents the training and updating of the NNC model.

Patients play the game on mobile devices (i.e., tablets), that upload the gameplay data via

Internet connection on a cloud server in Google Firebase (if a device is not connected, data is stored locally and then uploaded once the connection is restored); in the figure, the data collection is represented by the Data Storage block. On the data that is gathered and aggregated from all patients, it is performed Behavioural Cloning for augmenting the dataset dimension, through the second distinct NNBC specifically built for this purpose. Data processing, executed by the Preprocessing and Labelling block, is performed in Google Colab: the augmented dataset is labelled, used for generating statistics, and preprocessed so as to derive the suitable inputs to the NNC for training, by selecting the parameters about patients and about the game that are relevant for the autonomous adaptation, as further discussed in the following. Such data preprocessing is analogously executed in the Preprocessing block of the outer loop. The NNC is updated regularly at successive iterations in the same way. The NNC is structured so as to have 1 input layer, 3 hidden layers, of 64, 128 and 64 neurons, and an output layer with a single sigmoid neuron. The ReLu activation function was selected, with Adam as optimizer, Cross-entropy loss function and a learning rate equal to 0.001.

As shown in the outer loop of Fig. 9.2, the NNC model autonomously updates online the difficulty level of the game: on the basis of past gameplay data from a specific player, it returns as output the estimated optimal choice for the values of the difficulty parameters of the game for that player. The output of the NNC is a control action indicating if the values of the difficulty parameters have to be set in order to maintain, increase or reduce the current difficulty level.

In the first training phase for data gathering, only the inner loop of Fig. 9.2 is activated, whereas the control actions that shall derive from the outer loop are replaced by classical and simple threshold-based rules that are conventionally used in video games. In this way, the outer loop is closed and data for the training of the NNC is gathered. However, since a large amount of data is needed to train the NNC, a data augmentation method based on BC is applied to perform the data gathering task (i.e., the inner loop of Fig. 9.2, as described in section 9.3.2).

Pediatric doctors and psychologists are in charge of providing the logic to perform data labeling, i.e., to specify the most appropriate control action to be executed after the initial data collection, so that the algorithm shall adapt to such specifications and learn to perform choices following a reasoning as close as possible to the one of doctors and psychologists. In particular, in order to determine the parameters about patients and about the game that are relevant inputs for the NN, the Griffiths Scales of Mental Development were used to identify the neural correlates of game activities [129], as reported in TABLE 9.1. Moreover, the NNC is built so as to return as output a control action in the action space $A = \{downgrade, maintain, upgrade\}$ which shall adapt the difficulty of the game, that depends on the speed and complexity of the backing piece of music and on the number and interchange of the obstacles.

In summary, for each played match, the NNC estimates a function that takes as input the past gameplay data of the child and returns as output the control action that changes the difficulty of the next match. The aim of the control action is to set the difficulty parameters of the next match so that the difficulty of the next match is the most suitable to the actual patient's level. In the context of rehabilitation, the medical outcomes are twofold: first, if the game difficulty fits the current cognitive capabilities of the patient, he/she is encouraged to play and train; secondly, thanks to the training, the patient improves future performances on the game, leading to improvements in cognitive capabilities.

Table 9.1: Inputs to the NN Classifier

GAME	GRIFFITHS SCALES	COGNITIVE FACULTIES	NN INPUTS
<i>RhythmRun</i>	Hand-eye Coordination, Large Motor Skills.	Motor Coordination, Rhythm.	Age, Difficulty level of last match, Hit obstacles, Error mean, Level repetitions.

9.3.2 Behavioural Cloning Implementation

A challenging issue of the framework proposed in section 9.3.1 relies in the difficulty of obtaining a sufficiently large training dataset, a common problem in the context of neurorehabilitation [112]. To tackle this problem of data scarcity, BC is applied for data augmentation.

Fig. 9.3 depicts the BC framework. BC can be defined as a special case of Reinforcement Learning (RL) [117], a subclass of Machine Learning where an agent interacts with an environment. The RL agent observes the current state s_t of the environment, performs an action a_t and observes the new state s_{t+1} . The transition from a state to the next one depends on the action. In RL, the agent also receives a numerical reward signal r_t associated to the performed action and the next state. Differently, reward signals are not available to BC agents, and r is ignored [130].

The BC problem is formulated as follows: given a dataset D , originated from human gameplay data and consisting of tuples (s, a) , learn an imitation model $M = P(a|s)$, i.e., a conditional probability distribution, where $P(a|s)$ denotes the probability that a human performs action a when in state s . $M = P(a|s)$ is modelled using the NNBC that minimizes the cross-entropy between predictions and real values of the dataset. After learning the model, the BC agent plays the game by selecting the predicted action $a \sim (a|s)$ for a given state s .

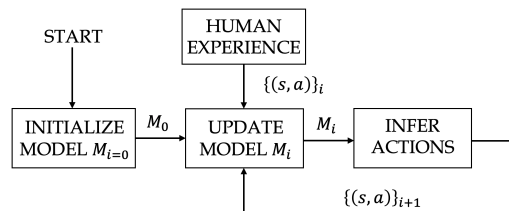
In the proposed *RhythmRun* game, the states and actions are defined as follows:

$$s = (\text{Obstacle Distance}, \text{Obstacle Type}), \quad (9.1)$$

$$a = (\text{Idle}, \text{Jump}, \text{Slide}), \quad (9.2)$$

where *Obstacle Distance* is the time distance from the next obstacle and *Obstacle Type* indicates whether the obstacle shall be dodged by jumping or sliding. State sampling is executed at a constant cadence, i.e., in this implementation, every 100 milliseconds.

The NNBC is structured so as to have 1 input layer, 3 hidden layers, with dimensions 256, 528, 256, and 1 output layer. The ReLu activation function was selected, with SGD as optimizer,

**Figure 9.3:** Behavioural Cloning framework.

Cross-entropy loss function, learning rate equal to 0.005 and momentum equal to 0.9.

9.4 Experimental Validation

A 1.0 beta-version of the app *MusicalMente* [126] with the first 10 levels of the game *RhythmRun* was developed in Unity.

As already illustrated, at first the game is programmed so as to adapt difficulty by following conventional static and simple threshold-based rules. Specifically, the difficulty of the next match is selected base on the score of the last match: the difficulty is chosen to be downgraded, maintained or upgraded if the game score, expressed in percentage of avoided obstacles, is lower than or equal to 30%, between 30% and 60% or larger than or equal to 60%, respectively.

For the scope of the present proof of concept, the app was provided to a test group of 27 healthy children aged between 5 and 14, yielding a gameplay dataset of 307 played matches. Larger sample sizes will be involved in future works. The dataset was handled as follows: 10% of the original dataset was allocated to testing; on the remaining 90% part, data augmentation via BC was applied to increase the dataset dimension; 80% of the resulting augmented dataset was used for training, whereas the remaining 20% for validation.

The application of BC yielded a larger number of played matches while preserving the features of the original dataset, as it is shown in TABLE 9.2, that also reports in the last column the statistics that will be introduced in the following section 9.4.

Fig. 9.4 shows a comparison of the average error percentage per level in the two cases of the real dataset and of the augmented dataset.

Note that, as it was desirable, the error percentages are very similar - in some levels the average errors percentage is higher in the case of the real dataset, whereas in other levels the opposite result is observed.

In order to evaluate the effectiveness of data augmentation via BC, it was executed a comparison on performances in terms of validation accuracy between the cases with and without data augmentation. As shown in Fig. 9.5, the validation accuracy exhibits a faster transient with data augmentation and reaches an higher value of 0.94, while in the other case the accuracy stops at value 0.87.

Finally, the performances of the proposed system were evaluated on the test set: Fig. 9.6 shows the classification performances when the game difficulty is adapted following classical threshold-based rules while Fig. 9.7 shows to the performances of the proposed system, exhibiting significant

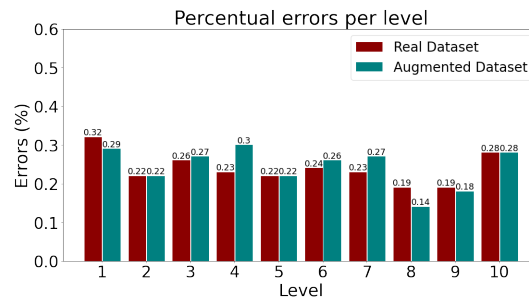


Figure 9.4: Comparison of the average error percentage per level in the case of the real dataset, and in the case of the augmented dataset.

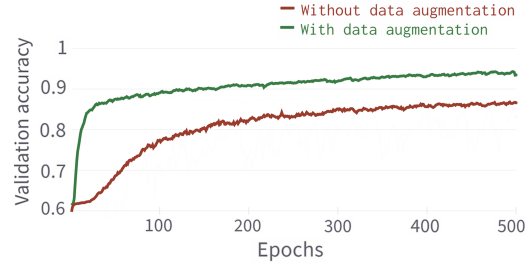


Figure 9.5: The lower red plot shows the validation accuracy obtained with the NNC trained without data augmentation; the upper green plot shows the validation accuracy obtained with the NNC trained on the augmented dataset.

improvements.

An additional benchmark comparison was performed using Support Vector Machines (SVM) [131], a model for classification that takes data as input and returns as output an hyperplane separating data into classes. An SVM model with polynomial kernel of third-degree was trained on the augmented dataset. This method does outperform the difficulty adaptation via simple threshold-based rules, but still yields lower performances with respect to the proposed method, as it is summarized in TABLE 9.3.

Finally, a 2.0 beta-version of the app was implemented, where the game is autonomously adapted online by exploiting the trained NNC. For the scope of this proof of concept, a different test group of 8 healthy children played the game and provided a new gameplay dataset, whose main statistics are reported in the last column of TABLE 9.2. Larger sample sizes will be involved in future works.

Considering this second group of players, Fig. 9.8 compares their performances in the case where the game difficulty level was adapted by the NNC with respect to the case where difficulty would have been adapted by simple threshold-based rules: a different action with respect to the threshold-based reasoning was selected in a percentage of cases equal to 27%.

Note that, even if in the first levels the performances are similar or even worse, in the last levels the children who played the game version that is adapted by the NNC achieve better performances with the respect to the children who played the game version that is adapted by simple threshold-based rules. Therefore, the proposed method has shown to provide a difficulty adjustment which enables children to achieve higher scores over time and, consequently, a more effective cognitive training.

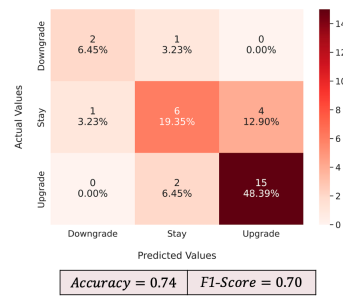


Figure 9.6: Classification performances when the game difficulty is adapted following classical threshold-based rules.

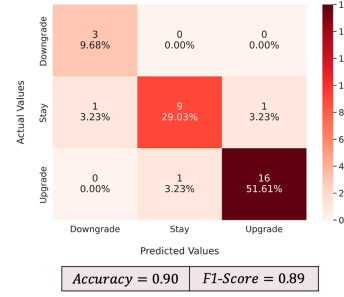


Figure 9.7: Classification performances when the game difficulty is adapted with the proposed system.

9.5 Conclusions

This chapter presented a Serious Game with a system for deploying autonomous difficulty adaptation by means of Neural Networks trained over datasets that are obtained via Behavioural Cloning data augmentation.

The outcomes provided by this work are outlined in the following list:

- A Serious Game aimed at supporting pediatric neurorehabilitation via musical rehabilitative techniques was developed; the proposed rehabilitation game was originally conceived based on findings in pediatric academic literature;
- A NN-based approach was conceived to improve the effectiveness of the rehabilitation game by adapting the difficulty level of the game to the current child's cognitive conditions; the effectiveness of the NNC approach was validated against standard threshold-based strategies and a simpler Artificial Intelligence method such as Support Vector Machines;
- Behavioural Cloning for data augmentation was implemented in order to enable the NNC training under data paucity (a severe problem in the considered children rehabilitation task); to our knowledge, this is first the application of Behavioural Cloning techniques in the context of Serious Games.

The key research objectives of this study have been accomplished:

- The proposed Behavioural Cloning approach for data augmentation has proven to yield higher classification performances with respect to the case with the non-augmented dataset.

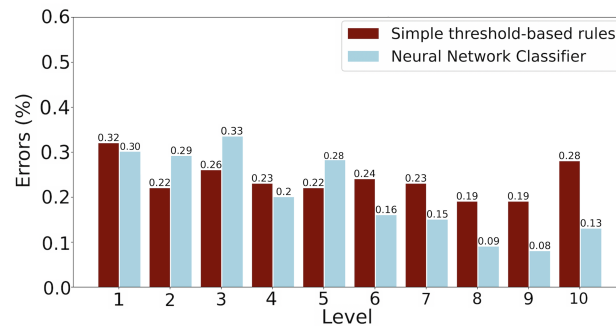


Figure 9.8: Comparison of the average error percentage in the case of adaptation via online NN classification, with respect to the case of adaptation via threshold-based rules.

- The proposed Neural Network Classification method for autonomous difficulty adaptation has proven to yield higher children performances, and thus stronger cognitive improvements, with respect to the case of difficulty adaptation via conventional and simple threshold-based rules, and also with respect to SVM.

Future works shall involve:

- Actual testing and validation of the proposed system in a pediatric hospital, with larger sample sizes for the experimental validation. As a matter of fact, the project has already been presented to the pertinent medical staff of a pediatric hospital, who have expressed strong interest in the topic and availability to test its effectiveness on patients.
- Experimental validation of additional Serious Games that train different cognitive faculties, such as attention, verbal fluency, empathy and collaboration [126].
- Application of Federated Learning, that enables to share useful data between hospitals without the need of sharing sensitive information protected by privacy [132, 133].
- Extension of the app usage with digital music instruments, in order to make the playing experience even more engaging. Moreover, activities involving collaboration of multiple children shall be included.
- Study of possible applications of the system in support of neurorehabilitation for different conditions.

Table 9.2: Comparison of dataset statistics

	Standard Serious Game		Adaptive Serious Game
	Before BC	After BC	
Nr. of players	27	127	8
Average age of players	9.3	9.1	8.6
Total nr. of matches	307	1728	111
Average error percentage	0.24	0.26	0.28
Error percentage st. dev.	0.28	0.17	0.30

Table 9.3: Comparison of classification performances

	Threshold based rules	SVM	Proposed system
Accuracy	0.74	0.76	0.90
F-1 Score	0.70	0.72	0.89

General Conclusions

This dissertation advances the synthesis and analysis of **nonlinear, safety-aware controllers** through a unifying **energy/passivity** perspective and computation-ready methods. First, we compare **energy-based** and **exponential** Control Barrier Functions (CBFs), showing how barrier constructions reshape energy flows, shaping, dissipation, interconnection, and thereby impact invariance, transients, and robustness. The analysis spans theory, software (3R planar, 7-DoF manipulator), and hardware, clarifying when each CBF offers predictable transients or introduces sensitivity to modeling and tuning. Second, we address a core bottleneck of IDA-PBC by solving the kinetic-energy assignment **PDE** with **PINNs** via a tailored loss. Under a simple potential structure, solving only the kinetic PDE up to a small residual and injecting damping suffices for **asymptotic stability**, extending IDA-PBC to analytically intractable cases while preserving a clear energetic interpretation. Third, we introduce a **model-based learning** method to synthesize **passivity-preserving CBFs** under **input saturation**, co-engineering safety, performance, and energy balance on cart-pole and 2R robots, where classical safety filters may erode stability margins.

Limitations and outlook. The methods rely on models accurate enough to expose energetic structure; broader experiments (e.g., contact-rich tasks) and **formal bounds** linking PINN residuals to stability margins remain open. Promising directions include: (i) robust/stochastic energetic designs for disturbances and uncertainty; (ii) certificate-aware learning (Lyapunov/CBF conditions embedded in training); (iii) extensions to hybrid dynamics and richer actuator constraints; (iv) structure-exploiting solvers for scalable PDE training and real-time filtering; (v) compositional safety/stability for networks of agents; and (vi) energy-consistent identification to adapt models without violating passivity or safety.

Concluding remark. By unifying energy shaping, barrier-based safety, and learning-assisted computation, the dissertation offers a coherent route to **safe, stable, and interpretable** nonlinear control that scales from canonical benchmarks to realistic robotic platforms.

Appendix A

Appendix

A.1 Asymptotic Stability and Small Perturbations

It is recalled the following Lemma on Asymptotic Stability and Small Perturbations [8].

Lemma 6. *Suppose the equilibrium $x = 0$ of system:*

$$\dot{x} = f(x), \quad (\text{A.1})$$

in which $x \in \mathbb{R}^n$, $f(0) = 0$ and $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ locally Lipschitz, is locally asymptotically stable. Suppose $g(x, t)$ is piecewise continuous in t and satisfies a Lipschitz condition:

$$\|g(x', t) - g(x'', t)\| \leq L\|x' - x''\|, \quad (\text{A.2})$$

$\forall t \geq 0$ and $\forall x', x''$ in some neighbourhood U of $x = 0$. Then, given $\epsilon > 0$, there exists $\delta_1 > 0$, $\delta_2 > 0$ such that, if:

$$\begin{aligned} \|x_0\| &\leq \delta_1, \\ \|g(x, t)\| &\leq \delta_2 \forall \|x\| \leq \epsilon, \forall t \geq 0, \end{aligned} \quad (\text{A.3})$$

the solution $x(t)$ of the perturbed system:

$$\dot{x} = f(x) + g(x, t), \quad (\text{A.4})$$

satisfying $x(0) = x_0$ is such that:

$$\|x(t)\| \leq \epsilon, \forall t \geq 0. \quad (\text{A.5})$$

A.2 Proofs of injected power equation for each analyzed CBF

A.2.1 Power injection: Proof of (5.27)

To evaluate the power equation (5.14) corresponding to (5.25), the term $\frac{\partial h}{\partial \dot{q}}$ must be computed as:

$$\begin{aligned}\frac{\partial h}{\partial \dot{q}} &= \frac{\partial}{\partial \dot{q}} \left(\frac{K_e(q, \dot{q})}{1 + K_e(q, \dot{q})} \right) = \\ &= \left[\frac{\dot{q}^\top M(q)}{1 + K_e} - \frac{\dot{q}^\top M(q) K_e}{(1 + K_e)^2} \right] = \frac{\dot{q}^\top M(q)}{(1 + K_e)^2}\end{aligned}$$

which, when substituted into expression (5.14), yields a stability preserving property:

$$P_{h,\alpha} = -1_{\Psi_R} \frac{\dot{q}^\top M^{-1}(q) \frac{M(q)}{(1+K_e)^2} \dot{q}}{\|M^{-1} \frac{M(q)}{(1+K_e)^2} \dot{q}\|^2} \Psi_R. \quad (\text{A.6})$$

$$= -1_{\Psi}(1 + K_e)^2 \Psi \leq 0 \quad (\text{A.7})$$

The equation for (5.27) can be proven by computing the expression for $\dot{h}$:

$$\begin{aligned}\dot{h}(q, \dot{q}, u) &= \\ &= -\frac{1}{\phi(q)^2} \left(\frac{\partial \phi}{\partial q} \dot{q} \right) + \left(\frac{\dot{K}_e}{1 + K_e} \right) - \left(\frac{K_e \dot{K}_e}{(1 + K_e)^2} \right) = \\ &= -\frac{1}{\phi(q)^2} \left(\frac{\partial \phi}{\partial q} \dot{q} \right) + \left(\frac{\dot{K}_e}{(1 + K_e)^2} \right) = \\ &= -\frac{1}{\phi(q)^2} \left(\frac{\partial \phi}{\partial q} \dot{q} \right) + \left(\frac{u - g(q)}{(1 + K_e)^2} \right)\end{aligned}$$

A.2.2 Proof of Eq. (5.30)

$$\begin{aligned}\Psi_Z = \dot{h} + \alpha h &= \\ &= \frac{d}{dt} \left[\phi(q) + \gamma \frac{\partial \phi}{\partial q} \dot{q} \right] + \alpha \left[\phi(q) + \gamma \frac{\partial \phi}{\partial q} \dot{q} \right] = \\ &= \frac{\partial \phi}{\partial q} \dot{q} + \gamma \frac{\partial \phi}{\partial q} \ddot{q} + \gamma \dot{q}^\top \frac{\partial^2 \phi}{\partial q^2} \dot{q} + \alpha \left[\phi(q) + \gamma \frac{\partial \phi}{\partial q} \dot{q} \right] = \\ &= \alpha \phi(q) + (1 + \gamma \alpha) \frac{\partial \phi}{\partial q} \dot{q} + \gamma \frac{\partial \phi}{\partial q} \ddot{q} + \gamma \dot{q}^\top \frac{\partial^2 \phi}{\partial q^2} \dot{q}\end{aligned}$$

Bibliography

- [1] Arturo Maiani, Federico Califano, Lorenzo Govoni, and Antonio Pietrabissa. Comparative analysis of energy transfers and performance in safety-critical control using control barrier functions. Under review for the *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.
- [2] Antonio Di Paola, Arturo Maiani, and Danilo Menegatti. A physics-informed neural networks based method for interconnection and damping assignment passivity-based control. Accepted for publication in the *IEEE Conference on Decision and Control (CDC)*, 2025.
- [3] Arturo Maiani, Federico Baldisseri, and Antonio Pietrabissa. A machine learning approach to passivity-preserving safety-critical control. Submitted to the *Journal of the Franklin Institute*, 2025.
- [4] Danilo F. Baldisseri, Menegatti, Arturo Maiani, Alessandro Giuseppe, and Antonio Pietrabissa. Neural mpc for safety-critical biological systems: an application to tumor-immune cancer dynamics. Submitted to *International Journal of Control*, 2025.
- [5] Federico Baldisseri, Edoardo Montecchiani, Arturo Maiani, Danilo Menegatti, Alessandro Giuseppe, Antonio Pietrabissa, Vincenzo Fogliati, and Francesco Delli Priscoli. Behavioural cloning for serious games in support of pediatric neurorehabilitation. In *2023 31st Mediterranean Conference on Control and Automation (MED)*, pages 487–492, 2023. doi: 10.1109/MED59994.2023.10185734.
- [6] Federico Baldisseri, Arturo Maiani, Edoardo Montecchiani, Francesco Delli Priscoli, Alessandro Giuseppe, Danilo Menegatti, and Vincenzo Fogliati. An integrated music and artificial intelligence system in support of pediatric neurorehabilitation. *Healthcare*, 10(10), 2022. ISSN 2227-9032. doi: 10.3390/healthcare10102014. URL <https://www.mdpi.com/2227-9032/10/10/2014>.
- [7] Hassan K Khalil and Jessy W Grizzle. *Nonlinear systems*, volume 3. Prentice hall Upper Saddle River, NJ, 2002.
- [8] Alberto Isidori. *Nonlinear control systems: an introduction*. Springer, 1985.
- [9] AD Ames, X Xu, JW Grizzle, and P Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2017.

- [10] AD Ames, S Coogan, G Notomista, P Fernandes, and P Tabuada. Control barrier functions: Theory and applications. In *2019 18th European Control Conference (ECC)*, pages 3420–3431. IEEE, 2019.
- [11] Aaron D. Ames, Jessy W. Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs with application to adaptive cruise control. In *53rd IEEE Conference on Decision and Control*, pages 6271–6278, 2014. doi: 10.1109/CDC.2014.7040372.
- [12] Wei Xiao and Calin Belta. Control barrier functions for systems with high relative degree. In *2019 IEEE 58th conference on decision and control (CDC)*, pages 474–479. IEEE, 2019.
- [13] Wei Xiao and Calin Belta. High-order control barrier functions. *IEEE Transactions on Automatic Control*, 67(7):3655–3662, 2022. doi: 10.1109/TAC.2021.3105491.
- [14] Quan Nguyen and Koushil Sreenath. Exponential control barrier functions for enforcing high relative-degree safety-critical constraints. In *2016 American Control Conference (ACC)*, pages 322–328, 2016. doi: 10.1109/ACC.2016.7524935.
- [15] Muhammad Zakiyullah Romdlony and Bayu Jayawardhana. Stabilization with guaranteed safety using control lyapunov–barrier function. *Automatica*, 66:39–47, 2016. ISSN 0005-1098. doi: <https://doi.org/10.1016/j.automatica.2015.12.011>.
- [16] Boqian Li, Shiping Wen, Zheng Yan, Guanghui Wen, and Tingwen Huang. A survey on the control lyapunov function and control barrier function for nonlinear-affine control systems. *IEEE/CAA Journal of Automatica Sinica*, 10(3):584–602, 2023.
- [17] Mrdjan Jankovic. Robust control barrier functions for constrained stabilization of nonlinear systems. *Automatica*, 96:359–367, 2018. ISSN 0005-1098. doi: <https://doi.org/10.1016/j.automatica.2018.07.004>.
- [18] Changliu Liu and Masayoshi Tomizuka. Control in a safe set: Addressing safety in human-robot interactions. In *ASME 2014 Dynamic Systems and Control Conference, DSCC 2014*, volume 3, 11 2014. doi: 10.1115/DSCC2014-6048.
- [19] Randy A. Freeman and Petar V. Kokotovic. *Robust nonlinear control design: state-space and Lyapunov techniques*. Birkhauser Boston Inc., USA, 1996. ISBN 0817639306.
- [20] X Xu, P Tabuada, JW Grizzle, and AD Ames. Robustness of control barrier functions for safety critical control. In *IFAC-PapersOnLine*, volume 48, pages 54–61. IFAC, 2015.
- [21] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated, 2010. ISBN 1849966346.
- [22] Mark W Spong, Seth Andrew Hutchinson, and Mathukumalli Vidyasagar. Robot modeling and control. *IEEE Control Systems*, 26(6):113–115, 2006.
- [23] Arjan Schaft. *L2-Gain and Passivity Techniques in Nonlinear Control*. 01 2017. ISBN 978-3-319-49991-8. doi: 10.1007/978-3-319-49992-5.
- [24] Arjan Van der Schaft. *L2-gain and passivity techniques in nonlinear control*. Springer, 2000.

- [25] Arjan J van der Schaft. Port-hamiltonian systems: network modeling and control of nonlinear physical systems. In *Advanced dynamics and control of structures and machines*, pages 127–167. Springer, 2004.
- [26] V Duindam, A Macchelli, S Stramigioli, and H Bruyninckx. *Modeling and Control of Complex Physical Systems: The Port-Hamiltonian Approach*. Springer, 2009.
- [27] Romeo Ortega, Arjan Van Der Schaft, Fernando Castanos, and Alessandro Astolfi. Control by interconnection and standard passivity-based control of port-hamiltonian systems. *IEEE Transactions on Automatic control*, 53(11):2527–2542, 2008.
- [28] Romeo Ortega, Mark W Spong, Fabio Gómez-Estern, and Guido Blankenstein. Stabilization of a class of underactuated mechanical systems via interconnection and damping assignment. *IEEE transactions on automatic control*, 47(8):1218–1233, 2002.
- [29] Jose Angel Acosta, Romeo Ortega, Alessandro Astolfi, and Arun D Mahindrakar. Interconnection and damping assignment passivity-based control of mechanical systems with underactuation degree one. *IEEE Transactions on Automatic Control*, 50(12):1936–1955, 2005.
- [30] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986. doi: 10.1038/323533a0.
- [31] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [32] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4:251–257, 1991.
- [33] A Paszke. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.
- [34] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951. doi: 10.1214/aoms/1177729586.
- [35] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. URL <https://arxiv.org/abs/1412.6980>.
- [36] KP Waber, AD Ames, and P Tabuada. Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems. *IEEE Control Systems*, 43(5):137–177, 2023.
- [37] F Ferraguti, A Villa, A De Luca, and G Ferretti. Safety and efficiency in robotics: The control barrier functions approach. *IEEE Robotics and Automation Magazine*, 29(3):139–151, 2020.
- [38] A Singletary, S Kolathaya, and AD Ames. Safety-critical kinematic control of robotic systems. In *Proceedings of the American Control Conference*, volume 2021-May, pages 14–19. IEEE, 2021.
- [39] Shishir Kolathaya. Energy based control barrier functions for robotic systems, August 2020. TechRxiv.

- [40] Andrew Singletary, Karl Klingebiel, Joseph Bourne, Andrew Browning, Phil Tokumaru, and Aaron Ames. Comparative analysis of control barrier functions and artificial potential fields for obstacle avoidance. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8129–8136. IEEE, 2021.
- [41] Quan Nguyen and Koushil Sreenath. Exponential control barrier functions for enforcing high relative-degree safety-critical constraints. In *2016 American Control Conference (ACC)*, pages 322–328. IEEE, 2016.
- [42] Xiao Tan and Dimos V Dimarogonas. Distributed implementation of control barrier functions for multi-agent systems. *IEEE Control Systems Letters*, 6:1879–1884, 2021.
- [43] Beatrice Capelli, Cristian Secchi, and Lorenzo Sabattini. Passivity and control barrier functions: Optimizing the use of energy. *IEEE Robotics and Automation Letters*, 7(2):1356–1363, 2022. doi: 10.1109/LRA.2021.3139951.
- [44] R Califano, F Rashad, C Secchi, and S Stramigioli. The use of energy tanks for robotics systems. In *Soft Robotics*, pages 172–188. Springer, 2018.
- [45] Gennaro Notomista, Xiaoyi Cai, Junya Yamauchi, and Magnus Egerstedt. Passivity-based decentralized control of multi-robot systems with delays using control barrier functions. In *2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, pages 231–237, 2019. doi: 10.1109/MRS.2019.8901085.
- [46] Hannah M. Sweatland, Axton Isaly, Emily J. Griffis, and Warren E. Dixon. Optimization-based controllers for passivity and safety constraints. *IEEE Transactions on Automatic Control*, pages 1–8, 2024. doi: 10.1109/TAC.2024.3513894.
- [47] Federico Califano. Passivity-preserving safety-critical control using control barrier functions. *IEEE Control Systems Letters*, 7:1742–1747, 2023. doi: 10.1109/LCSYS.2023.3280082.
- [48] Qingkai Meng, Andreas Kasis, and Marios M. Polycarpou. Passivity preserving safety-critical control of switched systems. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pages 8596–8601, 2024. doi: 10.1109/CDC56724.2024.10886724.
- [49] Wenceslao Shaw Cortez, Christos K. Verginis, and Dimos V. Dimarogonas. Safe, passive control for mechanical systems with application to physical human-robot interactions. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3836–3842, 2021. doi: 10.1109/ICRA48506.2021.9561981.
- [50] Federico Califano, Riccardo Zanella, Alessandro Macchelli, and Stefano Stramigioli. The effect of control barrier functions on energy transfers in controlled physical systems, 2024.
- [51] Simin Liu, Changliu Liu, and John M. Dolan. Safe control under input limits with neural control barrier functions. In *Proceedings of (CoRL) Conference on Robot Learning*, pages 1–11, December 2022.
- [52] Günter Schreiber, Andreas Stemmer, and Rainer Bischoff. The fast research interface for the kuka lightweight robot. In *IEEE workshop on innovative robot control architectures for*

- demanding (Research) applications how to modify and enhance commercial controllers (ICRA 2010)*, pages 15–21. Citeseer, 2010.
- [53] Tamas G Molnar and Aaron D Ames. Composing control barrier functions for complex safety specifications. *IEEE Control Systems Letters*, 7:3615–3620, 2023.
- [54] Nikhil Chopra, Masayuki Fujita, Romeo Ortega, and Mark W Spong. Passivity-based control of robots: Theory and examples from the literature. *IEEE Control Systems Magazine*, 42(2): 63–73, 2022.
- [55] Nikhil Chopra and Mark W Spong. Passivity-based control of multi-agent systems. *Advances in robot control: from everyday physics to human-like movements*, pages 107–134, 2006.
- [56] F. Gomez-Estern, R. Ortega, F.R. Rubio, and J. Aracil. Stabilization of a class of underactuated mechanical systems via total energy shaping. In *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No.01CH37228)*, volume 2, pages 1137–1143 vol.2, 2001. doi: 10.1109/CDC.2001.981038.
- [57] Hamidreza Eivazi, Mojtaba Tahani, Philipp Schlatter, and Ricardo Vinuesa. Physics-informed neural networks for solving reynolds-averaged navier–stokes equations. *Physics of Fluids*, 34(7), 2022.
- [58] M Kang and NJ Jameson. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Prognostics and Health Management of Electronics: Fundamentals, Machine Learning, and the Internet of Things*, 2018.
- [59] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.
- [60] Ehsan Kharazmi, Zhongqiang Zhang, and George Em Karniadakis. Variational physics-informed neural networks for solving partial differential equations. *arXiv preprint arXiv:1912.00873*, 2019.
- [61] Wankun Sirichotiyakul and Aykut C Satıcı. Data-driven passivity-based control of underactuated mechanical systems via interconnection and damping assignment. *International Journal of Control*, 96(6):1448–1456, 2023.
- [62] R. Ortega, A.J. Van Der Schaft, I. Mareels, and B. Maschke. Putting energy back in control. *IEEE Control Systems Magazine*, 21(2):18–33, 2001. doi: 10.1109/37.915398.
- [63] Lorenzo P Pagani, George E Apostolakis, and Pavel Hejzlar. The impact of uncertainties on the performance of passive systems. *Nuclear Technology*, 149(2):129–140, 2005.
- [64] Maria E Guerrero, Diego A Mercado, Rogelio Lozano, and Carlos D García. Ida-pbc methodology for a quadrotor uav transporting a cable-suspended payload. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 470–476. IEEE, 2015.

- [65] Chunxiao Huang and Lijun Long. Safety-critical multi-tasks control for switched systems via several barrier functions. *Journal of the Franklin Institute*, 361(9):106906, 2024. ISSN 0016-0032. doi: <https://doi.org/10.1016/j.jfranklin.2024.106906>. URL <https://www.sciencedirect.com/science/article/pii/S0016003224003272>.
- [66] Xinyu Zhang, Zheng H. Zhu, Siyu Xie, Hui Gao, and Gun Li. Barrier function-based prescribed-performance adaptive attitude tracking control for spacecraft with uncertainties. *Journal of the Franklin Institute*, 360(12):8075–8095, 2023. ISSN 0016-0032. doi: <https://doi.org/10.1016/j.jfranklin.2023.06.022>. URL <https://www.sciencedirect.com/science/article/pii/S0016003223003800>.
- [67] Chenchen Fan, Kai-Fung Chu, Xiaomei Wang, Ka-Wai Kwok, and Fumiya Iida. State transition learning with limited data for safe control of switched nonlinear systems. *Neural Networks*, 180:106695, 2024.
- [68] Cheng Hu and Zhe Wu. Model predictive control of switched nonlinear systems using online machine learning. *Chemical Engineering Research and Design*, 209:221–236, 2024.
- [69] Xiaofeng Li, Lu Dong, Lei Xue, and Changyin Sun. Hybrid reinforcement learning for optimal control of non-linear switching system. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):9161–9170, 2022.
- [70] Haoran Zhang, Chunhui Zhao, and Jinliang Ding. Robust safe reinforcement learning control of unknown continuous-time nonlinear systems with state constraints and disturbances. *Journal of Process Control*, 128:103028, 2023.
- [71] Michael Green and David JN Limebeer. *Linear robust control*. Courier Corporation, 2012.
- [72] Ahmad Taher Azar, Fernando E Serrano, Ibrahim A Hameed, Nashwa Ahmad Kamal, and Sundarapandian Vaidyanathan. Robust h-infinity decentralized control for industrial cooperative robots. In *International Conference on Advanced Intelligent Systems and Informatics*, pages 254–265. Springer, 2019.
- [73] Daniel Larby and Fulvio Forni. A passivity preserving h-infinity synthesis technique for robot control. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 1416–1422. IEEE, 2022.
- [74] Javier Serrano-Delgado, Santiago Cobrecas, Emilio J Bueno, and Mario Rizo. Passivity-based fixed-order h-infinity controller design for grid-forming vscs. In *2021 IEEE Applied Power Electronics Conference and Exposition (APEC)*, pages 2459–2466. IEEE, 2021.
- [75] Paul Schwerdtner and Matthias Voigt. Fixed-order h-infinity controller design for port-hamiltonian systems. *Automatica*, 152:110918, 2023.
- [76] Beatrice Capelli, Cristian Secchi, and Lorenzo Sabattini. Passivity and control barrier functions: Optimizing the use of energy. *IEEE Robotics and Automation Letters*, 7(2):1356–1363, 2022. doi: 10.1109/LRA.2021.3139951.

- [77] Gennaro Notomista, Xiaoyi Cai, and Magnus Egerstedt. Passivity-based decentralized control of multi-robot systems with delays using control barrier functions. *CoRR*, abs/1904.04801, 2019.
- [78] Charles Dawson, Zengyi Qin, Sicun Gao, and Chuchu Fan. Safe nonlinear control using robust neural lyapunov-barrier functions, 2021.
- [79] Ya-Chien Chang, Nima Roohi, and Sicun Gao. Neural lyapunov control. *CoRR*, abs/2005.00611, 2020.
- [80] Wei Xiao, Tsun-Hsuan Wang, Ramin Hasani, Makram Chahine, Alexander Amini, Xiao Li, and Daniela Rus. Barriernet: Differentiable control barrier functions for learning of safe robot control. *IEEE Transactions on Robotics*, 39(3):2289–2307, 2023. doi: 10.1109/TRO.2023.3249564.
- [81] Aaron D. Ames, Xiangru Xu, Jessy W. Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2017. doi: 10.1109/TAC.2016.2638961.
- [82] Hassan K Khalil. *Nonlinear systems; 3rd ed.* Prentice-Hall, Upper Saddle River, NJ, 2002. The book can be consulted by contacting: PH-AID: Wallet, Lionel.
- [83] John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization. *CoRR*, abs/1502.05477, 2015. URL <http://arxiv.org/abs/1502.05477>.
- [84] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.
- [85] Danilo Menegatti, Alessandro Giuseppe, and Antonio Pietrabissa. Tractable data-driven model predictive control using one-step neural networks predictors. *IEEE Transactions on Automation Science and Engineering*, 22:6740–6751, 2025. ISSN 1558-3783. doi: 10.1109/TASE.2024.3453668.
- [86] Yujie Yang, Yuhang Zhang, Wenjun Zou, Jianyu Chen, Yuming Yin, and Shengbo Eben Li. Synthesizing control barrier functions with feasible region iteration for safe reinforcement learning. *IEEE Transactions on Automatic Control*, 69(4):2713–2720, 2024. doi: 10.1109/TAC.2023.3336263.
- [87] Carlos E Garcia, David M Prett, and Manfred Morari. Model predictive control: Theory and practice—a survey. *Automatica*, 25(3):335–348, 1989.
- [88] Sushma Parihar, Pritesh Shah, Ravi Sekhar, and Jui Lagoo. Model predictive control and its role in biomedical therapeutic automation: A brief review. *Applied System Innovation*, 5(6): 118, 2022.
- [89] Zhiwei Ji, Ke Yan, Wenyang Li, Haigen Hu, and Xiaoliang Zhu. Mathematical and computational modeling in complex biological systems. *BioMed research international*, 2017(1): 5958321, 2017.

- [90] Olalekan Ogunmolu, Xuejun Gu, Steve Jiang, and Nicholas Gans. Nonlinear systems identification using deep dynamic neural networks. *arXiv preprint arXiv:1610.01439*, 2016.
- [91] Andreas Draeger, Sebastian Engell, and Horst Ranke. Model predictive control using neural networks. *IEEE Control Systems Magazine*, 15(5):61–66, 1995.
- [92] VS Kodogiannis, Paulo JG Lisboa, and J Lucas. Neural network modelling and control for underwater vehicles. *Artificial Intelligence in Engineering*, 10(3):203–212, 1996.
- [93] Donald Soloway and Pamela J Haley. Neural generalized predictive control. In *Proceedings of the 1996 IEEE international symposium on intelligent control*, pages 277–282. IEEE, 1998.
- [94] *The Duffing Equation: Nonlinear Oscillators and their Behaviour*. Wiley, March 2011. ISBN 9780470977859. doi: 10.1002/9780470977859. URL <http://dx.doi.org/10.1002/9780470977859>.
- [95] Saeid Bahremand, Hoo Sang Ko, Ramin Balouchzadeh, H Felix Lee, Sarah Park, and Guim Kwon. Neural network-based model predictive control for type 1 diabetic rats on artificial pancreas system. *Medical & biological engineering & computing*, 57:177–191, 2019.
- [96] Wang Mengling, Yan Huaicheng, and Shi Hongbo. Neural network based model predictive control of spatially cellular uptake in tissue engineering. In *2017 36th Chinese Control Conference (CCC)*, pages 4610–4613. IEEE, 2017.
- [97] Wallace Tan Gian Yion and Zhe Wu. Robust machine learning modeling for predictive control using lipschitz-constrained neural networks. *ArXiv*, abs/2308.13721, 2023. doi: 10.48550/arXiv.2308.13721.
- [98] Guillermo Lorenzo, Syed Rakin Ahmed, David A Hormuth II, Brenna Vaughn, Jayashree Kalpathy-Cramer, Luis Solorio, Thomas E Yankeelov, and Hector Gomez. Patient-specific, mechanistic models of tumor growth incorporating artificial intelligence and big data. *Annual Review of Biomedical Engineering*, 26, 2024.
- [99] Sicheng Li, Shun Wang, and Xiufen Zou. Data-driven mathematical modeling and quantitative analysis of cell dynamics in the tumor microenvironment. *Computers & Mathematics with Applications*, 113:300–314, 2022.
- [100] Alex Viguerie, Malú Grave, Gabriel F Barros, Guillermo Lorenzo, Alessandro Reali, and Alvaro LGA Coutinho. Data-driven simulation of fisher–kolmogorov tumor growth models using dynamic mode decomposition. *Journal of Biomechanical Engineering*, 144(12):121001, 2022.
- [101] Maria Dassow, Seddik Djouadi, and Kaouther Moussa. Optimal control of a tumor-immune system with a modified stepanova cancer model. *IFAC-PapersOnLine*, 54(15):227–232, 2021.
- [102] Urszula Ledzewicz, Mohammad Naghnaeian, and Heinz Schättler. Optimal response to chemotherapy for a mathematical model of tumor–immune dynamics. *Journal of mathematical biology*, 64:557–577, 2012.

- [103] Andreas Draeger, Sebastian Engell, and Horst Ranke. Model predictive control using neural networks. *IEEE Control Systems Magazine*, 15(5):61–66, 1995.
- [104] Shila Alizadehghobadi. Centralized model predictive control with prediction horizon tuning for multi-vehicle platoon. November 2024. doi: 10.21203/rs.3.rs-5358250/v1. URL <http://dx.doi.org/10.21203/rs.3.rs-5358250/v1>.
- [105] H. Alsmeier, A. Savchenko, and R. Findeisen. Neural horizon model predictive control - increasing computational efficiency with neural networks. In *2024 American Control Conference (ACC)*, page 1646–1651. IEEE, July 2024. doi: 10.23919/acc60939.2024.10644452. URL <http://dx.doi.org/10.23919/ACC60939.2024.10644452>.
- [106] Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with relu activation function. *Annals of Statistics*, 48, 08 2017. doi: 10.1214/19-AOS1875.
- [107] Danilo Menegatti, Alessandro Giuseppi, and Antonio Pietrabissa. Tractable data-driven model predictive control using one-step neural networks predictors. *IEEE Transactions on Automation Science and Engineering*, 2024.
- [108] Kaouthar Moussa, Mirko Fiacchini, and Mazen Alamir. Robust optimal scheduling of combined chemo-and immunotherapy: Considerations on chemotherapy detrimental effects. In *2020 American Control Conference (ACC)*, pages 4252–4257. IEEE, 2020.
- [109] Ananth Ranganathan. The levenberg-marquardt algorithm. *Tutorial on LM algorithm*, 11(1): 101–110, 2004.
- [110] Inc. The MathWorks. *Optimization Toolbox: User’s Guide*. The MathWorks, Inc., Natick, Massachusetts, 2025. URL <https://www.mathworks.com/help/optim>. Version R2025a.
- [111] Richard E Kopp. Pontryagin maximum principle. In *Mathematics in Science and Engineering*, volume 5, pages 255–279. Elsevier, 1962.
- [112] S. Moon et al. Artificial neural networks in neurorehabilitation: A scoping review. *NeuroRehabilitation*, 46(3):259–269, 2020.
- [113] F. Sciancalepore et al. Computer-based cognitive training in children with primary brain tumours: A systematic review. *Cancers*, 14(16):3879, 2022.
- [114] C. S. M. Bain. *A Framework for Behavioural Cloning*. University of New South Wales, Sydney, Australia, 2001.
- [115] T. Susi, M. Johannesson, and P. Backlund. Serious games – an overview. Technical report, HS-IKI-TR, 2007.
- [116] M. Thaut, A. Leins, R. Rice, H. Argstatter, and G. Kenyon. Rhythmic auditory stimulation improves gait more than ndt training in near-ambulatory patients early post-stroke. *Neural Repair*, 21:455–459, 2017.
- [117] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2018.

- [118] V. Mnih et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529, 2015.
- [119] O. Vinyals, I. Babuschkin, W. M. Czarnecki, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.
- [120] J. Ho and S. Ermon. Generative adversarial imitation learning. *CoRR*, 1606(3476):1–8, 2016.
- [121] S. Fujimoto, E. Conti, M. Ghavamzadeh, and J. Pineau. Benchmarking batch deep reinforcement learning algorithms. *CoRR*, 1910(1708):1–8, 2019.
- [122] A. Y. Ng and S. J. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 1, 2000.
- [123] W. H. Guss, B. Houghton, N. Topin, P. Wang, C. Codel, M. Veloso, and R. Salakhutdinov. Minerl: A large-scale dataset of minecraft demonstrations. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, volume 1907, pages 4–6, 2019.
- [124] C. Renman. Creating human-like ai movement in games using imitation learning. Master’s thesis, KTH, School of Computer Science and Communication, 2017.
- [125] S. Ross, G. J. Gordon, and J. A. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. *CoRR*, 1011(686):1–8, 2011.
- [126] F. Baldisseri, A. Maiani, E. Montecchiani, F. Delli Priscoli, A. Giuseppi, D. Menegatti, and V. Fogliati. An integrated music and artificial intelligence system in support of pediatric neurorehabilitation. *Healthcare*, 10(2014):1–10, 2022.
- [127] M. H. Thaut, G. C. McIntosh, and R. R. Rice. Rhythmic facilitation of gait training in hemiparetic stroke rehabilitation. *Journal of Neurological Sciences*, 151(2):1–5, 2003.
- [128] C. Y. Wan and G. Schlaug. Music making as a tool for promoting brain plasticity across the life span. *Neuroscientist*, 16(5):566–577, 2010.
- [129] R. Griffiths. *The Abilities of Young Children: A Comprehensive System of Mental Measurement for the First Eight Years of Life*. Child Development Research Centre, Lubbock, TX, USA, 1970.
- [130] J. Pussinen. Behavioural cloning in video games. Master’s thesis, University of Eastern Finland, Kuopio, Finland, 2021.
- [131] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, 1998.
- [132] A. Giuseppi, S. Manfredi, D. Menegatti, C. Poli, and A. Pietrabissa. Decentralised federated learning for hospital networks with application to covid-19 detection. *IEEE Access*, 10:92681–92691, 2022.
- [133] A. Giuseppi, S. Manfredi, and A. Pietrabissa. A weighted average consensus approach for decentralized federated learning. *Machine Intelligence Research*, 19:319–330, 2022.