Achieving reliable control : robustness and stability in nonlinear systems via DNN-based feedback design

Samuele Zoboli

Université Claude Bernard Lyon 1

THESE de DOCTORAT DE
L'UNIVERSITE CLAUDE BERNARD LYON 1

Ecole Doctorale N°**160**
Électronique, Électrotechnique, Automatique

Discipline: Automatique

Soutenue publiquement le 28/09/2023, par :

Samuele ZOBOLI

Achieving reliable control: robustness and stability in nonlinear systems via DNN-based feedback design

---

Devant le jury composé de:

Bernhard MASCHKE	PU, Université Lyon 1	Président
Christopher KELLETT	Pr, Université nationale australienne	Rapporteur
Marc JUNGERS	DR, CNRS Nancy	Rapporteur
Maryam KAMGARPOUR	PA, École polytechnique fédérale de Lausanne	Examinatrice
Sophie CERF	CR, INRIA Lille	Examinatrice
Vincent ANDRIEU	DR, CNRS Lyon	Directeur de thèse
Daniele ASTOLFI	CR, CNRS Lyon	Co-directeur de thèse
Jilles Steeve DIBANGOYE	Pr, University of Groningen	Co-directeur de thèse

Samuele ZOBOLI

Université Claude Bernard Lyon 1, LAGEPP bâtiment CPE, 3ème étage, bureau G315, 43 bd du 11 Novembre 1918, Villeurbanne Cedex, France.

Email:samuele.zoboli@gmail.com

Keywords: Nonlinear systems, robustness, stability, deep neural networks, contraction, multi-agent synchronization, reinforcement learning.

Mots-clés : Systèmes non linéaires, robustesse, stabilité, réseaux de neurones profonds, contraction, synchronisation multi-agent, apprentissage par renforcement.

Thanks

Before delving into the depths of my PhD work, I can't help but express my deepest gratitude to all the incredible individuals who turned this seemingly unachievable objective into a reality. This manuscript wouldn't exist without them and their support.

First, I would like to thank Marc Jungers and Christopher Kellett for agreeing to review this document and for providing invaluable insights and comments. I am equally thankful to Bernhard Maschke, Sophie Cerf and Maryam Kamgarpour for their participation in the jury. My appreciation extends to everyone I had the chance to collaborate with over the course of these three years, especially Giacomo Casadei, Luca Zaccarian, Andrea Mattioni, Bojan Mavkov, Christophe Prieur, Santiago Sanchez Escalonilla Plaza and Bayu Jayawardhana. I learned a lot from you, and I hope we will be able to work together again in the future.

It is not easy to express in words how grateful I am to my supervisors: Vincent Andrieu, Daniele Astolfi and Jilles S. Dibangoye. You dedicated to me more time then I probably deserved. Throughout these three years, you not only supported and listened to my ideas with the utmost kindness and open-mindedness, but you also instilled in me your contagious love for research. You managed to convince me to pursue an academic career without even trying. You were always ready to steer me back on track when I was veering off-course, and you never shied away from questioning your initial thoughts. You taught me to always be ready to change my mind and to never limit myself to my (often narrow) point of view. Your influence on me goes well beyond academia, and I consider myself incredibly lucky to have had you as my mentors.

Similar feelings of gratitude arise when I think of all the people I met in Lyon and at LAGEPP. You made me feel at home, even though I was 500km away from my family, and I cannot thank you enough for that. To my girlfriend Maya, you've been my light during these three years, leading me through the darkness and warming me when I needed it most. You made me feel like I truly belonged. To Tommaso, Silvia, Michele, Paula and Juliette, you have been my (mostly) Italian family in Lyon. Thank you for welcoming me like an old friend from my very first day, and for all of the dinners, trips, movies and Mario Kart races. To Rita, Steeven, Romain, Xiaoqian, Maroua, Carla, Kristy, Georgina, Mario, Mohammed, Quentin, Silvia, Valentina and all the PhDs, postdocs and Master students who passed by the LAGEPP, thank you for all the time and experiences we shared. A huge thanks goes also to the members of office G315: Johan, Andreu, Laura, Lucas, Ling, Xueru, Idriss, Alexandre, Francois, Francesco, Idir and Andrea. I'll miss our discussions in the office, ranging from differential geometry to best pubs and restaurants in Lyon. Perhaps now, without my "quiet" background conversations, you can finally work in peace (at least to some extent). A special thanks is reserved to Mattia Giaccagli. We spent far too many hours rewriting the same equations in a thousand different ways, hoping the rules of mathematics would bend to our will and reveal the answers to unsolvable problems. I am thankful for those "wasted" hours, for they were filled with stupid jokes, laughter and questionable music, recalling us that

"We’re all in this together". To Madiha Nadri, Ulysse Serres, Pascal Dufour, Boussad Hamroun and all the permanent members of the lab with whom I had the pleasure of working with, thank you for your invaluable advice and precious support, especially on the teaching front.

Last but definitely not least, I want to express the most heartfelt thanks to my beloved family: Mauro, Tommaso, Daniela and Irene. Without you, I wouldn’t have found the determination to embark on this journey. Your love and encouragement have been my driving force during this years. A similar debt of gratitude is owed to Pi, Manu and my lifelong friends in Modena. Thank you for your presence, friendship and support throughout this years.

A final thought goes to my mother, Patrizia. Even if you’ve not been an active part in this journey, your love, care and presence have shaped me into who I am today. Your teachings reverberate in me and in all my actions, and they will continue to do so for the rest of my life. You will always live in my heart.

Abstract

Context

This thesis focuses on the integration of robustness and stability guarantees in feedback controllers modeled by deep neural networks (DNNs), also known as neural controllers. The main objective is to combine machine learning tools and control theoretic approaches to derive practically applicable controllers with strong theoretical guarantees. Overall, the proposed methodologies and theoretical results are aimed at offering insights into challenges related to stability guarantees, robustness, and the application of control theoretic feedback designs, both in discrete-time and continuous-time settings. The manuscript is divided into two main parts.

In the first part, we investigate how control theory results can be utilized to enhance modern machine learning approaches for system control with stability and robustness guarantees. One key issue with recent DNN-based controllers is the challenge of understanding whether these control laws, trained in simulated environments, maintain their stabilizing properties in real-world scenarios and generalize over model uncertainties. Additionally, the guarantees of stabilizing behavior provided by machine learning-based controllers remain unclear. Therefore, the first part of this manuscript focuses on discrete-time nonlinear systems and analyzes various forms of stability and robustness properties.

To address these challenges, in the first chapter we investigate how guaranteed stability properties can be embedded in DNN-based feedback designs. Recent works derived algorithms capable of achieving guaranteed convergence to desired equilibria. However, they typically require a sufficient number of training steps, whose amount is hard to know in practice. Additionally, these proposed algorithms are often tailored to achieve stabilizing solutions and cannot be easily generalized to existing well-established approaches. Therefore, our primary objective is to propose an algorithm-agnostic methodology that incorporates local stability guarantees in such controllers, independently from their training process or time. To achieve this, we combine local guaranteed controllers from optimal control theory with DNNs.

In the second and last chapter of the first part of the manuscript, we study robustness to model uncertainties. Although robustness-related results exist for the discrete-time nonlinear framework, fundamental contributions addressing non-vanishing disturbances, such as model mismatches, are still lacking. Our goal is to establish conditions for the transfer of stability properties based solely on norms of model mismatches. This analysis will determine whether the existence and stability of equilibrium points for a nominal system imply the existence and stability of equilibrium points for sufficiently similar systems. In turn, these findings will justify the use of adequately accurate simulators for training neural controllers that are intended for real-world scenarios. Moreover, the derived theoretical results will motivate the utilization of

robust output regulation tools, such as integrators, to enhance the robustness of discrete-time controllers (including learned ones), for which formal justification is currently absent in the literature. By leveraging on the tools introduced in the previous chapter, the presented results enable the development of guaranteed robustly stabilizing neural controllers.

Compared to the first part, the second part of the manuscript explores the reverse direction by investigating how optimization and machine learning tools can assist in the application of control theoretic feedback designs without compromising their stabilizing properties. Specifically, we focus on deriving controllers based on contraction analysis for discrete and continuous-time nonlinear systems. Hence, we provide several optimization-based approaches allowing the design of nonlinear control laws making the closed-loop dynamics a contraction.

In the third chapter we start by examining the concept of incremental stability for discrete-time nonlinear systems. While previous works have studied incremental stabilization in the discrete-time context, approaches related to contraction analysis have only emerged in recent years. These techniques provide guarantees of robust convergence to unique trajectories and are particularly suitable for optimization approaches. Hence, our goal is to present novel results on the design of incrementally stabilizing feedback controllers based on discrete-time contraction theory. The proposed design is based on the solution of a convex optimization problem and we show its effectiveness by tackling the problem of multi-agent synchronization. Although DNNs are not the central focus of this chapter, we demonstrate how the presented tools can be applied to systems incorporating DNNs. Furthermore, by establishing a link between the proposed controllers and the solution of specific optimization problems, we pave the way for the future derivation of neural control laws that guarantee robust convergence to unique trajectories.

In the fourth chapter, we shift our focus to continuous-time nonlinear systems. In continuous-time, controllers based on Riemannian metrics have recently been proposed to enforce contraction of closed-loop systems. These control laws offer appealing properties, as they can be applied in very general frameworks and guarantee robust closed-loop behaviors. However, their analytical computation is typically challenging. Therefore, we investigate how DNN-based controllers can help overcome this computational obstacle. Previous studies have already leveraged learning techniques to derive suitable numerical approximations. However, these approaches often rely on nested online optimization, which limits their applicability. Therefore, we derive new closed-form feedback designs that can be directly approximated using DNNs. This direct approximation enables the development of fast and reliable controllers, further highlighting the potential of solutions at the intersection of control theory and machine learning.

Motivated by the aforementioned results, Chapter 5 introduces the concept of k-contraction as a generalization of the notion of contraction. The objective is to provide a broader geometrical understanding of the properties exhibited by contractive dynamics, thereby emphasizing the underlying motivations driving the interest in these properties. Furthermore, this chapter opens up new avenues for exploring future application of neural controllers within the context of contractive systems. Presently, the available results on k-contractive dynamics lack conditions for designing corresponding control laws. To address this gap, we propose a reworking of existing definitions, tailored specifically for feedback design purposes. By doing so, we shed light on the geometric interpretation of k-contraction and take an initial step towards developing controllers that can enforce a rich set of closed-loop behaviors.

Main contributions

In Chapter 1, we propose a methodology to embed local stability guarantees in neural controllers trained using deep reinforcement learning. To this aim, we study how to effectively blend local guaranteed controllers with DNNs, and we validate our solution on multiple benchmarks. The main theoretical contribution of this chapter can be found in the author's publication [S4].

Contribution 1: We provide an algorithm-agnostic methodology to embed local stability guarantees in DNN-based controllers, independently from their training. The proposed solution is experimentally validated on a number of control tasks and standard deep reinforcement learning algorithms.

In Chapter 2, we study robustness properties in the discrete-time nonlinear framework through the analysis of the transfer of stable behaviors between similar systems. Then, we specialize the derived results in the context of output regulation under constant references. Finally, we demonstrate the validity of our theoretical findings by learning a robust DNN-based controller for nuclear fusion reactors. The experimental validation is a joint work with Andrea Mattioni and other researchers at GIPSA-Lab, Grenoble, France. The main contribution of this chapter can be found in the author's publications [S3, S5].

Contribution 2: Through the concept of total stability, we show that conditions based on norms of model differences are sufficient to conclude the transfer of stability properties between discrete-time nonlinear dynamical systems.

Contribution 3: We formally justify the use of integral action for output regulation in discrete-time nonlinear systems.

Contribution 4: We derive the explicit forwarding law for the control of discrete-time linear systems in cascade with an integrator. By showing the optimality of such a control law with respect to a quadratic cost, we generalize the approach of Chapter 1 to derive robustly stabilizing neural controllers.

Contribution 5: We experimentally demonstrate the effectiveness of discrete-time integrators in improving generalization properties of neural controllers learned via model-free deep reinforcement learning.

In Chapter 3, we study incremental stability and contraction analysis for discrete-time nonlinear systems which are not differentiable everywhere. We propose a closed-form feedback design making the closed-loop incrementally stable, and we apply it on the problem of multi-agent synchronization. Most results presented in this chapter can be found in the author's publication [S6].

Contribution 6: We extend existing results on contraction analysis in discrete-time to the non-smooth framework and we propose a closed form feedback design showing gain margin properties.

Contribution 7: We show that the proposed design is solution to an optimization problem with quadratic cost function, opening the way for data-driven learning approaches.

Contribution 8: We propose numerically tractable conditions in the form of generalized eigenvalue problems to compute incrementally stabilizing controllers.

Contribution 9: By leveraging on the above results, we propose controllers solving the multi-agent synchronization problem for homogeneous networks of discrete-time nonlinear agents under generic communication graphs. These results are derived by building on novel conditions for linear systems synchronization.

In Chapter 4, we study the use of Riemannian metric-based incrementally stabilizing controllers in the context of continuous-time multi-agent synchronization and output tracking. We circumvent the complexity of analytically computing the control law by exploiting DNNs. These findings are the results of the joint work with Mattia Giaccagli (synchronization and tracking) and Steeven Janny (tracking). The contributions of this chapter can be found in the author's publications [S2, S8].

Contribution 10: We show that recent conditions for incrementally stabilizing controllers can be relaxed without the loss of stability and robustness guarantees. This directly enables learning guaranteed neural control laws for incremental stabilization based on Riemannian metric.

Contribution 11: We propose an algorithm for learning Riemannian metrics paired with synchronizing and tracking DNN-based controllers, and we experimentally validate the proposed results.

In Chapter 5, we reformulate the notion of k-contraction. Thanks to our novel conditions, we highlight interesting links between multiple definitions of partial stability existing in the literature. The main contributions of this chapter have been obtained working with Andreu Cecilia Piñol. The results of this chapter can be found in the author's publications [S7].

Contribution 12: We propose novel design-friendly conditions for k-contraction. Moreover, we shed light on connections between multiple existing notions in the field of partial stability analysis.

Résumé

Contexte

Cette thèse se concentre sur l'intégration des garanties de robustesse et de stabilité dans les contrôleurs de rétroaction modélisés par des réseaux neuronaux profonds (DNNs), également connus sous le nom de contrôleurs neuronaux. L'objectif principal est de combiner des outils d'apprentissage automatique et des approches théoriques du contrôle pour dériver des contrôleurs applicables dans la pratique, tout en offrant des garanties théoriques solides. Dans l'ensemble, les méthodologies proposées et les résultats théoriques visent à donner un aperçu des défis liés aux garanties de stabilité, à la robustesse et à l'application des lois de rétroaction de la théorie du contrôle, à la fois dans des contextes à temps discret et à temps continu. Le manuscrit est divisé en deux parties principales.

Dans la première partie, nous étudions comment les résultats de la théorie du contrôle peuvent être utilisés pour améliorer les approches modernes d'apprentissage automatique pour le contrôle des systèmes avec des garanties de stabilité et de robustesse. L'un des principaux problèmes posés par les contrôleurs récents basés sur les DNN est la difficulté de comprendre si ces lois de contrôle, entraînée dans des environnements simulés, conservent leurs propriétés stabilisatrices dans des scénarios réels et se généralisent en fonction des incertitudes du modèle. En outre, les garanties de comportement stabilisant fournies par les contrôleurs basés sur l'apprentissage automatique restent floues. Par conséquent, la première partie de ce manuscrit se concentre sur les systèmes non linéaires à temps discret et analyse diverses formes de propriétés de stabilité et de robustesse.

Pour relever ces défis, dans le premier chapitre nous étudions la comment les propriétés de stabilité garantie peuvent être intégrées dans les lois de rétroaction basées sur les DNN. Des travaux récents ont dérivé des algorithmes capables de garantir la convergence vers les équilibres souhaités. Cependant, ils nécessitent généralement un nombre suffisant d'étapes d'apprentissage, dont la quantité est difficile à connaître dans la pratique. En outre, les algorithmes proposés sont souvent conçus pour atteindre des solutions stabilisantes et ne peuvent pas être facilement généralisés aux approches existantes bien établies. Par conséquent, notre principal objectif est de proposer une méthodologie indépendante des algorithmes qui intègre des garanties de stabilité locale dans ces contrôleurs, indépendamment de leur processus ou temps d'apprentissage. Pour ce faire, nous combinons des contrôleurs à garantie locale issus de la théorie du contrôle optimal avec des DNNs.

Dans le deuxième et dernier chapitre de la première partie du manuscrit, nous étudions la robustesse aux incertitudes du modèle. Bien que des résultats relatifs à la robustesse existent pour le cadre non linéaire à temps discret, des contributions fondamentales traitant des perturbations constantes, telles que les incertitudes de modèles, font encore défaut. Notre objectif est d'établir

des conditions permettant le transfert des propriétés de stabilité, en se basant uniquement sur les normes des erreurs entre les modèles. Cette analyse déterminera si l'existence et la stabilité des points d'équilibre d'un système nominal impliquent l'existence et la stabilité des points d'équilibre de systèmes suffisamment similaires. À leur tour, ces résultats justifieront l'utilisation de simulateurs suffisamment précis pour l'entraînement de contrôleurs neuronaux destinés à des scénarios réels. En outre, les résultats théoriques dérivés motiveront l'utilisation d'outils de régulation de sortie robustes, tels que les intégrateurs, pour améliorer la robustesse des contrôleurs à temps discret (y compris les contrôleurs appris), pour lesquels il n'existe actuellement aucune justification formelle dans la littérature. En s'appuyant sur les outils introduits dans le chapitre précédent, les résultats présentés permettent le développement de contrôleurs neuronaux à stabilisation robuste garantie.

Par rapport à la première partie, la deuxième partie du manuscrit explore la direction inverse en étudiant comment les outils d'optimisation et d'apprentissage automatique peuvent aider à l'application des conceptions de rétroaction de la théorie du contrôle sans compromettre leurs propriétés stabilisatrices. Plus précisément, nous nous concentrons sur la dérivation de contrôleurs basés sur l'analyse de contraction pour les systèmes non linéaires à temps discret et continu. Ainsi, nous fournissons plusieurs approches basées sur l'optimisation permettant la conception de lois de contrôle non linéaires faisant de la dynamique de la boucle fermée une contraction.

Dans le troisième chapitre, nous commençons par examiner le concept de stabilité incrémentale pour les systèmes non linéaires à temps discret. Alors que des travaux antérieurs ont étudié la stabilisation incrémentale dans le contexte du temps discret, les approches liées à l'analyse de contraction n'ont émergé que ces dernières années. Ces techniques fournissent des garanties de convergence robuste vers des trajectoires uniques et sont particulièrement adaptées aux approches d'optimisation. Notre objectif est donc de présenter de nouveaux résultats sur la conception de contrôleurs à rétroaction à stabilisation incrémentale basés sur la théorie de la contraction en temps discret. La conception proposée est basée sur la résolution d'un problème d'optimisation convexe et nous démontrons son efficacité en abordant le problème de la synchronisation multi-agents. Bien que les DNNs ne soient pas au centre de ce chapitre, nous démontrons comment les outils présentés peuvent être appliqués à des systèmes incorporant des DNNs. De plus, en établissant un lien entre les contrôleurs proposés et la solution de problèmes d'optimisation spécifiques, nous ouvrons la voie à la dérivation future de lois de contrôle neuronales qui garantissent une convergence robuste vers des trajectoires uniques.

Dans le quatrième chapitre, nous nous concentrons sur les systèmes non linéaires à temps continu. En temps continu, des contrôleurs basés sur des métriques riemanniennes ont récemment été proposés pour garantir la contraction des systèmes en boucle fermée. Ces lois de contrôle présentent des propriétés attrayantes, car elles peuvent être appliquées dans des cadres très généraux et garantissent des comportements robustes souhaitables pour le système en boucle fermée. Malheureusement, leur calcul analytique est généralement difficile. C'est pourquoi nous étudions comment les contrôleurs basés sur les DNN peuvent aider à surmonter cet obstacle. Des études antérieures ont déjà exploité des techniques d'apprentissage pour dériver des approximations numériques appropriées. Cependant, ces approches reposent souvent sur une optimisation en ligne imbriquée, ce qui limite leur applicabilité. Par conséquent, nous dérivons de nouvelles lois de rétroaction de forme fermée qui peuvent être directement approximées à l'aide des DNNs. Cette approximation directe permet de développer des contrôleurs rapides et fiables, mettant ainsi en évidence le potentiel des solutions à l'intersection de la théorie du

contrôle et de l'apprentissage automatique.

Motivé par les résultats susmentionnés, le Chapitre 5 introduit le concept de k-contraction comme une généralisation de la notion de contraction. L'objectif est de fournir une compréhension géométrique plus large des propriétés exhibées par les dynamiques contractives, mettant ainsi l'accent sur les motivations sous-jacentes à l'origine de l'intérêt pour ces propriétés. En outre, ce chapitre ouvre de nouvelles voies pour explorer les applications futures des contrôleurs neuronaux dans le contexte des systèmes contractifs. Actuellement, les résultats disponibles sur la dynamique k-contractive manquent de conditions pour concevoir les lois de contrôle correspondantes. Pour combler cette lacune, nous proposons un remaniement des définitions existantes, adapté spécifiquement à la conception de lois de contrôle. Ce faisant, nous mettons en lumière l'interprétation géométrique de la k-contraction et faisons un premier pas vers le développement de contrôleurs capables d'appliquer un riche ensemble de comportements en boucle fermée.

Contributions principales:

Dans le Chapitre 1, nous proposons une méthodologie pour intégrer des garanties de stabilité locale dans les contrôleurs neuronaux formés à l'aide de l'apprentissage par renforcement profond. Dans ce but, nous étudions comment combiner efficacement des contrôleurs à garantie locale avec des DNN, et nous validons notre solution sur de nombreux benchmarks. La principale contribution théorique de ce chapitre se trouve dans la publication de l'auteur [S4].

Contribution 1: Nous fournissons une méthodologie agnostique pour intégrer des garanties de stabilité locale dans les contrôleurs basés sur les DNN, indépendamment de leur formation. La solution proposée est validée expérimentalement sur un certain nombre de tâches de contrôle et d'algorithmes standards d'apprentissage par renforcement profond.

Dans le Chapitre 2, nous étudions les propriétés de robustesse dans le cadre non linéaire à temps discret à travers l'analyse du transfert de comportements stables entre des systèmes similaires. Ensuite, nous spécialisons les résultats dérivés dans le contexte de la régulation de la sortie sous des références constantes. Enfin, nous démontrons la validité de nos résultats théoriques en apprenant un contrôleur robuste basé sur un DNN pour les réacteurs à fusion nucléaire. La validation expérimentale est un travail conjoint avec Andrea Mattioni et d'autres chercheurs au GIPSA-Lab, Grenoble, France. La contribution principale de ce chapitre peut être trouvée dans les publications de l'auteur [S3, S5].

Contribution 2: Grâce au concept de stabilité totale, nous montrons que les conditions basées sur les normes des différences de modèles sont suffisantes pour conclure le transfert des propriétés de stabilité entre les systèmes dynamiques non linéaires à temps discret.

Contribution 3: Nous justifions formellement l'utilisation de l'action intégrale pour la régulation de sortie dans les systèmes non linéaires à temps discret.

Contribution 4: Nous dérivons la loi de forwarding explicite pour le contrôle des systèmes linéaires à temps discret en cascade avec un intégrateur. En montrant l'optimalité d'une telle loi de contrôle par rapport à un coût quadratique, nous généralisons l'approche du Chapitre 1 pour dériver des contrôleurs neuronaux robustes et stabilisants.

Contribution 5: Nous démontrons expérimentalement l'efficacité des intégrateurs à temps discret dans l'amélioration des propriétés de généralisation des contrôleurs neuronaux appris via l'apprentissage par renforcement profond sans modèle.

Dans le Chapitre 3, nous étudions la stabilité incrémentale et l'analyse de contraction pour les systèmes non linéaires à temps discret qui ne sont pas différentiables partout. Nous proposons une conception de rétroaction de forme fermée qui rend la boucle fermée incrémentalement stable, et nous l'appliquons au problème de la synchronisation multi-agents. La plupart des résultats présentés dans ce chapitre peuvent être trouvés dans la publication de l'auteur [S6].

Contribution 6: Nous étendons les résultats existants sur l'analyse de la contraction en temps discret au cadre non lisse et nous proposons une conception de rétroaction de forme fermée montrant des propriétés de marge de gain.

Contribution 7: Nous montrons que la conception proposée est une solution à un problème d'optimisation avec une fonction de coût quadratique, ce qui ouvre la voie à des approches d'apprentissage basées sur les données.

Contribution 8: Nous proposons des conditions numériques traçables sous la forme de problèmes de valeurs propres généralisées pour calculer des contrôleurs à stabilisation incrémentale.

Contribution 9: En s'appuyant sur les résultats ci-dessus, nous proposons des contrôleurs qui résolvent le problème de synchronisation multi-agents pour des réseaux homogènes d'agents non linéaires à temps discret dans des graphes de communication génériques. Ces résultats sont dérivés de nouvelles conditions de synchronisation des systèmes linéaires.

Dans le Chapitre 4, nous étudions l'utilisation de contrôleurs à stabilisation incrémentale basés sur des métriques riemanniennes dans le contexte de la synchronisation multi-agents à temps continu et du suivi de sortie. Nous contournons la complexité du calcul analytique de la loi de contrôle en exploitant les DNN. Ces résultats sont le fruit d'un travail conjoint avec Mattia Giaccagli (synchronisation et suivi) et Steeven Janny (suivi). Les contributions de ce chapitre peuvent être trouvées dans les publications de l'auteur [S2, S8].

Contribution 10: Nous montrons que les conditions récentes pour les contrôleurs à stabilisation incrémentale peuvent être assouplies sans perte de garanties de stabilité et de robustesse. Cela permet directement d'apprendre des lois de contrôle neuronales garanties pour la stabilisation incrémentale basée sur la métrique de Riemann.

Contribution 11: Nous proposons un algorithme d'apprentissage des métriques riemanniennes associé à des contrôleurs basés sur des DNN de synchronisation et de suivi, et nous validons expérimentalement les résultats proposés.

Dans le Chapitre 5, nous reformulons la notion de k-contraction. Grâce à nos nouvelles conditions, nous mettons en évidence des liens intéressants entre les multiples définitions de la stabilité partielle existant dans la littérature. Les principales contributions de ce chapitre ont été obtenues en travaillant avec Andreu Cecilia Piñol. Les résultats de ce chapitre peuvent être trouvés dans les publications de l'auteur [S1, S7].

Contribution 12: Nous proposons de nouvelles conditions pour la k-contraction. En outre, nous mettons en lumière les liens entre plusieurs notions existantes dans le domaine de l'analyse de la stabilité partielle.

Contents

List of Figures

List of Tables

Notation

$\mathbb{R}$	Real numbers.
$\mathbb{R}_{\geqslant 0}$	Non-negative real numbers $\mathbb{R}_{\geqslant 0} := [0, +\infty)$.
$\mathbb{R}_{>0}$	Positive real numbers, $\mathbb{R}_{>0} := (0, +\infty)$.
$\mathbb{Z}$	Integer numbers.
$\mathbb{N}$	Non-negative integer numbers.
$\mathbb{C}$	Complex numbers.
$\mathbb{S}^n_{\succ 0}$	Set of symmetric positive definite matrices of dimension n, i.e., $\mathbb{S}^n_{\succ 0} := \{A \in \mathbb{R}^{n\times n} \vert A = A^\top, A \succ 0\}$.
$\mathbb{S}^n_{\succeq 0}$	Set of symmetric positive semi-definite matrices of dimension n, i.e., $\mathbb{S}^n_{\succ 0} := \{A \in \mathbb{R}^{n\times n} \vert A = A^\top, A \succeq 0\}$.
$\mathfrak{Re}(\lambda)$	Real part of the complex number $\lambda \in \mathbb{C}$.
$\mathfrak{Im}(\lambda)$	Imaginary part of the complex number $\lambda \in \mathbb{C}$.
λ^*	Complex conjugate of the complex number $\lambda \in \mathbb{C}$.
Λ^*	Conjugate transpose of the complex matrix $\Lambda \in \mathbb{C}^{n\times m}$.
$\mathrm{spec}(A)$	Spectrum of $A \in \mathbb{R}^{n\times n}$.
$\mathfrak{t}$	Discrete time instant, $\mathfrak{t} \in \mathbb{Z}$.
t	Continuous time instant, $t \in \mathbb{R}$.
x^+	Next discrete-time state , $x^+ := x(\mathfrak{t}+1) \in \mathbb{R}^n$.
$\dot{x}$	Continuous-time state time-derivative, $\dot{x} := \frac{dx}{dt} \in \mathbb{R}^n$.
$\vert x\vert$	Standard Euclidean norm of $x \in \mathbb{R}^n$.
$\mathrm{d}(x, \mathcal{X})$	Generic distance function between $x \in \mathbb{R}^n$ and a closed set $\mathcal{X} \subset \mathbb{R}^n$, e.g., $\mathrm{d}(x, \mathcal{X}) = \vert x\vert_{\mathcal{X}} := \inf_{z\in\mathcal{X}} \vert x - z\vert$.
$\mathcal{B}_r(x)$	Open ball of radius r centered in $x \in \mathbb{R}^n$, i.e., $\mathcal{B}_r(x) := \{z \in \mathbb{R}^n : \vert z - x\vert \leqslant r\}$.
$\mathcal{X} \setminus \mathcal{Y}$	Intersection between $\mathcal{X}$ and the complement of $\mathcal{Y}$.
$\mathcal{X} \subset \mathcal{Y}$	The set $\mathcal{S}$ is strictly included in $\mathcal{Y}$.
$\partial\mathcal{X}$	Boundary of $\mathcal{X}$.
$\alpha \in \mathcal{K}$	The function $\alpha : \mathbb{R}_{\geqslant 0} \to \mathbb{R}_{\geqslant 0}$ is of class $\mathcal{K}$, i.e., it is continuous, zero at zero and strictly increasing.
$\alpha \in \mathcal{K}_\infty$	The function $\alpha : \mathbb{R}_{\geqslant 0} \to \mathbb{R}_{\geqslant 0}$ is of class $\mathcal{K}_\infty$, i.e., $\alpha \in \mathcal{K}$ and $\lim_{s\to\infty} \alpha(s) = \infty$.
$\alpha \in \mathcal{L}$	The function $\alpha : \mathbb{R}_{\geqslant 0} \to \mathbb{R}_{\geqslant 0}$ is of class $\mathcal{L}$, i.e., it is continuous, strictly decreasing and $\lim_{s\to\infty} \alpha(s) = 0$.
$\alpha \in \mathcal{KL}$	The function $\alpha : \mathbb{R}_{\geqslant 0} \times \mathbb{R}_{\geqslant 0} \to \mathbb{R}_{\geqslant 0}$ is of class $\mathcal{KL}$, i.e., it is class-$\mathcal{K}$ in its first argument and class$\mathcal{L}$ in its second argument.
I_n	Identity matrix of dimension n.
$\mathbf{1}$ (resp. $\mathbf{0}$)	Column vector of 1s (resp. 0s) of appropriate dimension.
$\mathrm{col}(x, u)$	Column vector composed by the elements of $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$, i.e., $\mathrm{col}(x, u) := (x^\top\ u^\top)^\top \in \mathbb{R}^{m+n}$.

$\mathrm{diag}(A, B)$	Block-diagonal matrix with block-diagonal elements $A \in \mathbb{R}^{n\times m}$ and $B \in \mathbb{R}^{p\times q}$, i.e., $\mathrm{diag}(A, B) := \left(\begin{smallmatrix} A & 0 \\ 0 & B \end{smallmatrix}\right) \in \mathbb{R}^{(n+p)\times(m+q)}$.
$\binom{n}{k}$	Binomial coefficient $\binom{n}{k} := \frac{n!}{k!(n-k)!}$, with $n!$ the factorial of $n \in \mathbb{N}$.
$A \otimes B$	Kronecker product between matrices $A \in \mathbb{R}^{n\times m}$ and $B \in \mathbb{R}^{p\times q}$.
$\mathrm{He}\{A\}$	Hermitian of $A \in \mathbb{R}^{n\times n}$, i.e., $\mathrm{He}\{A\} := A^\top + A$.
$\mathbb{E}[x\|y]$	Expected value of x given y.
$\mathrm{Var}[x]$	Variance of x.

Acronyms

DNN	Deep Neural Network
RNN	Recurrent Neural Network
DRL	policy-based Deep Reinforcement Learning
PDE	Partial Differential Equation
PDI	Partial Differential Inequality
ARE	continuous-time Algebraic Riccati Equation
DARE	Discrete-time Algebraic Riccati Equation
DARI	Discrete-time Algebraic Riccati Inequality
MARI	Modified discrete-time Algebraic Riccati Inequality
DOA	Domain Of Attraction
δGAS	Incrementally Globally Asymptotically Stable
δGES	Incrementally Globally Exponentially Stable
ISS	Input-to-State Stable
δISS	Incrementally Input-to-State Stable
LQR	Linear Quadratic Regulator
DDPG	Deep Deterministic Policy Gradient
PPO	Proximal Policy Optimization
TD3	Twin Delayed Deep Deterministic Policy Gradient
SAC	Soft Actor-Critic
PSU	Pendulum Swing-Up
IPSU	Inverted Pendulum Swing-Up
DIPSU	Double Inverted Pendulum Swing-Up

Part I

Robust optimal feedback design

Introduction

In recent years, the use of deep neural networks (DNNs) for controlling dynamical systems has gained significant attention due to advances in machine learning techniques [94, 129, 188]. DNN-based controllers, also known as *neural controllers*, have shown promising results on various tasks. Training these controllers involves optimizing the network parameters to minimize a loss function (or maximize a reward function), employing techniques such as supervised learning and policy-based deep reinforcement learning (DRL) algorithms [15, 91, 125, 187]. In particular, recent years saw the surge of model-free DRL methods for system control, due to their ability to autonomously learn how to solve complex high-dimensional control tasks without the need of a known model of the system, e.g., [147, 188, 195]. While model-free DRL methods excel at solving complicated tasks on a performance level, only few of them study the stability properties of the closed-loop in a Lyapunov sense [113]. As a consequence, these approaches often suffer from brittleness, which has prompted efforts to enhance their robustness, either via modified objectives or via specific algorithm designs, e.g. [26, 49, 66, 150, 169, 214].

It is well-known that robustness to uncertainties is of paramount importance in the design of feedback controllers. Ensuring convergence guarantees to a predetermined behavior is crucial for the reliability and effectiveness of controlled systems. Traditionally, stability properties are inferred through the analysis of the system model in closed-loop. However, uncertainties inherently exist in control applications due to unmodeled effects and parameter mismatches. This is especially true for complex nonlinear systems, where simplification is unavoidable for obtaining a tractable model. To address this challenge, robust control and robust stability analysis have become fundamental tools for control design [246].

As a matter of fact, the problem of synthesizing robust controllers can be formulated as an optimization problem, both in linear and nonlinear scenarios, as demonstrated in multiple prior works [59, 115, 123, 167, 168]. Nevertheless, a strong theoretical foundation is crucial to understand how to properly formulate such robust learning tasks. Hence, in this first part of the manuscript, we introduce simple yet theoretically sound methods to embed stability and robustness properties in neural controllers. As a fundamental constraint, we assume these control laws to be trained on simplified model. Hence, the proposed robustness tools will serve as a key means for translating simulated results to real-world applications.

In particular, we will focus on discrete-time nonlinear systems. In practical control implementations, control laws are often implemented using digital devices that operate at discrete time instants due to the limitations of digital sensors and internal clocks. While the theory on robust stability and regulation is well-developed for continuous-time systems [17, 77], there are still gaps in the understanding of the discrete-time nonlinear scenario. Although it is known that continuous-time results can be valid for small sampling times in the discrete framework [152], this approach may not always be applicable or may be too restrictive for certain control applications. Therefore, designing controllers based on a discrete-time model of the plant can be effective [153]. Moreover, it's worth recalling that learning-to-control tasks arising in machine learning, e.g. in DRL, are often formulated in discrete-time [28, 122, 210].

Given the above discussion, we start our analysis by studying how to embed closed-loop stability guarantees in the learned controller. This first step lays the foundations for the developments of the successive chapter, as the methods presented for obtaining robust regulation require local stability properties. The proposed methodology is experimentally validated and thoroughly analyzed. A portion of this chapter was presented in [S4].

In Chapter 2, we introduce robust stability results for discrete-time nonlinear systems. In the first sections, we focus our attention towards stabilization of an equilibrium point and we show that stability alone is insufficient to achieve robust regulation. To do so, we firstly recover and reinterpret a notion of robustness (namely *total stability*) which is strongly related to generalization capabilities over model uncertainties, directly addressing the gap between simulated and real world tasks. Then, we propose tools to embed such robustness guarantees in discrete-time neural controllers, specializing our study on the important problem of constant signals rejection (e.g., caused by unknown system parameters). Inspired by the results of the previous chapter, we study stabilization properties and optimality of forwarding-based laws in the linear framework, with the aim of globalizing them to the nonlinear context via learning. The proposed solution is also experimentally validated, reinforcing the theoretical claims and stressing the importance of robustness properties in DNN-based controllers. All of the above results are presented in Chapter 2 and covered in [S3, S5].

1 Local stability via neural controllers

Machine learning optimization tools have been proven to be valuable and effective in addressing complex control problems [15, 125]. However, it is worth noting that conventional optimization of controller parameters using off-the-shelf model-free DRL algorithms may not provide desirable stability properties. As a consequence, the applicability of such methods in real-world scenarios, where stringent stability and robustness guarantees are indispensable at least in a local context, can be significantly compromised. Therefore, it is in our interest to investigate how (local) exponential stability guarantees can be integrated into a DNN-based controller, where the weights are learned using DRL methods. For an introduction on the fundamental DRL concepts used in this chapter, we refer the reader to Appendix A.1.

Recently, the inclusion of Lyapunov theory in learning algorithms received increasing attention [25, 26, 46, 48, 49, 93, 241, 244]. One early solution was proposed in [26], following the work in [25]. The authors relied on a learned Gaussian model to certify Lyapunov stability, while assuming a suitable initial safe controller and Lyapunov function are known. In [49], the authors claim near-constraint satisfaction via policy gradient methods by projecting either the controller parameters or the input onto a feasible set described by a Lyapunov constraint. Last, [95] uses a Lyapunov function as a cost estimator for guaranteeing stability in the mean cost of the learned control law. However, most existing approaches asymptotically learn stabilizing controllers, but provide no guarantees throughout the training process. This characteristic is critical to disentangle fundamental properties, such as local stability, from training time, and to ensure a robust behavior even for sub-optimally trained controllers.

In what follows, we propose a methodology to integrate basic model knowledge in standard model-free DRL algorithms. The aim is to safely blend a local controller guaranteeing relevant properties with a general function approximator (a DNN). By enforcing a predefined structure to the control law, we provide theoretical guarantees concerning the local stability of the system via Lyapunov indirect theorem. This provides a theoretical framework to enhance many DRL algorithms with local exponential stability guarantees, including at the training phase. We focus on the optimal control of nonlinear systems under quadratic cost functions. The approach can be seen as enhancing the capabilities of a linear controller by learning to operate outside its domain of attraction, while maintaining its local properties. However, as it will be shown in the last section of the chapter, these results can be extended to more general objective functions at the price of sub-optimal performances. With a slight abuse of terminology, throughout the rest of

this chapter we will use the terms locally exponentially stable, locally asymptotically stable and locally stable interchangeably. The results of this chapter are partially covered in [S4].

1.1 Globalizing LQR policies via deep reinforcement learning

Since we aim at embedding local properties in nonlinear controllers optimal with respect to a quadratic function, we start by recalling some important facts about the linear quadratic regulator (LQR) problem. Consider a discrete-time linear system

$$x^+ = Ax + Bu, \tag{1.1}$$

with state $x \in \mathbb{R}^{n_x}$ and control input $u \in \mathbb{R}^{n_u}$ at time $\mathfrak{t}$. We suppose that the pair (A, B) is controllable. Moreover, without loss of generality, we assume the origin to be an equilibrium point for system (1.1). The infinite-horizon LQR problem asks for a stabilizing controller optimal with respect to a quadratic cost function. By letting $\gamma \in (0, 1]$ be a discount factor and $\mathfrak{t} \in \mathbb{N}$ be an initial time, the discounted problem considers the minimization of the following cost function

$$J_\gamma(x(\mathfrak{t}), u(\mathfrak{t})) = \sum_{k=0}^{\infty} \gamma^k \left(x(k+\mathfrak{t})^\top Q_\gamma x(k+\mathfrak{t}) + u(k+\mathfrak{t})^\top R_\gamma u(k+\mathfrak{t}) \right), \tag{1.2}$$

where $Q_\gamma \in \mathbb{S}^{n_x}_{\succeq 0}$ and $R_\gamma \in \mathbb{S}^{n_u}_{\succ 0}$. Under proper observability and controllability assumptions, for a given discount factor γ, the optimal controller $u^\star$ for (1.1) with cost (1.2) is given as

$$u^\star = K^\star_\gamma x, \quad K^\star_\gamma = -\gamma(R_\gamma + \gamma B^\top P_\gamma B)^{-1} B^\top P_\gamma A, \tag{1.3}$$

where $K^\star_\gamma$ is the optimal discounted gain and P_γ is the unique positive definite solution of the discounted Discrete-time Algebraic Riccati Equation (DARE)

$$P_\gamma = Q_\gamma + \gamma A^\top (P_\gamma - \gamma P_\gamma B (R_\gamma + \gamma B^\top P_\gamma B)^{-1} B^\top P_\gamma) A, \tag{1.4}$$

see, e.g., [27, Chapter 4.3]. Equation (1.4) can be rewritten and solved as a standard DARE for the new matrices $\widetilde{A} = \sqrt{\gamma} A$, $\widetilde{B} = \sqrt{\gamma} B$. From [33, Section 3], the value functions for the discounted LQR problem under the optimal controller are

$$\mathbf{J}^\star_\gamma(x) = x^\top P_\gamma x, \qquad \mathbf{Q}^\star_\gamma(x, u) = z^\top \mathbf{H}_\gamma z, \tag{1.5}$$

with $z := \mathrm{col}(x, u) \in \mathbb{R}^{n_x + n_u}$ and $\mathbf{H}_\gamma$ given by

$$\mathbf{H}_\gamma = \begin{pmatrix} Q_\gamma + \gamma A^\top P_\gamma A & \gamma A^\top P_\gamma B \\ \gamma B^\top P_\gamma A & R_\gamma + \gamma B^\top P_\gamma B \end{pmatrix}.$$

In the discounted framework, closed-loop stability is dependent on the choice of the discount factor γ. This may be intuitively seen from the fact that finding the matrix P_γ accounts to solving a standard DARE for a system rescaled by γ, hence different from the plant. In [170, Section IV, Corollary 3] the authors define a conservative lower bound $\gamma^* \in (0, 1]$, depending on Q_γ and R_γ, such that for any $\gamma \in (\gamma^*, 1]$ the origin of the closed-loop (1.1)-(1.3) is exponentially stable. Note that for $\gamma = 1$, we recover the so-called "undiscounted LQR" problem in which the cost function reads

$$J(x(\mathfrak{t}), u(\mathfrak{t})) := J_1(x(\mathfrak{t}), u(\mathfrak{t})) = \sum_{k=0}^{\infty} x(k+\mathfrak{t})^\top Q x(k+\mathfrak{t}) + u(k+\mathfrak{t})^\top R u(k+\mathfrak{t}), \tag{1.6}$$

with optimal solution $u^\star$ given by

$$u^\star = K^\star x, \quad K^\star = -(R + B^\top P B)^{-1} B^\top P A, \tag{1.7}$$

with P the symmetric positive definite matrix solution of the DARE

$$P = A^\top P A - A^\top P B (R + B^\top P B)^{-1} B^\top P A + Q. \tag{1.8}$$

For the undiscounted problem, stability of the closed-loop system is always guaranteed.

Remark 1.1. *In what follows, we will focus on algorithms learning the parameter vector θ of a deterministic parametrized policy $\boldsymbol{\pi}^\theta : \mathbb{R}^{n_\theta} \to \mathbb{R}^{n_u}$ via* (A.3)*. However the proposed results can be directly extended to include methods exploiting stochastic policies since the proposed solution is independent from the choice of the DRL algorithm.*

Recalling the concepts of Appendix A.1, we now tackle the nonlinear framework. For simplicity of exposition, we consider the single input scenario ($n_u = 1$). However, the proposed results can be straightforwardly generalized to the multi-input case. Consider a deterministic discrete-time nonlinear system

$$x^+ = f(x, u), \tag{1.9}$$

with state $x \in \mathbb{R}^{n_x}$, control input $u \in \mathcal{U} \subseteq \mathbb{R}$, and $f : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_x}$ being a continuously differentiable function in a neighborhood of the origin. Without loss of generality we suppose that $f(0, 0) = 0$. Similarly to the problem in Section 2.3, we aim at learning an optimal policy with respect to an undiscounted quadratic cost function. However, we also require such a policy to be locally stabilizing at all times. Moreover, as a first step, we assume a good model for the system is known solely around the desired equilibrium. In other words, we assume f is unknown, and only its linearization of the form (1.1) with $A := \frac{\partial f}{\partial x}(0, 0)$ and $B := \frac{\partial f}{\partial u}(0, 0)$ is known. At experiment time , we will also relax this assumption and we will rely on system identification tools to estimate the required quantities. We formalize the problem as follows

Problem 1.1. *Let J be an undiscounted cost function of the form* (1.6) *and assume a linearized model of the form* (1.1) *for* (1.9) *is known, with (A, B) stabilizable. The goal is to learn an optimal parametrized control policy $\boldsymbol{\pi}^\theta : \mathbb{R}^{n_x} \times \mathbb{R}^{n_\theta} \to \mathcal{U}$ with parameters $\theta \in \mathbb{R}^{n_\theta}$ such that the origin of the closed-loop system* (1.9) *with $u = \boldsymbol{\pi}^\theta(x)$ is locally asymptotically stable for all $\theta \in \mathbb{R}^{n_\theta}$, namely*

$$\frac{\partial \boldsymbol{\pi}^\theta}{\partial x}(0) = K^\star, \qquad \forall\, \theta \in \mathbb{R}^{n_\theta}, \tag{1.10}$$

where $K^\star \in \mathbb{R}^{n_x}$ is the LQR optimal gain given in (1.7)*.*

Remark 1.2. *Solving the optimal problem implies requirement* (1.10) *since $K^\star$ is the unique optimal solution to the local problem. Moreover* (1.10) *ensures local stability around the origin by Lyapunov indirect theorem, being $K^\star$ stabilizing for the linearized system.*

1.1.1 Reward shaping

To solve the problem, we want to rely on DRL algorithms. Rewards play a fundamental role in reinforcement learning. Hence, we need to design a suitable cost describing the learning objective. Most DRL algorithms require the value function $\mathbf{J}^{\boldsymbol{\pi}}$ to have a finite value in order to converge. However, when defined as an infinite sum over time, boundedness of $\mathbf{J}^{\boldsymbol{\pi}}$ may not be

always ensured. Hence, starting from J in (1.6) we look for a reward function $\mathbf{r} : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}$ and a discount factor $\gamma \in (0, 1)$ defining a suitable γ-discounted function

$$J_{\gamma,\mathrm{DRL}}(x(\mathfrak{t}), u(\mathfrak{t})) = \sum_{k=0}^{\infty} \gamma^k \mathbf{r}(x(k+\mathfrak{t}), u(k+\mathfrak{t})) \tag{1.11}$$

which sets the learning objective for the agent. In the following lemma, we show that given any undiscounted problem of the form (1.6)-(1.7), we can always redefine an associated discounted problem of the form (1.2)-(1.3) so that the optimal gains (1.7) and (1.3) coincide. In the nonlinear framework, this allows addressing a discounted problem whose local solution perfectly matches the undiscounted one, which is known to be stabilizing.

Lemma 1.1. *Consider system* (1.1) *and an associated undiscounted optimal control problem of the form* (1.6)*. Moreover, let $Q \in \mathbb{S}^{n_x}_{\succ 0}$ and the pair $(A, \sqrt{Q})$ to be observable. Then, for any $\gamma \in (0, 1]$, the optimal gain $K^\star$ defined in* (1.7) *is the unique optimal solution of the discounted problem* (1.2) *with Q_γ, R_γ defined as*

$$Q_\gamma = \gamma Q + (1-\gamma)P, \qquad R_\gamma = \gamma R. \tag{1.12}$$

Moreover, the state-value function (1.5) *of the discounted problem is finite.*

Proof. If $\gamma = 1$ the proof is trivial, hence we will focus on $\gamma \in (0, 1)$. First, note that since Q, P and R are symmetric positive definite matrices, so are Q_γ and R_γ for any $\gamma \in (0, 1)$. By multiplying both sides of the DARE (1.8) by γ and by using (1.12), we get

$$P = Q_\gamma + \gamma A^\top P A - \gamma^2 A^\top P B (R_\gamma + \gamma B^\top P B)^{-1} B^\top P A.$$

Under assumptions of controllability of (A, B) and observability of $(A, \sqrt{Q})$, the solution to the Riccati equation is unique. Hence, from the definition of the discounted DARE (1.4) we can conclude that $P = P_\gamma$. By inserting the latter and (1.12) in the optimal gain (1.7) we obtain

$$K^\star = -\gamma(R_\gamma + \gamma B^\top P_\gamma B)^{-1} B^\top P_\gamma A = K^\star_\gamma,$$

which shows that (1.7) is also solution to the discounted problem defined by (1.2) when considering weights in the form of (1.12).
The state-value function for a given policy is defined as the *cost-to-go* starting from an initial state and following the policy. Given a quadratic cost function as in (1.6), it holds that

$$\mathbf{J}^u(x^+) - \mathbf{J}^u(x) = -x^\top Q x - u^\top R u. \tag{1.13}$$

By plugging the matrices (1.12) in (1.13) and considering the value function (1.5) with (1.8) for the undiscounted problem under the optimal input (1.7), yields

$$\mathbf{J}^\star(x^+) = \gamma^{-1}\mathbf{J}^\star(x) - \gamma^{-1}(x^\top Q_\gamma x + u^{\star\top} R_\gamma u^\star). \tag{1.14}$$

Then, by solving (1.14) we have

$$\gamma^{\mathfrak{t}} \mathbf{J}^\star(x(\mathfrak{t})) = \mathbf{J}^\star(x(0)) - \sum_{j=0}^{\mathfrak{t}-1} \gamma^j (x(j)^\top Q_\gamma x(j) + u^\star(j)^\top R_\gamma u^\star(j)), \tag{1.15}$$

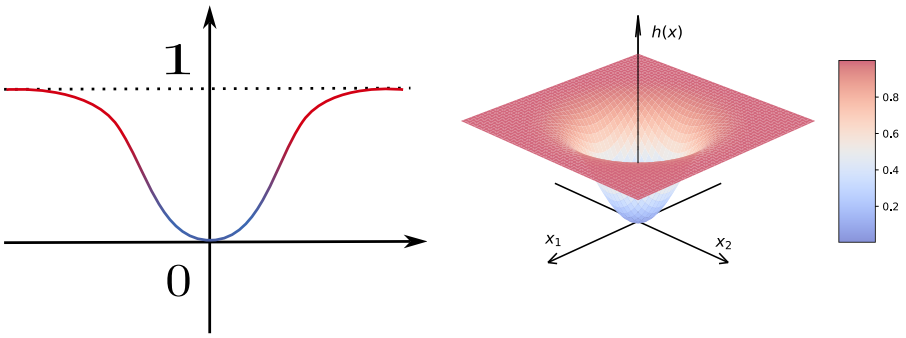


Figure 1.1: Example of blending function $h(x) \in \mathcal{H}_1$ with $x \in \mathbb{R}^2$

where $\mathbf{J}^\star(x(0)) \in \mathbb{R}$ denotes the initial condition. Since $\gamma \in (0,1)$ by letting $\mathfrak{t} \to \infty$ equation (1.15) simplifies in

$$\sum_{j=0}^{\infty} \gamma^j (x(j)^\top Q_\gamma x(j) + u^\star(j)^\top R_\gamma u^\star(j)) = \mathbf{J}^\star(x(0)). \tag{1.16}$$

Being $\mathbf{J}^\star(x(0))$ finite for the linear system (1.1) under the optimal controller (1.7), the state-value function for the discounted problem under the undiscounted solution (1.7) is also finite and this concludes the proof. □

Based on Lemma 1.1, the undiscounted cost J in (1.6) can be replaced by its discounted version (1.2) by using the weights (1.12). Then, we can obtain a suitable form of the cost $J_{\gamma,\mathrm{DRL}}$ for DRL algorithms without affecting the local optimal solution to the problem. Hence, we set the reward $\mathbf{r}$ from (1.11) as

$$\mathbf{r}(x,u) = x^\top Q_\gamma x + u^\top R_\gamma u, \tag{1.17}$$

with Q_γ and R_γ defined as in (1.12). The value of γ is now a free parameter to be chosen as the most suitable one for the algorithm convergence.

1.1.2 Learning the policy

With the previous result in mind, we can move to the control policy design. Our goal is to ensure local stability of the closed-loop system (1.9) with $u = \boldsymbol{\pi}^\theta(x)$, i.e. to satisfy the local approximation constraint (1.10) independently from the parameter vector θ. To this end, we enforce a specific structure in our control policy via a *blending function*. Given any $k \in \mathbb{N}$, we define $\mathcal{H}_k$ as the set of continuously differentiable functions $h : \mathbb{R}^{n_x} \to \mathbb{R}_{\geqslant 0}$ satisfying the following for all $x \in \mathbb{R}^{n_x}$

$$\mathcal{H}_k := \left\{ h \in C^{k+1} : \lim_{s\to 0} \frac{h(sx)}{s^j |x|^j} = 0, \lim_{s\to 0} \frac{h(sx)}{s^{k+1}|x|^{k+1}} \neq 0, \lim_{|s|\to\infty} h(sx) = 1, j \in [0,k] \right\}. \tag{1.18}$$

In few words, the elements of the set $\mathcal{H}_k$ are sufficiently smooth real-valued positive functions that are 0 in the origin, that saturate to 1 far from it and that behave as a function of order $k+1$ with respect to their input in the non-saturating region. For instance, the function $h(x) := \tanh(x^\top x)$ in Figure 1.1 belongs to $\mathcal{H}_1$. The proposed control law is then designed as

$$\boldsymbol{\pi}^\theta(x) = \boldsymbol{\pi}_{\mathrm{loc}}(x) + \boldsymbol{\pi}^\theta_{\mathrm{glo}}(x), \quad \boldsymbol{\pi}_{\mathrm{loc}}(x) = K^\star x, \quad \boldsymbol{\pi}^\theta_{\mathrm{glo}}(x) = h_1(x) \left(\boldsymbol{\mu}^\theta(x) - \boldsymbol{\pi}_{\mathrm{loc}}(x) \right). \tag{1.19}$$

where $h_1 \in \mathcal{H}_1$ and $\boldsymbol{\mu}^\theta : \mathbb{R}^{n_x} \to \mathbb{R}$ is a scalar function to be learned by the DRL agent which is parametrized by the set of parameters $\theta \in \mathbb{R}^{n_\theta}$ and satisfies the following assumption:

Assumption 1.1. *The function $\boldsymbol{\mu}^\theta$ is locally Lipschitz.*

Remark 1.3. *Note that most DNNs are locally Lipschitz since they are compositions of locally Lipschitz functions. When learning a deterministic policy, the function $\boldsymbol{\mu}^\theta$ denotes directly the parametrized approximator to be trained, i.e., the DNN. These policies are often learned with deterministic policy gradient methods (e.g. DDPG [129] and TD3 [80]). Nevertheless, the results of this chapter can be derived for stochastic policies. These are typically learned via stochastic policy gradient methods (e.g. PPO [188] and SAC [94]) and often take the form of a Gaussian distribution whose mean and variance are modeled by DNNs. When not addressing an adversarial scenario, randomness is not necessary at test time, and a deterministic policy is often extracted from the learned stochastic one (e.g., by taking its mean). Then, since we consider a single-agent scenario, we will assume $\boldsymbol{\mu}^\theta$ to be modeled by a DNN, independently from the algorithm used for training.*

Remark 1.4. *The size of the guaranteed domain of attraction for policy* (1.19) *can be controlled by shaping the function h_1. The effect of the learned component in a neighborhood of the origin is scaled by such a function. Hence, it is possible to strongly reduce the contribution of $\boldsymbol{\pi}^\theta_{\text{glo}}(x)$ in regions where we trust the LQR controller $\boldsymbol{\pi}_{\text{loc}}$ to stabilize the system. In the most extreme case, h_1 can be chosen as a smooth step-like function, completely removing the effect of the learned policy in the zero region.*

In order to justify our choice for the control policy, in the following proposition we prove that, by enforcing structure (1.19) and assuming $\boldsymbol{\mu}^\theta$ satisfies Assumption 1.1, we can learn an arbitrary C^2 policy satisfying the problem constraint (1.10). However, before presenting the result, we need to introduce an auxiliary lemma related to first order approximations of functions. Then, for this brief technical result, we will identify by x_i the i^{th} component of vector $x \in \mathbb{R}^{n_x}$. First, we recall some multi-index definitions. For a general multi-index $\mathfrak{i} \in \mathbb{N}^n$ we denote

$$|\mathfrak{i}| = i_1 + \cdots + i_n, \qquad \mathfrak{i}! = i_1! \dots i_n!, \qquad x^{\mathfrak{i}} = x_1^{i_1} \dots x_n^{i_n},$$

for any $x \in \mathbb{R}^{n_x}$. Then, given a function $\Phi : \mathbb{R}^{n_x} \to \mathbb{R}$ whose l-th order partial derivatives are continuous, it is possible to define its derivative of order l as

$$D^{\mathfrak{i}}\Phi := \frac{\partial^{|\mathfrak{i}|}\Phi}{\partial x_1^{i_1} \dots \partial x_n^{i_n}}, \qquad |\mathfrak{i}| = l\,.$$

Finally, we define with $o(x^{m+1})$ the standard little-o notation for functions of order smaller than $|x|^m$. We state now the following lemma.

Lemma 1.2. *For a given point $y \in \mathbb{R}^{n_x}$ and two real-valued functions $\Psi_1, \Psi_2 \in C^{m+1} : \mathbb{R}^{n_x} \to \mathbb{R}$ such that $D^{\mathfrak{i}}\Psi_1(y) = D^{\mathfrak{i}}\Psi_2(y)$, $|\mathfrak{i}| \leqslant m$ where $\mathfrak{i} \in \mathbb{N}^n$ is a multi index, it holds that $\Psi_1(x) = \Psi_2(x) + o(x^{m+2})$.*

Proof. Introduce the multi-indices $\mathfrak{i}, \mathfrak{j} \in \mathbb{N}^n$, $\mathfrak{i} = (i_1, i_2, \dots, i_n)$, $\mathfrak{j} = (j_1, j_2, \dots, j_n)$. Consider an arbitrary real-valued function $\Phi : \mathbb{R}^{n_x} \to \mathbb{R}, \Phi \in C^{l+1}, l \in \mathbb{N}$. Given a point $y \in \mathbb{R}^{n_x}$, by Taylor's theorem for multivariate functions [118] it holds that

$$\Phi(x) = \sum_{|\mathfrak{i}| \leqslant l} \frac{D^{\mathfrak{i}}\Phi(y)}{\mathfrak{i}!}(x-y)^{\mathfrak{i}} + \sum_{|\mathfrak{j}|=l+1} \Upsilon^{\Phi}(x)(x-y)^{\mathfrak{j}},$$

where $\Upsilon^{\Phi}(x) = \frac{|\mathfrak{j}|}{\mathfrak{j}!}\int_0^1 (1-t)^{|\mathfrak{j}|-1} D^{\mathfrak{j}}\Phi(y+s(x-y))ds$. Then, if $\Psi_1, \Psi_2 \in C^{m+1}$ they can be equivalently expressed as

$$\Psi_1(x) = \sum_{|\mathfrak{i}|\leqslant m} \frac{D^{\mathfrak{i}}\Psi_1(y)}{\mathfrak{i}!}(x-y)^{\mathfrak{i}} + o(x^{m+2}), \quad \Psi_2(x) = \sum_{|\mathfrak{i}|\leqslant m} \frac{D^{\mathfrak{i}}\Psi_2(y)}{\mathfrak{i}!}(x-y)^{\mathfrak{i}} + o(x^{m+2}).$$

If $D^{\mathfrak{i}}\Psi_1(y) = D^{\mathfrak{i}}\Psi_2(y)$, $|\mathfrak{i}| \leqslant m$ we can rearrange the last identity to obtain

$$\sum_{|\mathfrak{i}|\leqslant m} \frac{D^{\mathfrak{i}}\Psi_1(y)}{\mathfrak{i}!}(x-y)^{\mathfrak{i}} = \Psi_2(x) - o(x^{m+2}).$$

Finally, by combining all previous equation we obtain $\Psi_1(x) = \Psi_2(x) + o(x^{m+2})$, thus completing the proof. □

We are now ready to state the result showing the generality of structure (1.19) for learning C^2 policies.

Proposition 1.1. *The following statements hold.*

(i) Given any $K^\star \in \mathbb{R}^{n_x}$ and any function $\boldsymbol{\pi}^\theta : \mathbb{R}^{n_x} \to \mathbb{R}, \boldsymbol{\pi}^\theta \in C^2$ satisfying the local approximation constraint (1.10)*, for any function $h_1 \in \mathcal{H}_1$ there always exists a locally Lipschitz function $\boldsymbol{\mu}^\theta : \mathbb{R}^{n_x} \to \mathbb{R}$ satisfying the equality* (1.19) *for all $x \in \mathbb{R}^{n_x}$.*

(ii) Let $\boldsymbol{\mu}^\theta$ satisfy Assumption 1.1, $h_1 \in \mathcal{H}_1$ and $\boldsymbol{\pi}^\theta : \mathbb{R}^{n_x} \to \mathbb{R}$ as in (1.19)*. Then* (1.10) *is satisfied.*

Proof. Let us address point *(i)*. By keeping in mind the multi-index properties, we consider a multi-index $\mathfrak{i} \in \mathbb{N}^n$. Constraint (1.10) implies

$$D^{\mathfrak{i}}\boldsymbol{\pi}^\theta(0) = D^{\mathfrak{i}}\boldsymbol{\pi}_{\mathrm{loc}}(0), \qquad |\mathfrak{i}| \leqslant 1. \tag{1.20}$$

Select $x_0 = 0$. Since $\boldsymbol{\pi}_{\mathrm{loc}} \in C^2$, if $\boldsymbol{\pi}^\theta \in C^2$ by Lemma 1.2, in a neighborhood of the equilibrium we obtain $\boldsymbol{\pi}^\theta(x) = \boldsymbol{\pi}_{\mathrm{loc}}(x) + o(x^3)$. It is possible to find a suitable definition of $\boldsymbol{\pi}^\theta_{\mathrm{glo}}$ satisfying equation (1.19) by recalling that $h_1 \in \mathcal{H}_1$. This shows the first item of the proposition. We prove now the second item. If (1.19) is satisfied for all $x \in \mathbb{R}^{n_x}$, constraint (1.10) is verified if and only if

$$\lim_{s\to 0} \frac{h_1(sx)\boldsymbol{\pi}^\theta_{\mathrm{glo}}(sx)}{s|x|} = 0, \qquad \forall x \in \mathbb{R}^{n_x}, \tag{1.21}$$

where $s \in \mathbb{R}$. If $\boldsymbol{\pi}^\theta_{\mathrm{glo}}$ is a locally Lipschitz function, then $\lim_{s\to 0}\left|\boldsymbol{\pi}^\theta_{\mathrm{glo}}(sx)\right| \leqslant \omega_\pi$ for all $x \in \mathbb{R}^{n_x}$, being $\omega_\pi \in \mathbb{R}_{\geqslant 0}$. This implies that (1.21) holds if $h_1 \in \mathcal{H}_1$ and this concludes the proof. □

Remark 1.5. *Due to* (1.19) *being always enforced, the local guaranteed properties of classical LQR are ensured even for the untrained policy.*

Up to this point, we established how to design the objective for the DRL agent and how to structure the policy to be learned. Hence, we are ready to present our solution to Problem 1.1 by invoking Lemma 1.1, Proposition 1.1 and by letting Assumption 1.1 hold.

Theorem 1.1. *Let be given any algorithm* $\mathbf{a} \in \mathbb{A}$*. Consider the cost* (1.11) *with the reward function* $\mathbf{r}$ *given by* (1.17)*. Then by selecting the control policy* $\boldsymbol{\pi}^\theta$ *as in* (1.19) *and determining its parameter vector* θ *via* $\mathbf{a}$*, Problem 1.1 is solved.*

Proof. The reward function defined in (1.17) allows us to use the discounted cost (1.11) without affecting the local solution $K^\star$ thanks to Lemma 1.1, and without restricting our choice of $\gamma \in (0, 1]$. Then (1.19) ensures constraint (1.10) is always satisfied via Proposition 1.1 and Assumption 1.1. Finally $\mathbf{a} \in \mathbb{A}$ allows the learning of the locally optimal policy parameters. □

The proposed solution is independent from the choice of the deep reinforcement learning algorithm and it can be applied to a wide variety of existing model-free solutions.

In what follows, we study the behavior of the proposed solution during the training process. We focus on the specific class actor-critic algorithms (see Appendix A.1), since many modern solutions exploit the actor-critic architecture. Moreover, as previously said, we assume the policy to be trained via policy gradient methods, which are a common choice in the DRL framework. For exploring the training behavior, we can combine the update law (A.3) and the policy (1.19). Plugging the equality (1.19) in the deterministic policy gradient equation (A.3) highlights that

$$\nabla_\theta \boldsymbol{\pi}^\theta(x) = h_1(x) \nabla_\theta \boldsymbol{\mu}^\theta(x), \tag{1.22}$$

being $\boldsymbol{\mu}^\theta(x_t)$ the only term depending on the parameters. Equation (1.22) shows that the closer the system gets to the equilibrium point the smaller the updates become, since $h_1 \in \mathcal{H}_1$. Note that one can substitute the policy (1.19) in the nonlinear model (1.9) and obtain a new system under the learned input $\boldsymbol{\mu}^\theta(x)$. Hence, the exploration noise which is typically added to deterministic policies can be applied directly on $\boldsymbol{\mu}^\theta(x)$. This reflects the fact that $\boldsymbol{\mu}^\theta$ is the only function to be learned. By doing so, $\boldsymbol{\pi}^\theta(x)$ is a random variable during training with

$$\begin{aligned} \mathbb{E}\left[\boldsymbol{\pi}^\theta(x)\right] &= (1 - h_1(x))K^\star x + h_1(x)\mathrm{E}\left[\boldsymbol{\mu}^\theta(x)\right], \\ \mathrm{Var}\left[\boldsymbol{\pi}^\theta(x)\right] &= h_1(x)^2 \mathrm{Var}\left[\boldsymbol{\mu}^\theta(x)\right], \end{aligned}$$

being $\boldsymbol{\mu}^\theta(x)$ the only random component. Since $h_1 \in \mathcal{H}_1$, the variance decreases the closer the system is to the equilibrium. The analysis shows that, in a neighborhood of the equilibrium point defined by h_1, the updates to the parameters take into account the reduced importance of the learned component to the received reward. Moreover, the explored actions focus around the near-optimal linear input. Finally, the study suggests that each training episode should be concluded once the equilibrium of the state-space (i.e. an arbitrarily small neighborhood of it) is reached. This is in accordance with the fact that the agent is actually learning only how to steer the system to the equilibrium point and not how to keep it there, being the solution to the latter problem already provided by the local controller $\boldsymbol{\pi}_{\mathrm{loc}}$. Note that the framework of additive exploration noise with respect to $\boldsymbol{\mu}^\theta$ instead of $\boldsymbol{\pi}^\theta$ mimics the blending between a stochastic policy trained with stochastic policy gradient [210] and a local policy. As such, the similar results hold for the class of stochastic policy gradient-based methods.

1.1.3 Improving learning by reshaping the value function estimation

For designing critics, model-free RL algorithms usually rely on value functions to drive the policy towards the optimal solution. Inspired by the structure of the control policy (1.19), we propose

a value function estimator built on the knowledge of its local behavior and of the linear model. Suppose we are interested in the *action*-value function under the control policy (1.19). Let us denote the its estimator by $\hat{\mathbf{Q}}^{\phi}_{\boldsymbol{\pi}}(x_{\mathfrak{t}}, u_{\mathfrak{t}})$. Due to the structure of $\boldsymbol{\pi}^{\theta}$, equation (1.5) provides a suitable local approximation. Consequently, by defining $z := \mathrm{col}(x, u)$ and the set $\mathcal{Z} := \{z \in \mathbb{R}^{n_x+n_u} : z = (0, u), \forall u \in \mathcal{U}\}$, we impose the following local constraint for all $z \in \mathcal{Z}$

$$D^{\mathfrak{i}}\hat{\mathbf{Q}}^{\phi}_{\boldsymbol{\pi}}(z) = D^{\mathfrak{i}}\mathbf{Q}^{\star}_{\gamma}(z),\ |\mathfrak{i}| \leqslant 2\,, \tag{1.23}$$

with $D^{\mathfrak{i}}$ denoting the derivative of order $\mathfrak{i}$ with the multi-index notation for multi-variable functions. Constraint (1.23) parallels (1.10) by imposing the action-value function estimation to be exactly (1.5) whenever $x = 0$. The idea comes from the fact that, in the origin, system (1.9) behaves as its linearization and $\boldsymbol{\pi}^{\theta}(0) = \boldsymbol{\pi}_{\mathrm{loc}}(0)$. As a consequence, (1.5) is the optimal action-value function. Then, similarly to the policy equation (1.19), the estimated action-value function $\hat{\mathbf{Q}}^{\phi}_{\boldsymbol{\pi}}(x, u) = \hat{\mathbf{Q}}^{\phi}_{\boldsymbol{\pi}}(z)$ for the nonlinear system (1.9) under (1.19) is modeled as

$$\hat{\mathbf{Q}}^{\phi}_{\boldsymbol{\pi}}(z) = \mathbf{Q}_{\mathrm{loc}}(z) + \hat{\mathbf{Q}}^{\phi}_{\mathrm{glo}}(z),\ \mathbf{Q}_{\mathrm{loc}}(z) = \mathbf{Q}^{\star}_{\gamma}(z),\ \hat{\mathbf{Q}}^{\phi}_{\mathrm{glo}}(z) = h_2(x)(\boldsymbol{\Omega}^{\phi}(z) - \mathbf{Q}_{\mathrm{loc}}(z)), \tag{1.24}$$

where $h_2 \in \mathcal{H}_2$, $\boldsymbol{\Omega}^{\phi} : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}$ is a parametrized function whose parameter vector $\phi \in \mathbb{R}^q$ is learned by the DRL agent and satisfying the following assumption:

Assumption 1.2. *The function $\boldsymbol{\Omega}^{\phi}$ is locally Lipschitz.*

As for the control policy, we justify the design choice (1.24) via the following Proposition, showing that (1.24) and Assumption 1.2 allow learning a generic C^3 function satisfying the local constraint (1.23).

Proposition 1.2. *The following statements hold.*

1. *For any C^3 function $\mathbf{Q}_{\mathrm{loc}} \in: \mathbb{R}^{n_x+1} \to \mathbb{R}$ and any function $\hat{\mathbf{Q}}^{\phi}_{\boldsymbol{\pi}} : \mathbb{R}^{n_x+1} \to \mathbb{R}, \hat{\mathbf{Q}}^{\phi}_{\boldsymbol{\pi}} \in C^3$ satisfying the local approximation constraint* (1.23) *for any $h_2 \in \mathcal{H}_2$, there always exists a locally Lipschitz function $\boldsymbol{\Omega}^{\phi} : \mathbb{R}^{n_x+1} \to \mathbb{R}$ satisfying the equality* (1.24) *for all $x \in \mathbb{R}^{n_x}, \forall u \in \mathcal{U} \subseteq \mathbb{R}$.*
2. *Let $\boldsymbol{\Omega}^{\phi}$ satisfy Assumption 1.2, $h_2 \in \mathcal{H}_2$ and $\hat{\mathbf{Q}}^{\phi}_{\boldsymbol{\pi}} : \mathbb{R}^{n+1} \to \mathbb{R}$ as in* (1.24)*. Then* (1.23) *is satisfied.*

Proof. The proof follows the same steps performed in the proof of Proposition 1.1. Note that (1.23) holds for all points in $\mathcal{Z}$ instead of the single point scenario of Proposition 1.1. Moreover, we match the second order approximation via $h_2 \in \mathcal{H}_2$. □

Algorithm 1 presents the procedure to implement the proposed solution with an actor-critic algorithm. It takes as input the undiscounted problem formulation, the linearized matrices, an actor-critic algorithm and a discount factor. Successively, it computes the linear optimal solution and the associated discounted problem and sets the saturation functions enforcing the local stability. Finally, it sets up the structured policy and value function(s) and runs the actor-critic algorithm for learning the parameters. The output is a learned optimal locally asymptotically stable deterministic policy.

1.2 Experimental results

In view of the results of Section 1.1, we test the proposed methodology for embedding local knowledge in actor-critic algorithms. The main objective is to show that the learned policy improves the local LQR in terms of domain of attraction (DOA) and the standard DNN controller

Algorithm 1 Algorithm for learning a Locally Asymptotically Stabilizing (LAS) control policy.

1: Input: (A, B), Q, R, γ, $\mathbf{a} \in \mathbb{A}$
2: Compute P, $K^\star$, $\mathbf{H}_\gamma$, Q_γ, R_γ
3: Pick $h_1 \in \mathcal{H}_1$, $h_2 \in \mathcal{H}_2$
4: Set $\mathbf{r}$ as in (1.17)
5: Set $\boldsymbol{\pi}^\theta$ as in (1.19) with parameters θ_0
6: Set $\hat{\mathbf{Q}}_{\boldsymbol{\pi}}^\phi$ as in (1.24) with parameters ϕ_0
7: Run $\mathbf{a}(\theta_0, \phi_0)$
8: Output: LAS control policy $\boldsymbol{\pi}^\theta$

in terms of robustness (provided by the local stability property). We recall that robustness in a total stability sense can be interpreted as generalization over a bounded, continuous set of parameters. This is a valuable property when dealing with DNN-based controllers. We propose two situations. First, we consider the simpler case of a known local model. Then, we address the scenario where only the desired equilibrium point is known. This framework is more similar to the one of standard DRL, where a model of the environment is not supposed to be available.

1.2.1 Known local model example

We run simulations in a frictionless inverted pendulum environment. The system nonlinear model is

$$\begin{cases} \alpha^+ = \alpha + \omega\,\Delta\mathrm{t}, \\ \omega^+ = \omega - \frac{3g}{2\ell}\sin(\alpha+\pi)\Delta\mathrm{t} + \frac{3}{2m\ell^2}(\mathrm{sat}(u)+d)\Delta\mathrm{t}, \end{cases}$$

where $\alpha \in [-\pi, \pi)$ is the angle between the position of the pendulum and the top vertical one, $\omega \in \mathbb{R}$ is its rate of change, $\mathrm{sat}(\cdot)$ is a saturation function limiting the control input torque u in $[-2, 2]$, d is wind disturbance affecting the system, $\Delta\mathrm{t} = 0.05$ is the discretization step, $g = 10$ is the approximated gravitational acceleration, $\ell = 1$ is the length of the pendulum and $m = 1$ is its mass. We denote $x := \mathrm{col}(\alpha, \omega)$ and we linearize the system around the unstable equilibrium $(x^\star, u^\star) = (0, 0)$, thus obtaining

$$A = \begin{pmatrix} 1 & \Delta\mathrm{t} \\ \frac{3g}{2\ell}\Delta\mathrm{t} & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ \frac{3}{2}\frac{\Delta\mathrm{t}}{m\ell^2} \end{pmatrix}.$$

The goal is to stabilize the pendulum at $(x^\star, u^\star)$, corresponding to the top vertical position, starting from any random initial condition. The parameters of the discounted LQR problem for the linearized system are

$$Q_\gamma = \begin{pmatrix} 1 & 0 \\ 0 & 0.1 \end{pmatrix},\ R_\gamma = 0.001, \quad \gamma = 0.99,\ K^\star = \begin{pmatrix} -19.3006 & -5.9918 \end{pmatrix},\ P = \begin{pmatrix} 8.088 & 0.4782 \\ 0.4782 & 0.1624 \end{pmatrix}.$$

Due to input saturation, the only solution to the problem is to learn how to swing in order to gain momentum. Clearly, the standard solution to an LQR problem is not describing such a control behavior. We train the proposed policy using an off-the-shelf DRL algorithm, namely TD3 [80]. The nominal version of the learning algorithm comes from Stable Baselines 3 library [176] and it is adapted to include our policy (1.19) and value function (1.24), following the steps in Algorithm 1. Since most of DRL algorithms are designed to maximize the expected reward, we simply invert the sign of $\mathbf{r}(x, u)$ and $\mathbf{Q}_{\mathrm{loc}}(x, u)$. Each training episode is stopped as soon as the state enters a small ball of radius 10^{-5} centered in the origin, or their time limit is reached. We use

the functions $h_1(x) = \tanh(\tanh^{-1}(0.99)\frac{x^\top P x}{c})$ and $h_2(x) = \tanh(\tanh^{-1}(0.99)(\frac{x^\top P x}{c})^{\frac{3}{2}})$, which saturate outside the Lyapunov level set $\mathcal{V}_c(V) := \{V(x) = c\}$, with $c \in \mathbb{R}_{>0}$. The value of c is estimated by sampling random initial conditions and testing the convergence to the equilibrium point, see Figure 1.2a, and we select $c = 0.47$.

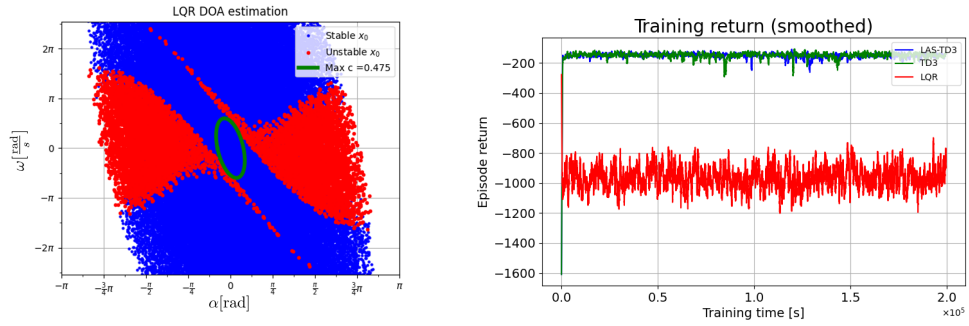
We evaluate the performances and stability of our solution by comparing the results with the simple LQR and the nominal version of the algorithm. Standard TD3 is trained with the same reward function (1.17). In order to allow good training, episodes are not stopped as soon as a neighborhood of the origin is reached. They are stopped only once the episode length limit has been reached. We use the same hyperparameters for both the standard and the locally asymptotically stable version of the algorithm: 3-layers fully-connected DNNs with ReLU activation functions, 64 units for the first hidden layer and 32 units as the second one, both for the critic and for the actor, and a learning rate of $\lambda = 0.00371$.

We first analyze the learning performances. During training, we remove the disturbance wind by setting a constant $d = 0$, and the initial condition is randomly sampled. From Figure 1.2b we can infer that classical LQR succeeds only if the initial state lies inside it's domain of attraction. Its performances strongly fluctuates, since the magnitude of the return of an episode (i.e. the final cost of the episode) is very big (bad) if the initial condition lies outside of its domain of attraction or quite small (good) if the system starts close to the equilibrium. On the other hand, our solution behaves comparably to the nominal TD3 algorithm.

We also study the stability of the closed loop system. During training, we periodically evaluate the policy by running an experiment in a different environment corrupted by uncertainties. We simulate measurement noise $u = \boldsymbol{\pi}^\theta(x + w)$, $w \sim \mathcal{N}(0, 0.03)$ and mass mismatch $m_{\text{real}} = 1.2 m_{\text{train}}$. Moreover, the input is perturbed by external sinusoidal wind $d = 0.36 \sin(\frac{\pi}{50}\mathrm{t})$. A "stability score" is extracted as the maximum of the norm of the error vector at steady state, $\mathrm{s} := \max_{\mathrm{t}\in[T_{ss}, T_{ep}]} |x(\mathrm{t}) - x^\star|$, with T_{ep} the maximum episode length and steady-state time $T_{ss} = 0.8\, T_{ep}$. We run the stability evaluation episodes twice, at first with initial condition $x_0 = (0.945\pi, 0)^\top$, close to the position "down" of the pendulum, then with $x_0 = (0, 0)^\top$, corresponding to the desired equilibrium point. Figure 1.3 clearly shows that the DRL approach focuses solely on performances, as shown by the fact that more training steps do not imply a more robust policy. For example, the norm of the steady-state error of the policy trained for approximately 0.8×10^5 simulated seconds is much worse then the one of the policy trained for approximately 0.6×10^5 simulated seconds, independently from the initial condition. Differently, the scores for the modified algorithm remain fairly constant throughout training. It can be clearly seen that the local behavior expressed when starting in the equilibrium point is comparable to the one of the classical LQR policy. However, due to the learning procedure, also the global behavior converges to a stable one. This shows that the learning component reliably learned how to drive the system sufficiently close to the desired equilibrium for the local component to guarantee stability. However we note that, as $h_1(x) \to 1$, the learned component becomes predominant, possibly affecting stability.

1.2.2 Unknown local model algorithm

Aiming at getting closer to the classical reinforcement learning framework, we now provide an end-to-end algorithm addressing the case of completely unknown plant, along with practical experimental validations. We are interested in showing the proposed approach still provides robust policies equipped with local stability guarantees, even when we have no access to the local dynamics information. We will use simple techniques, adapting classical results from system



(a) Estimated level set for saturated LQR (b) Episode return during training

Figure 1.2: Domain of attraction and training returns

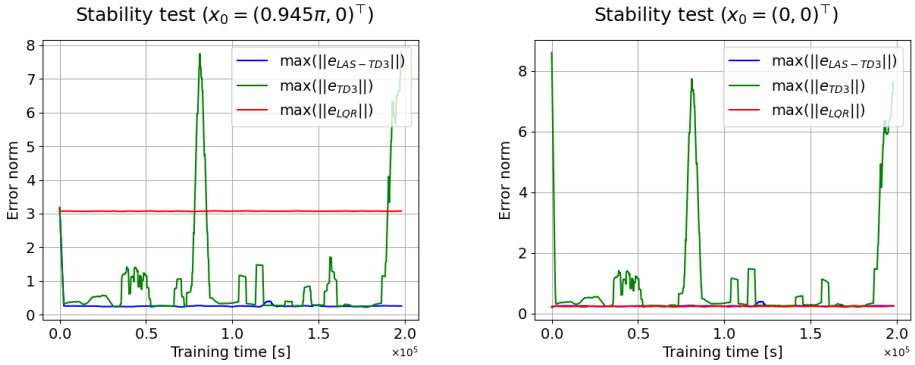


Figure 1.3: Stability of policies during training (smoothed)

identification and machine learning. These choices are mostly based on convenience and, given the agnostic nature of the methodology, each step can be adapted with appropriate techniques, given the task at hand. Hence, these experiments are aimed at emphasizing the importance of enforcing stability properties in learned controllers to gain generalization and robustness. We conduct experiments against standard deep actor-critic algorithms, specifically PPO [188], DDPG [129], SAC [94], and TD3 [80], over three benchmarks from DRL literature.

To extend the approach, we start by assuming the desired equilibrium state $x^\star$ is known. Then, we identify three required quantities: the steady-state input $u^\star$, the local model matrices A, B, and the domain of attraction. Since we now consider arbitrary equilibrium pairs $(x^\star, u^\star)$, we need to perform the change of coordinates $\tilde{x} := x - x^\star$, $\tilde{u} := u - u^\star$ and $\tilde{z} := z - z^\star$, with $z := \text{col}(x, u)$ and $z^\star := \text{col}(x^\star, u^\star)$. Then, the related reward function (1.11) becomes $\mathbf{r}(x, u) := \tilde{x}^\top Q_\gamma \tilde{x} + \tilde{u}^\top R_\gamma \tilde{u}$. Similarly, the policy (1.19) and action-value estimator (1.24) are modified as

$$\boldsymbol{\pi}^\theta(x) = \boldsymbol{\pi}_{\text{loc}}(x) + \boldsymbol{\pi}^\theta_{\text{glo}}(x),\ \boldsymbol{\pi}_{\text{loc}}(x) = u^\star + K^\star \tilde{x},\ \boldsymbol{\pi}^\theta_{\text{glo}}(x) = h_1(\tilde{x}) \left(\boldsymbol{\mu}^\theta(x) - \boldsymbol{\pi}_{\text{loc}}(x) \right), \quad \text{(1.25a)}$$

$$\hat{\mathbf{Q}}^\phi_{\boldsymbol{\pi}}(z) = \mathbf{Q}_{\text{loc}}(z) + \hat{\mathbf{Q}}^\phi_{\text{glo}}(z),\ \mathbf{Q}_{\text{loc}}(z) = \mathbf{Q}^\star_\gamma(\tilde{z}),\ \hat{\mathbf{Q}}^\phi_{\text{glo}}(z) = h_2(\tilde{x})(\boldsymbol{\Omega}^\phi(z) - \mathbf{Q}_{\text{loc}}(z)). \quad \text{(1.25b)}$$

We now present some simple techniques used to estimate the unknown quantities. First, without any model knowledge, we need to estimate which input $u^\star$ would make the pair $(x^\star, u^\star)$ an equilibrium for the closed-loop system. Mathematically, we look for $u^\star$ such that $x^\star = f(x^\star, u^\star)$.

Algorithm 2 Linearized system identification

Input: $x^\star, \hat{u}^\star, b, N_e$;
Sample $u_i \sim \mathcal{N}(\widehat{u^\star}, \sigma^2)$, $i = 0, \dots, N_e - 1$;
Initialize the system in $x(0) = x^\star$;
for $\mathfrak{t} = 0, \dots, N_e - 1$ **do**
  Apply $u_\mathfrak{t}$ and observe $x_\mathfrak{t}^+ = f(x(\mathfrak{t}), u_\mathfrak{t})$;
  if $|x_\mathfrak{t}^{+j} - x^{\star j}| < b^j$ for all $j = 1, \dots, n$ **then**
    $x(\mathfrak{t}+1) = x_\mathfrak{t}^+$;
  else
    Reset the system in $x(\mathfrak{t}+1) = x^\star$;
    Store $(x(\mathfrak{t}), u_\mathfrak{t}, x_\mathfrak{t}^+)$;
  end if
end for
Compute $\widehat{M}$ with least squares;
Extract $\widehat{A}$ and $\widehat{B}$ as in (1.27);

Then, the estimated input $\widehat{u^\star}$ can be defined as the one minimizing the error between $x^\star$ and the successive one, which is observed once such input is applied. Hence, we aim at solving

$$J(x, u) := |x^+ - x|^2, \quad \widehat{u^\star} = \arg\min_u J(x^\star, u). \tag{1.26}$$

There are multiple possibilities to solve such an optimization problem, e.g., automatic differentiation tools. In our experiments, we use a genetic algorithm to find an initial estimate and successively refine it via gradient descent until the computed cost falls below a small threshold $\epsilon_u > 0$. Once $u^\star$ has been estimated, we focus our attention on the local model estimation. As the model is linear, given a dataset of tuples (x, u, x^+), we can rely on classical system identification techniques [202]. Indeed, a discrete-time linear time-invariant system of the form (1.1) can be rewritten as

$$x^+ = Mz, \quad M = \begin{pmatrix} A & B \end{pmatrix}, \tag{1.27}$$

and the system matrix estimation $\widehat{M}$ can be obtained by linear least square methods [202]. However, due to the nonlinear behavior of the plant, some care has to be taken during the data collection procedure in order to obtain a suitable local model. Since we look for a linearization of the system around the equilibrium, we sample $N_e \in \mathbb{N}$ random control actions u_i from a normal distribution $u_i \sim \mathcal{N}(\widehat{u^\star}, \sigma^2)$, $i = 0, \dots, N_e - 1$, where $\sigma > 0$ is a sufficiently small standard deviation. For a given index $i \in \mathbb{N}$, we introduce the variable $x_i^+ = f(x_i, u_i)$. We collect system trajectories by initializing its state in $x^\star$, i.e. $x(0) = x^\star$. Then, the control action u_0 is applied, the next state x_0^+ is sampled and the triplet $(x(0), u_0, x_0^+)$ is stored. Before applying the successive action u_1, the trajectory is stopped and the system is initialized back in the equilibrium state $x(1) = x^\star$ if the new state x_0^+ lies outside of a box centered in the equilibrium state $\mathbb{B}(x^\star) = \{x \in \mathbb{R}^{n_x} : |x^j - x^{\star j}| < b^j,\ j = 1, \dots, n\}$, where superscript j refers to the j^{th} component of the vector and $b \in \mathbb{R}^{n_x}$ is a vector of positive bounds. Otherwise, we directly apply the next input, i.e. $x(1) = x_0^+$. The process is repeated until N_e state-action-state samples are collected. As previously stated, once the data has been collected, the system matrix $\widehat{M}$ can be obtained by least square estimation [202]. Algorithm 2 presents the proposed procedure for linear model estimation given a setpoint and a vector of positive state bounds $b \in \mathbb{R}^{n_x}$.

Once the linear model has been estimated, the local solution $\boldsymbol{\pi}_{\text{loc}}$ in (1.19) can be computed. Yet, since the size of the region where $\boldsymbol{\pi}_{\text{loc}}$ stabilizes system (1.9) is still unknown, we propose

a simple automatic procedure for obtaining a conservative estimation of its domain of attraction so that to set the shape of the blending functions h_1, h_2. There is a rich literature on domain of attraction estimation, see e.g., [44, 220]. Our approach exploits the solution to the Discrete-time Algebraic Riccati Equation (DARE) and allows for a direct definition of suitable functions h_1, h_2. By means of Lyapunov inequalities, local asymptotic stability of an equilibrium point for the closed-loop system can be characterized via the existence of a local Lyapunov function $V : \mathbb{R}^{n_x} \to \mathbb{R}$, see e.g., [30]. Such a function has to satisfy the following conditions

$$V(x^\star) = 0, \tag{1.28}$$
$$V(x) > 0 \quad \forall x \in \mathbb{R}^{n_x} - \{x^\star\}, \tag{1.29}$$
$$V(x') - V(x) < 0 \quad \forall x \in \mathbb{R}^{n_x} - \{x^\star\}. \tag{1.30}$$

Furthermore, since $\boldsymbol{\pi}_{\text{loc}}$ is the linear optimal policy, we can choose the optimal value function in the linear framework as a candidate local Lyapunov function in a quadratic form, namely

$$V(x) = (x - x^\star)^\top P (x - x^\star), \tag{1.31}$$

where P is the positive definite solution of the DARE, see e.g. [28]. Clearly, conditions (1.28) and (1.29) are met by such a function by construction, hence it suffices to check whether (1.30) is satisfied at different points. Then, the domain of attraction is set as the states inside an ellipsoid described by (1.31) such that they all satisfy property (1.30). Practically, this size is approximately estimated by sampling $N_v \in \mathbb{N}$ random points on the surface of an increasingly bigger n-dimensional ellipsoids $\mathcal{E}(c_k) := \{x \in \mathbb{R}^{n_x} : V(x) \leqslant c_k\}$. Then, for each of these points, we check if the decrease condition (1.30) is satisfied by selecting actions with $\boldsymbol{\pi}_{\text{loc}}$ on the nonlinear system. We repeat the procedure for incremental values $c_{k+1} = c_k + \Delta_c$, $\Delta_c \in \mathbb{R}_{>0}$ until we find a sample violating the constraint (1.30). Then we set the estimated size $\widehat{c^\star} = c_k$. Note that, by continuity, a sufficiently fine sampling of the surface of the ellipsoid allows the generalization of the result to the complete surface. The same can be stated for sufficiently small Δ_c. Once $\widehat{c^\star}$ is known, we can define a normalized Lyapunov function $\nu(x) := V(x)/\widehat{c^\star}$. This is useful to easily identify the states at the borders of the domain of attraction. Indeed, it is smaller than 1 inside the estimated domain, larger outside and 1 on the borders. Then, as hinted in the previous experiments, the saturation functions h_1, h_2 in (1.19) and (1.24) can be shaped as

$$h_1(x) = \tanh(\alpha\, \nu(x)), \quad h_2(x) = \tanh\left(\alpha\, \nu(x)^{3/2}\right), \tag{1.32}$$

where $\alpha := \tanh^{-1}(\beta)$ and $\beta \in (0, 1)$ set the value of h_1,h_2 at points on the boundaries of the level set identified by $\widehat{c^\star}$. Note that equations (1.32) satisfy the requirements presented in (1.18) of being positive smooth, saturating, higher order functions with respect to $\boldsymbol{\pi}_{\text{loc}}$ and $\mathbf{Q}_{\text{loc}}$ respectively. In other words, they guarantee the control policy and the action value behave as the local quadratic optimal ones close to the equilibrium, and as the learned one far from it. Moreover, the saturating action can be controlled by the value of β. If β is very small, at the borders of the estimated domain the control policy will still be dominated by the local solution $K^\star$. On the contrary, if $\beta \approx 1$ we let the learned component modify the solution even inside the estimated domain. Algorithm 3 proposes the routine for estimating the domain of attraction of the local stabilizing policy. Note that the method presented in this section provides an estimation of the stable region which may be very conservative. Indeed, it looks for the smaller invariant ellipsoid contained in the domain. A less conservative approach may be to set a threshold of allowed failures per level set. This can be controlled by setting a value $\delta \in [0, 1)$ different from

Algorithm 3 Local domain of attraction estimation

Input: $x^\star, \widehat{u^\star}, \Delta_c, N_v, \delta$;
$(\widehat{c^\star}, c, \texttt{n\_fails}) \leftarrow (0, \Delta_c, 0)$;
while $\texttt{n\_fails} < \texttt{int}(\delta N_v) + 1$ **do**
  Reset $\texttt{n\_fails} = 0$;
  Sample x_i, $i = 1, \dots, N_v$ on $\mathcal{E}(c)$;
  for i=$1, \dots, N_v$ **do**
    Simulate $x_i^+ = f(x_i, \boldsymbol{\pi}_{\text{loc}}(x_i))$;
    if $V(x_i^+) - V(x_i) \geqslant 0$ **then**
      $\texttt{n\_fails} = \texttt{n\_fails} + 1$;
    end if
    if $\texttt{n\_fails} > \texttt{int}(\delta N_v)$ **then**
      $\widehat{c^\star} = c$;
      break;
    end if
  end for
  Update $c \leftarrow c + \Delta_c$;
end while

zero in Algorithm 3. However, the bigger δ, the less confident we can be in the estimation.

Once u^*, A, B have been estimated and h_1, h_2 have been selected, the last step is to select an actor-critic DRL algorithm to learn the networks' parameters. Thus, the complete algorithm runs in two consequent steps: unknown quantities estimation and policy learning. We recall that since the local policy is locally exponentially stabilizing, it is robust with respect to bounded errors in the model and stability is guaranteed even if the learned model is not perfect, see Chapter 2. Therefore, due to the robustness of the local solution $K^\star$, the control policy $\boldsymbol{\pi}^\theta$ is still ensured to be locally stabilizing as long as the estimated model provides a good approximation of the linearized behavior. This reflects in the choice of bounds b defining the box around the equilibrium used for the local model estimation. Indeed, they should not describe too big of a region around the goal state $x^\star$. Moreover, they may be dynamically changed if the resulting model does not fit sufficiently well the collected trajectories. We also highlight that intertwining the phases of model estimation and policy learning may cause the loss of the stability properties. Indeed, even switching between stable linear systems may cause instability [127, Section 2.4.1]. Therefore, once the local stabilizing policy is fixed, we only allow fine-tuning of the overall policy with the components learned by the DRL algorithm, which can still affect (slightly) the local behavior. Indeed, we stress that the overall policy perfectly matches the local LQR policy only at the equilibrium.

We can finally present the complete algorithm, whose pseudocode is presented in Algorithm 4. The algorithm asks for the desired equilibrium state $x^\star$ and some hyperparameters: the vector of bounds b, the number of estimation samples N_e, the number of samples per level set N_v, the step size Δ_c and the value β. It first estimates the steady-state input by solving the optimization problem (1.26). Then it estimates the linearized model. Once $\widehat{A}$ and $\widehat{B}$ are known, the local optimal policy $\boldsymbol{\pi}_{\text{loc}}$ is computed and its domain of attraction is estimated. Once the local problem has been solved, it sets the functions h_1 and h_2 as in (1.32) and trains the component $\boldsymbol{\mu}^\theta$. Given the undiscounted reward matrices Q, R describing the local quadratic optimization problem, it maps them in the discounted framework based on the discount factor γ. Then, given an actor-critic DRL algorithm, it outputs a near-optimal neural network policy guaranteeing local asymptotic stability of the closed-loop.

1.2.3 Unknown local model experiments

Similarly to the example in Section 1.2.1, we analyze multiple aspects of the proposed solution. First, we compare performances of standard DRL algorithms against those from their locally

Algorithm 4 Locally asymptotically stabilizing actor critic DRL (unknown model)

Input: $x^\star, N_e, b, \Delta_c, N_v, Q, R, \gamma, \beta$;
Estimate $\widehat{u^\star}$ by solving (1.26);
Obtain $\widehat{A}$ and $\widehat{B}$ as in Algorithm 2 with $\widehat{u^\star}$;
Compute P, $K^\star$, H with $\widehat{A}$ and $\widehat{B}$ as in (1.8), (1.7);
Set $\boldsymbol{\pi}_{\text{loc}}$ as in (1.25a) with $\widehat{u^\star}$ and $K^\star$;
Obtain $\widehat{c^\star}$ as in Algorithm 3;
Set h_1, h_2 as in (1.32) via $\widehat{c^\star}$;
Set $\boldsymbol{\pi}^\theta$ as in (1.19);
Set $\hat{\mathbf{Q}}^\phi_{\boldsymbol{\pi}}$ as in (1.24);
Compute Q_γ, R_γ with P and γ as in (1.12);
Train $\boldsymbol{\pi}^\theta$ and $\hat{\mathbf{Q}}^\phi_{\boldsymbol{\pi}}$ with $\mathbf{a} \in \mathbb{A}$ and reward (1.11);

asymptotically stabilizing variants (LAS policy). Next, we empirically study the stability properties of both family of algorithms. Then, we compare the difference in domain of attraction size between the LAS solution and the classical local control-theoretic approach (LQR). Finally, we also present experiments showing that the robustness properties generalize well to different instances.

We train four different DRL algorithms, namely DDPG [129], PPO [188], TD3 [80] and SAC [95], alongside their LAS versions. We train each algorithm over four random seeds and the training is performed on a single NVidia GTX Titan X with 12 GB RAM. As in Section 1.2.1, the standard formulation of the algorithms comes from Stable Baselines 3 [176] and it is modified for obtaining the LAS design. For the estimation routine we set $N_e = 3 \times 10^5$ and $N_v = 5 \times 10^3$. Table 1.3 presents the main hyperparameters used for each algorithm and control task. Most of them are provided by the Stable Baselines 3 library and they are not optimized, since the main goal is to show the added value of locally stabilizing policies, especially in non-optimal conditions. All the hyperparameters that are not shown in the table are also set as the default ones. For all the algorithms, we use a discount factor $\gamma = 0.99$, in order to provide a sufficiently big virtual time horizon to the optimization process.

Control tasks. We set up three different environments of increasing complexity for evaluating algorithms. Similarly to Section 1.2.1, each algorithm is trained on an ideal version of the environment, where noise and external disturbances are not included. Then, they are tested on the "corrupted" version of the environment, where parameter mismatches, measurement noise and sinusoidal disturbances perturb the task. In particular, each environment will evolve according to discrete-time nonlinear dynamics of the form

$$x^+ = f(\eta, x, \text{sat}_{[\underline{u},\bar{u}]}(\psi(x + w)) + d),$$

where $\psi : \mathbb{R}^{n_x} \to \mathbb{R}$ is an arbitrary policy (e.g. $\boldsymbol{\pi}^\theta$), $\text{sat} : \mathbb{R} \to \mathbb{R}$ is a hard saturation function clipping the control input defined as $\text{sat}_{[\underline{u},\bar{u}]}(\cdot) := \max(\min(\cdot, \bar{u}), \underline{u})$, η is a vector of constant system parameters, w is a random noise sample and d is an external input disturbance. During training w, d are identically 0 and the environment parameters are slightly different from the ones used at test time, namely $\eta_{\text{test}} \neq \eta_{\text{train}}$. The metric for assessing stability is defined as the maximum error at steady state, i.e. after some time we start looking at the maximum norm of the error with respect to the goal state. This "stability cost" can be described as $s_\psi = \max_{t \in [T_{ss}, T_{ep}]} |x - x^\star|$ where $T_{ep} \in \mathbb{N}$ represents the maximum episode length in time steps and $T_{ss} \in [0, T_{ep})$ is the

instant at which we start evaluating the stability properties. Following Algorithm 4, the global discounted objective is designed starting from a local undiscounted one. This ensures the local solution is stabilizing. Moreover, it allows standard DRL algorithm trained on such an objective to possibly converge to a locally stabilizing policy. For the local component, we use undiscounted quadratic costs described by a diagonal matrix $Q \in \mathbb{S}^{n_x}_{\succ 0}$ and a scalar $R > 0$. Given that typical DRL algorithms solve a maximization problem, we invert the sign of the instantaneous cost and of the local component of the critic. We now briefly present the three control tasks.

Pendulum swing-up (PSU). This task mimics the one proposed in Section 1.2.1. A frictionless pendulum has to be stabilized at a small angle from the unstable vertical position with limited input torque. The environment follows the same dynamics as OpenAI Gym [35] pendulum swing-up. The state vector is composed of the error angle with respect to the top vertical position and its rate of change ($n_x = 2$). The goal state is $x^\star = (\frac{\pi}{18}, 0)^\top$ and the initial condition is $x_0 = (\pi, 0)^\top$. Table 1.4 presents the environment parameters. For the PSU task, we use cost matrices $Q = \mathrm{diag}(1, 0.1),\ R = 0.001$.

Inverted pendulum swing-up (IPSU). A pendulum linked to a cart has to be stabilized in its unstable vertical position starting from the resting downward position. The environment is built using Pybullet [52]. The cart can slide on a rail of limited length and the input force on the cart is bounded. The cart is also required to reach the middle of the rail. The state vector is composed of the distance between the cart and the center of the rail, the error angle with respect to the top vertical position of the pole, the cart velocity and the angle rate of change ($n_x = 4$). The goal state is $x^\star = (0, 0, 0, 0)^\top$ and the initial condition is $x_0 = (0, \pi, 0, 0)^\top$. In Table 1.6 we report the environment parameters. For the IPSU task, we set $Q = \mathrm{diag}(0.1, 0.1, 0.01, 0.01)\ R = 0.01$.

Double inverted pendulum swing-up (DIPSU). The IPSU task has to be solved with a 2-link pendulum. The state vector is composed of the distance between the cart and the center of the rail, the error angle with respect to the top vertical position of the first pole, the error angle of the second pole with respect to the first pole's direction, the cart velocity and the angles rates of change ($n_x = 6$). The goal is to stabilize the system at $x^\star = (0, 0, 0, 0, 0, 0)^\top$ and the initial condition is $x_0 = (0, \pi, 0, 0, 0, 0)^\top$. The environment parameters are summarized in Table 1.5. Finally, the optimization objective of DIPSU is described by matrices $Q = \mathrm{diag}(0.001, 0.3, 0.3, 0.001, 0.01, 0.01),\ R = 0.0001$.

Results. Results are shown in Figures 1.4,1.5, 1.6 and Tables 1.1, 1.2, 1.7. We start by discussing the performances of the proposed algorithm with respect to the standard ones. The objective is to show that the addition of a locally stabilizing layer to the policy does not deteriorate performances of standard DRL solutions. The difference in performances can be clearly seen by looking at the episode return during training. From Table 1.1 and Fig. 1.4, we see that the LAS variants perform comparably to their standard versions. As expected, the approach embeds the local exact solution in the neural policy, thus it does not impact the performances once the $\boldsymbol{\pi}_{\mathrm{loc}}$ component is dominant. Moreover, in some instances the LAS policy performs even better than the native algorithm. This is probably due to the fact that the local component guides the exploration towards stabilizing actions. Since quadratic costs are proportional to the norm of the error, stabilizing actions encourage the reduction of the instantaneous cost.

We continue by assessing the stability properties of the closed-loop for both types of policies, i.e., standard and LAS policies. Table 1.2 presents the stability cost of each algorithm after training. It shows that LAS policies outperform the standard version in the local context, i.e., when starting sufficiently close to the equilibrium. Note also the reliability of such a local behavior, as shown by the small variation of the stability cost over different instances. As expected, this behavior is

	PPO		SAC	
	LAS	**Standard**	**LAS**	**Standard**
PSU	$-160,2 \pm 97,0$	$-162,4 \pm 91,9$	$-133,7 \pm 62,4$	$-133,9 \pm 77,5$
IPSU	$-142,4 \pm 35,8$	$-104,8 \pm 21,7$	$-92,3 \pm 1,8$	$-102,0 \pm 8,4$
DIPSU	$-579,1 \pm 156,3$	$-1706,9 \pm 481,2$	$-245,4 \pm 5,8$	$-447,1 \pm 5,9$

	TD3		DDPG	
	LAS	**Standard**	**LAS**	**Standard**
PSU	$-146,4 \pm 72,1$	$-135,5 \pm 101,4$	$-135,1 \pm 63,9$	$-156,8 \pm 77,3$
IPSU	$-99,0 \pm 7,3$	$-99,7,0 \pm 3,7$	$-120,4 \pm 29,0$	$-155,6 \pm 28,1$
DIPSU	$-251,3 \pm 5,7$	$-255,3 \pm 7,8$	$-487,4 \pm 287,4$	$-485,1 \pm 196,0$

Table 1.1: Mean evaluation rewards and standard deviations across ten trials (the higher the better).

	PPO		SAC		TD3		DDPG	
	LAS	**Standard**	**LAS**	**Standard**	**LAS**	**Standard**	**LAS**	**Standard**
PSU_1	$0,18 \pm 0,02$	$8,63 \pm 0.2$	$0,1 \pm 0,01$	$0,25 \pm 0,02$	$0,1 \pm 0,01$	$7,37 \pm 0,06$	$0,09 \pm 0,01$	$0,25 \pm 0,01$
PSU_2	$0,17 \pm 0,02$	$8,64 \pm 0,1$	$0,1 \pm 0,01$	$0,27 \pm 0,03$	$0,1 \pm 0,01$	$7,42 \pm 0,03$	$0,09 \pm 0,01$	$0,24 \pm 0,01$
$IPSU_1$	$0,69 \pm 0,05$	$1,95 \pm 2,76$	$0,80 \pm 0,12$	$1,07 \pm 0,25$	$0,76 \pm 0,06$	$0,92 \pm 0,10$	$0,66 \pm 0,04$	$3,27 \pm 2,08$
$IPSU_2$	$3,58 \pm 3,99$	$7,16 \pm 0,20$	$0,81 \pm 0,08$	$1,18 \pm 0,39$	$0,86 \pm 0,17$	$2,46 \pm 3,32$	$0,65 \pm 0,05$	$5,15 \pm 3,09$
$DIPSU_1$	$2,28 \pm 0,11$	$11,88 \pm 4,73$	$2,29 \pm 0,27$	$7,04 \pm 4,07$	$2,37 \pm 0,19$	$9,06 \pm 0,13$	$2,40 \pm 0,12$	$3,27 \pm 0,25$
$DIPSU_2$	$6,58 \pm 5,55$	$15,17 \pm 8,77$	$2,47 \pm 0,12$	$3,92 \pm 3,60$	$2,24 \pm 0,16$	$8,97 \pm 0,07$	$2,83 \pm 0,97$	$3,26 \pm 1,60$

Table 1.2: Maximum steady-state error norm in corrupted environments over ten trials (the lower the better). Subscript '1' indicates the system is initiated in $x^\star$. Subscript '2' indicates the system is initiated in x_0.

guaranteed by the local stabilizing policy. Differently, standard algorithms may not converge to a stabilizing behavior, due to a possibly poor approximation of the exact local optimal solution. The smoothed evolution of the stability cost during training is shown in Fig. 1.5. Again, we note from the plots of Fig. 1.5, that the standard algorithms are not capable of locally stabilizing the system in the corrupted environments, due to low diversity of the dataset. On the contrary, the LAS version shows a small error norm even with the untrained parameters, thanks to the local knowledge. We also emphasize that the local solution maintains a small error norm if the initial condition lies outside the domain of attraction of $\boldsymbol{\pi}_{\text{loc}}$ but the learned component manages to steer the system sufficiently close to the setpoint before T_{ss}, as shown by Table 1.2 and Fig. 1.5 (left). This does not always happen, see Fig. 1.5 (center) and Fig. 1.5 (right), and the LAS variants behave similarly to their native versions. Nevertheless, the local policy seems to improve the stability cost also in the global context. Finally, we stress that local stability of the closed-loop is preserved even if the algorithm does not reach a satisfactory solution, as for PPO in Fig. 1.4 (right) and Fig. 1.5 (right).

We now study whether standard DRL algorithms can achieve local asymptotic stability with an increased amount of data. We focus on the simplest task, the PSU. The standard algorithms are trained on three instances with different masses, while the LAS version is trained on a single one with the default mass $m = 1$. Hence, we are providing a richer dataset to the nominal algorithms to show that local generalization is obtained with small data overhead with respect to standard techniques. We set the corrupted environment mass to $m_{corr} = 1.2$ and we explore two training

	PSU			**IPSU**			**DIPSU**		
	lr	**af**	**NN**	**lr**	**af**	**NN**	**lr**	**af**	**NN**
DDPG	0.001	ReLU	400,300	0.001	ReLU	400,300	0.0001	ReLU	400,300
LAS-DDPG	0.001	ReLU	400,300	0.001	ReLU	400,300	0.0001	Tanh	400,300
PPO	0.003	Tanh	64,64	0.00025	Tanh	256,256	0.0001	Tanh	64,64
LAS-PPO	0.003	Tanh	64,64	0.00025	Tanh	256,256	0.0001	Tanh	64,64
TD3	0.001	ReLU	400,300	0.001	ReLU	400,300	0.0006	ReLU	400,300
LAS-TD3	0.001	ReLU	400,300	0.001	ReLU	400,300	0.0002	Tanh	256,256
SAC	0.001	ReLU	256,256	0.001	ReLU	256,256	0.001	ReLU	256,256
LAS-SAC	0.001	ReLU	256,256	0.001	ReLU	256,256	0.0001	Tanh	256,256

Table 1.3: Hyperparameters. **lr**: learning rate, **af**: activation function, **NN**: hidden layer sizes

situations. First, we present the results of experiments where the m_{corr} is between the smallest and biggest values experienced by the algorithms, see Fig. 1.6 (left). Then, we show the results for the case when the corrupted mass lies outside these boundaries, see Fig. 1.6 (right). We note that in both cases a locally stable behavior is not guaranteed by the standard algorithms, even when a higher amount of information is provided. This is particularly evident when the corrupted mass is bigger then the maximum experienced one, see Fig. 1.6 (right). Nevertheless, we can notice an improvement with respect to the single environment scenario, especially for PPO. On the other hand, the LAS approach ensures satisfactory generalization capabilities over a bounded, continuous set of parameters' variations around the nominal ones, even if trained on a single environment. The continuity of such a set and the reduced amount of required data further motivate the interest in the LAS approaches.

Finally, we evaluate the size of the domain of attraction of the composite policy and compare it to the linear control-theoretic solution (LQR). Hence, we uniformly sample $N_i \in \mathbb{N}$ random initial conditions sampled inside a box $\mathbb{B}(0)$ defined by the bounds vector $b_0 \in \mathbb{R}^{n_x}$. Then, we run experiments in the uncorrupted environment, once with the trained LAS neural policy and once with the LQR. Given an initial condition x and an episode length $T_{ep} > 0$, the experiment is considered a success if the error between the final state and the goal state is smaller than a small threshold $\epsilon > 0$. An evaluation metric $\mathsf{p} \in [0, 1]$ is given as a rate between the number of successes and the number of initial conditions, namely

$$\mathsf{p} = \frac{\texttt{n\_successes}}{N_i} = \frac{1}{N_i} \sum_{k=0}^{N_i} \mathfrak{s}(x_k(T_{ep})),$$

where the k superscript identifies the experiment initialized with the k^{th} initial condition and $\mathfrak{s} : \mathbb{R}^{n_x} \to \mathbb{R}$ is defined as

$$\mathfrak{s}(x) = \begin{cases} 1 & \text{if } |x - x^\star| \leqslant \epsilon \\ 0 & \text{otherwise} \end{cases}$$

Table 1.7 shows that for each task the neural solution presents a bigger rate of success than the local LQR. Hence, the size of the domain of attraction is improved by the learned component in comparison to the control-theoretic linearization technique. It shows we can safely embed and train a neural component into classical control-theoretic solutions without impairing their local properties. Moreover, it highlights the effectiveness of modern learning approaches in improving established control methods.

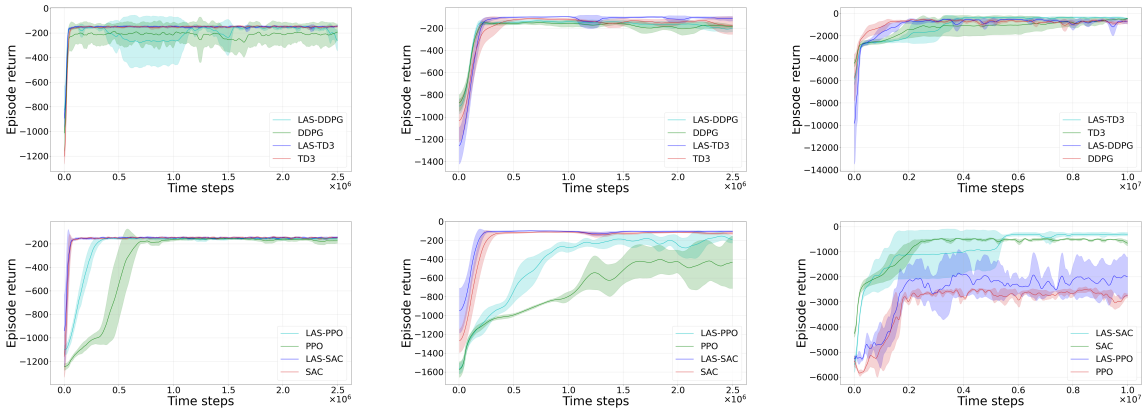


Figure 1.4: Training return: PSU (left), IPSU (center), DIPSU (right).

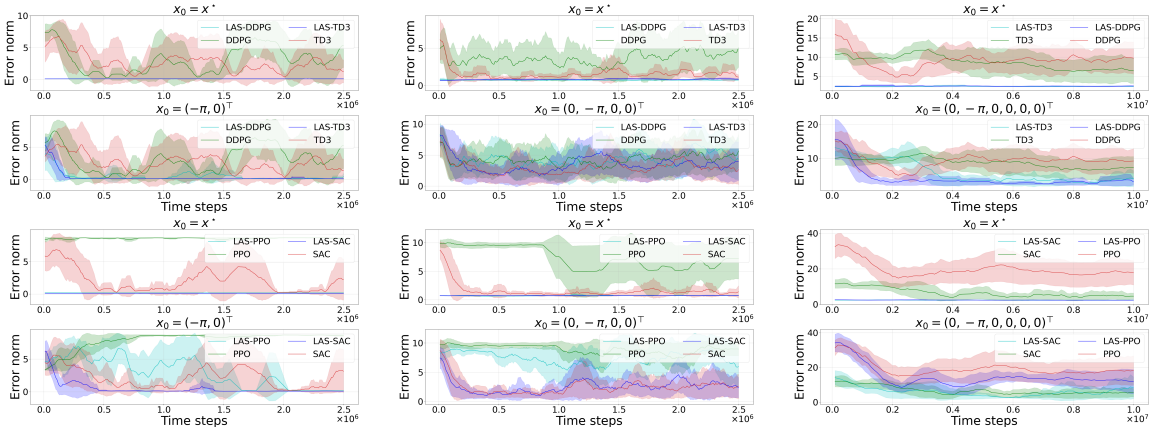


Figure 1.5: Stability score during training: PSU (left), IPSU (center), DIPSU (right). Row 1 and 3: local. Row 2 and 4: global.

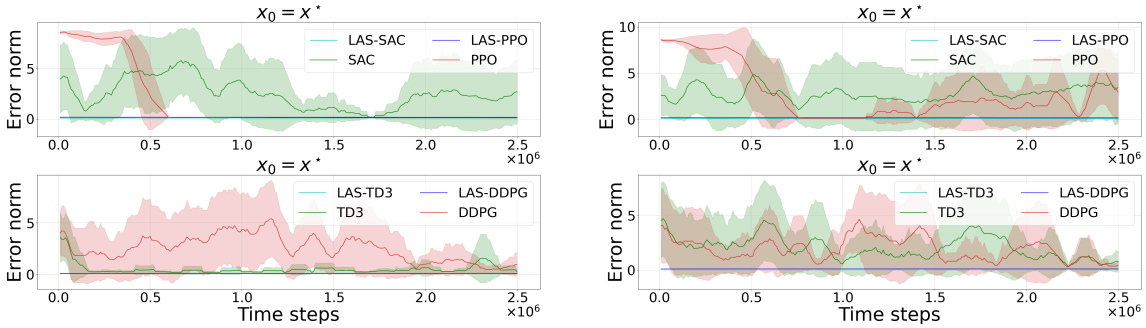


Figure 1.6: Local stability over multiple instances of PSU. Standard algorithms set of training masses $m_{train} = \{0.7, 1, 1.3\}$ (left), $m_{train} = \{0.9, 1, 1.1\}$ (right).

Parameters	Training	Corrupted
pole mass m	1	1.2
pole length l	1	1
gravity acceleration g	9.81	9.81
episode max length T_{ep}	200	1000
input bounds	[-2,2]	[-2,2]
noise w	0	$w \sim \mathcal{N}(0, 0.1)$
disturbance d	0	$0.2\sin(\frac{2\pi}{100}t)$
steady-state threshold T_{ss}	-	800

Table 1.4: Pendulum swing-up environment parameters

Parameters	Training	Corrupted
cart mass m_c	10.47	10.47
pole mass m_p	5	6.53
pole length l_p	0.6	0.8
rail bounds l_r	[-1,1]	[-1,1]
episode max length T_{ep}	1000	1000
input bounds	[-100,100]	[-100,100]
noise w	0	$w \sim \mathcal{N}(0, 0.173)$
disturbance d	0	$20\sin(\frac{2\pi}{50}t)$
steady-state threshold T_{ss}	-	800

Table 1.5: Inverted pendulum swing-up environment parameters

Parameters	Training	Corrupted
cart mass m_c	10	10
first pole mass m_{p1}	1	1
second pole mass m_{p2}	1	1.2
first pole length l_{p1}	0.6	0.6
second pole length l_{p2}	0.6	0.7
rail bounds l_r	[-2,2]	[-2,2]
episode max length T_{ep}	1000	2000
input bounds	[-200,200]	[-200,200]
noise w	0	$w \sim \mathcal{N}(0, 0.173)$
disturbance d	0	$20\sin(\frac{2\pi}{100}t)$
steady-state threshold T_{ss}	-	1600

Table 1.6: Double inverted pendulum swing-up environment parameters

	PSU	IPSU	DIPSU
LQR	0.6432	0.0856	0.0151
LAS-DDPG	1.0	0.8028	0.3362
LAS-PPO	0.9997	0.9987	0.0682
LAS-TD3	1.0	0.9454	0.4012
LAS-SAC	1.0	0.9501	0.4646

Table 1.7: Success rate over $N_i = 10^4$ initial conditions. Success threshold: $\epsilon = 0.001$. Initial condition bounds b_0 for the different tasks: $[\pi, 8]$ (PSU), $[1, \pi, 10, 3\pi]$ (IPSU), $[1, \pi, \pi, 10, 3\pi, 3\pi]$ (DIPSU).

1.2.4 Addressing general continuous reward functions

In this final section of the chapter, we discuss potential avenues for improving the previously presented results. In particular, we propose an objective reshaping technique that addresses more general cost functions, while ensuring local exponential stability at all times. By accepting a bounded performance error relative to an optimal policy defined by the initial objective, the techniques discussed in Section 1.1 can be utilized in a broader setting, thus providing local exponential stability properties.

In reinforcement learning, the choice of the objective function plays a pivotal role in determining the properties of the learned policy. When dealing with a non-quadratic reward function, ensuring that the learned policy is locally exponentially stabilizing becomes challenging [170]. This challenge is even more pronounced if we aim for this property to remain unaffected by the discount factor, similarly to Lemma 1.1. However, the requirement of a globally quadratic cost as in Section 1.1 may turn out to be restrictive to define general stabilization tasks. Furthermore, in the DRL community it is not common to use quadratic objectives.

To address these concerns, this section proposes a strategy to redesign a generic objective function, such that the optimal resulting policy is locally exponentially stabilizing independently from the discount factor. In particular, we modify the reward to be locally quadratic in proximity of the desired equilibrium point. As a consequence, the results of Section 1.1 can be directly applied. We consider a discrete-time nonlinear systems of the form

$$x^+ = f(x, u) \tag{1.33}$$

with $x \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$ and $u \in \mathcal{U} \subseteq \mathbb{R}$. We focus on objective functions of the form (1.11), namely,

$$J_\gamma(x(\mathfrak{t}), u(\mathfrak{t})) = \sum_{k=0}^{\infty} \gamma^k \mathbf{r}(x(k+\mathfrak{t}), u(k+\mathfrak{t})).$$

Moreover, we assume that at each time instant the environment returns an instantaneous reward defined by the continuous, possibly non-smooth function $\mathbf{r} : \mathcal{X} \times \mathcal{U} \to \mathbb{R}_{\geqslant 0}$, which is assumed to satisfy the following

$$\mathbf{r}(0,0) = 0, \qquad \mathbf{r}(x,u) > 0 \qquad \forall (x,u) \in \mathcal{X} \times \mathcal{U} \setminus (0,0). \tag{1.34}$$

Similarly to Section 1.1, the objective is to obtain a near-optimal parametrized control policy $\boldsymbol{\pi}^\theta : \mathcal{X} \times \mathbb{R}^{n_\theta} \to \mathcal{U}$ which guarantees local exponential stability independently from the choice of the parameter vector $\theta \in \mathbb{R}^{n_\theta}$. The characteristics in (1.34) seem natural when considering stabilization to an equilibrium point. However, local exponential stability properties of a general optimal solution to the presented problem are typically unknown, especially in the case of discounted objectives [170]. Therefore, we propose to reshape the reward function such that it ensures a locally exponentially stabilizing solution exists. In other words, we aim at designing a continuous function $\hat{\mathbf{r}} : \mathcal{X} \times \mathcal{U} \to \mathbb{R}_{\geqslant 0}$ such that the objective

$$\hat{J}_\gamma(x(\mathfrak{t}), u(\mathfrak{t})) = \sum_{k=0}^{\infty} \gamma^k \hat{\mathbf{r}}(x(k+\mathfrak{t}), u(k+\mathfrak{t})),$$

describes a locally exponentially stable closed-loop behavior independently from the discount factor γ.

In particular, we will pick a reshaped reward function which is locally quadratic and rely on the results in Section 1.1.1. We start by analyzing the effect of a reward reshaping. Hence, we study the performances with respect to $\mathbf{r}$ of the sub-optimal policy obtained under a modified reward function $\hat{\mathbf{r}}$. Let $\hat{\mathbf{r}} : \mathcal{X} \times \mathcal{U} \to \mathbb{R}_{\geqslant 0}$ be a continuous approximation of $\mathbf{r}$ such that

$$\hat{\mathbf{r}}(0,0) = 0, \quad \hat{\mathbf{r}}(x,u) > 0, \quad \forall (x,u) \in \mathcal{X} \times \mathcal{U} \setminus (0,0), \tag{1.35a}$$

$$|\hat{\mathbf{r}}(x,u) - \mathbf{r}(x,u)| \leqslant \epsilon, \quad \forall (x,u) \in \mathcal{X} \times \mathcal{U}. \tag{1.35b}$$

Following the state-value function definition (A.1), for an arbitrary policy $\boldsymbol{\pi} : \mathcal{X} \to \mathcal{U}$ and for any $x \in \mathcal{X}$, we have

$$\mathbf{J}_{\boldsymbol{\pi}}(x) = \sum_{k=0}^{\infty} \gamma^k \mathbf{r}(x(k+\mathfrak{t}), \boldsymbol{\pi}(x(k+\mathfrak{t}))), \tag{1.36a}$$

$$\hat{\mathbf{J}}_{\boldsymbol{\pi}}(x) = \sum_{k=0}^{\infty} \gamma^k \hat{\mathbf{r}}(x(k+\mathfrak{t}), \boldsymbol{\pi}(x(k+\mathfrak{t}))) \tag{1.36b}$$

Let us identify by $\boldsymbol{\pi}^\star$ and $\hat{\boldsymbol{\pi}}^\star$ the optimal policies minimizing $\mathbf{J}_{\boldsymbol{\pi}}$ and $\hat{\mathbf{J}}_{\boldsymbol{\pi}}$ respectively. As recalled in Appendix A.1, this implies

$$\mathbf{J}_{\boldsymbol{\pi}^\star}(x(\mathfrak{t})) = \min_{\boldsymbol{\pi}} \mathbf{J}_{\boldsymbol{\pi}}(x), \quad \boldsymbol{\pi}^\star(x) = \arg\min_{\boldsymbol{\pi}} \mathbf{J}_{\boldsymbol{\pi}}(x), \tag{1.37a}$$

$$\hat{\mathbf{J}}_{\hat{\boldsymbol{\pi}}^\star}(x(\mathfrak{t})) = \min_{\boldsymbol{\pi}} \hat{\mathbf{J}}_{\boldsymbol{\pi}}(x), \quad \hat{\boldsymbol{\pi}}^\star(x) = \arg\min_{\boldsymbol{\pi}} \hat{\mathbf{J}}_{\boldsymbol{\pi}}(x), \tag{1.37b}$$

for all $x \in \mathcal{X}$. Then, the following proposition presents an upper bound between policies performances in terms of state-value function. In particular, we show that $\lim_{\epsilon \to 0} \mathbf{J}_{\hat{\boldsymbol{\pi}}^\star}(x) = \mathbf{J}_{\boldsymbol{\pi}^\star}(x)$ for all $x \in \mathcal{X}$. Namely, we can arbitrarily approximate the optimal state-value function via a proper choice of the reshaped reward $\hat{\mathbf{r}}$.

Proposition 1.3. *Let $\hat{\mathbf{r}} : \mathcal{X} \times \mathcal{U} \to \mathbb{R}_{\geqslant 0}$ be a reward function as in* (1.35) *with $\epsilon \geqslant 0$ and let $\gamma \in (0, 1)$. Then, for all $x \in \mathcal{X}$, the following bound holds*

$$\mathbf{J}_{\hat{\boldsymbol{\pi}}^\star}(x) \leqslant \mathbf{J}_{\boldsymbol{\pi}^\star}(x) + \frac{2\epsilon}{1-\gamma} \,. \tag{1.38}$$

Proof. Consider the value functions (1.36). Properties (1.34), (1.35) and the triangle inequality imply the following inequalities

$$\mathbf{J}_{\boldsymbol{\pi}}(x) \leqslant |\mathbf{J}_{\boldsymbol{\pi}}(x) - \hat{\mathbf{J}}_{\boldsymbol{\pi}}(x)| + \hat{\mathbf{J}}_{\boldsymbol{\pi}}(x),$$
$$\hat{\mathbf{J}}_{\boldsymbol{\pi}}(x) \leqslant |\hat{\mathbf{J}}_{\boldsymbol{\pi}}(x) - \mathbf{J}_{\boldsymbol{\pi}}(x)| + \mathbf{J}_{\boldsymbol{\pi}}(x),$$

for all $x \in \mathcal{X}$ and for any arbitrary policy $\boldsymbol{\pi}$. Since $\boldsymbol{\pi}$ is assumed to be deterministic and acting on a deterministic environment, it generates unique deterministic trajectories for any initial condition. Then, bound (1.35b), $\gamma \in (0, 1)$ and the geometric series convergence properties let us conclude that

$$\mathbf{J}_{\boldsymbol{\pi}}(x) \leqslant \sum_{k=0}^{\infty} \gamma^k \epsilon + \hat{\mathbf{J}}_{\boldsymbol{\pi}}(x) \leqslant \frac{\epsilon}{1-\gamma} + \hat{\mathbf{J}}_{\boldsymbol{\pi}}(x), \tag{1.39a}$$

$$\hat{\mathbf{J}}_{\boldsymbol{\pi}}(x) \leqslant \sum_{k=0}^{\infty} \gamma^k \epsilon + \mathbf{J}_{\boldsymbol{\pi}}(x) \leqslant \frac{\epsilon}{1-\gamma} + \mathbf{J}_{\boldsymbol{\pi}}(x), \tag{1.39b}$$

where we used the fact that the policy generating the trajectories in $\mathbf{J}_{\boldsymbol{\pi}}$ and $\hat{\mathbf{J}}_{\boldsymbol{\pi}}$ is the same. From (1.37), optimality of $\boldsymbol{\pi}^\star$ and $\hat{\boldsymbol{\pi}}^\star$ implies

$$\mathbf{J}_{\boldsymbol{\pi}^\star}(x) \leqslant \mathbf{J}_{\boldsymbol{\pi}}(x), \qquad \hat{\mathbf{J}}_{\hat{\boldsymbol{\pi}}^\star}(x) \leqslant \hat{\mathbf{J}}_{\boldsymbol{\pi}}(x), \tag{1.40}$$

for all $x \in \mathcal{X}$ and all policies $\boldsymbol{\pi}$. Then, by combining (1.39) with (1.40) we obtain

$$\mathbf{J}_{\hat{\boldsymbol{\pi}}^\star}(x) \leqslant \frac{\epsilon}{1-\gamma} + \hat{\mathbf{J}}_{\hat{\boldsymbol{\pi}}^\star}(x) \leqslant \frac{\epsilon}{1-\gamma} + \hat{\mathbf{J}}_{\boldsymbol{\pi}^\star}(x) \leqslant \frac{2\epsilon}{1-\gamma} + \mathbf{J}_{\boldsymbol{\pi}^\star}(x),$$

thus concluding the proof. □

Unfortunately, bound (1.38) is not informative if $\gamma = 1$. This is due to the fact that (1.35b) allows for a non-zero discrepancy between $\hat{\mathbf{r}}$ and $\mathbf{r}$ in all points of the state-action space. In fact, we have not assumed any properties on the reward trajectories. To address this limitation and obtain more insightful bounds, we leverage the properties of these trajectories. Considering that our focus is on stabilization problems, it is reasonable to assume that all optimal policies derived from $\mathbf{r}$ and $\hat{\mathbf{r}}$ will eventually lead to convergence to the desired equilibrium, even if they are not stabilizing in a Lyapunov sense. In other words, we suppose that the respective infinite-horizon optimal state-value functions remain finite, causing the reward error to gradually vanish over time due to (1.34) and (1.35a). Then, we formally define this behavior through the following assumption.

Assumption 1.3. *Let $\boldsymbol{\pi}$ be a stabilizing policy for system* (1.33)*. Consider the reward functions $\mathbf{r}$ and $\hat{\mathbf{r}}$ satisfying* (1.34) *and* (1.35a) *respectively. We suppose there exists a bounded function $\epsilon_{\boldsymbol{\pi}} : (0, 1] \to \mathbb{R}_{\geqslant 0}$ such that, for any initial time $\mathfrak{t} \in \mathbb{N}$ and any initial condition $x(\mathfrak{t}) \in \mathcal{X}$, the*

solution of the closed-loop system satisfy

$$\sum_{k=1}^{\infty} \gamma^k |\hat{\mathbf{r}}(x(k+\mathfrak{t}), \boldsymbol{\pi}(x(k+\mathfrak{t}))) - \mathbf{r}(x(k+\mathfrak{t}), \boldsymbol{\pi}(x(k+\mathfrak{t})))| = \epsilon_{\boldsymbol{\pi}}(\gamma), \tag{1.41}$$

for any $\gamma \in (0, 1]$.

We emphasize that the conditions specified in Assumption 1.3 are relatively mild. Firstly, given the assumption of finite value functions and the shared global minimum between $\hat{\mathbf{r}}$ and $\mathbf{r}$, it is reasonable to anticipate that the disparity between instantaneous rewards will gradually diminish as both policies guide the system towards the same equilibrium. Secondly, the decay rate in time can be chosen to be extremely small. As an example, we may consider a case in which the difference between rewards satisfies along trajectories an harmonic convergence of the form

$$|\hat{\mathbf{r}}(x(k+\mathfrak{t}), \boldsymbol{\pi}(x(k+\mathfrak{t}))) - \mathbf{r}(x(k+\mathfrak{t}), \boldsymbol{\pi}(x(k+\mathfrak{t})))| \leqslant \frac{a}{k^{1+b}},$$

for some strictly positive a, b. In this case, one may verify that (1.41) is satisfied with

$$\epsilon_{\boldsymbol{\pi}}(\gamma) \leqslant a \frac{1+b}{b}.$$

Note that this bound is over-conservative and a more precise one can be obtained by studying convergence properties of polylogarithmic series. Under Assumption 1.3 we have the following result, showing that inequality (1.38) in Proposition 1.3 can be refined with an ultimate finite bound for any value of $\gamma \in (0, 1]$, with 1 included.

Proposition 1.4. *Let* $\hat{\mathbf{r}} : \mathcal{X} \times \mathcal{U} \to \mathbb{R}_{\geqslant 0}$ *be a reward function as in* (1.35a) *and let Assumption 1.3 hold. Then, for any compact set* $\mathcal{C} \subset \mathbb{R}^{n_x}$*, there exists a bounded function* $\varepsilon : (0, 1] \to \mathbb{R}_{\geqslant 0}$ *such that, for all* $x \in \mathcal{C}$*, the following bound holds*

$$\mathbf{J}_{\hat{\boldsymbol{\pi}}^\star}(x) \leqslant \mathbf{J}_{\boldsymbol{\pi}^\star}(x) + \varepsilon(\gamma). \tag{1.42}$$

Proof. By following the same first steps as in the proof of Proposition 1.3, we obtain

$$\mathbf{J}_{\boldsymbol{\pi}}(x(\mathfrak{t})) \leqslant \sum_{k=0}^{\infty} \gamma^k |\hat{\mathbf{r}}(x(k+\mathfrak{t}), \boldsymbol{\pi}(x(k+\mathfrak{t}))) - \mathbf{r}(x(k+\mathfrak{t}), \boldsymbol{\pi}(x(k+\mathfrak{t})))| + \hat{\mathbf{J}}_{\boldsymbol{\pi}}(x(\mathfrak{t})) \tag{1.43a}$$

$$\hat{\mathbf{J}}_{\boldsymbol{\pi}}(x(\mathfrak{t})) \leqslant \sum_{k=0}^{\infty} \gamma^k |\hat{\mathbf{r}}(x(k+\mathfrak{t}), \boldsymbol{\pi}(x(k+\mathfrak{t}))) - \mathbf{r}(x(k+\mathfrak{t}), \boldsymbol{\pi}(x(k+\mathfrak{t})))| + \mathbf{J}_{\boldsymbol{\pi}}(x(\mathfrak{t})). \tag{1.43b}$$

By Assumption 1.3 and (1.35), the infinite sum of reward errors satisfies the following bound

$$\sum_{k=0}^{\infty} \gamma^k |\hat{\mathbf{r}}(x(k+\mathfrak{t}), \boldsymbol{\pi}(x(k+\mathfrak{t}))) - \mathbf{r}(x(k+\mathfrak{t}), \boldsymbol{\pi}(x(k+\mathfrak{t})))| \\ \leqslant |\hat{\mathbf{r}}(x(\mathfrak{t}), \boldsymbol{\pi}(x(\mathfrak{t}))) - \mathbf{r}(x(\mathfrak{t}), \boldsymbol{\pi}(x(\mathfrak{t})))| + \epsilon_{\boldsymbol{\pi}}(\gamma).$$

Then, for any compact set of initial conditions $\mathcal{C}$, the error between rewards is bounded due

to the continuity of the functions. Hence,

$$\sum_{k=0}^{\infty} \gamma^k |\hat{\mathbf{r}}(x(k+\mathfrak{t}), \boldsymbol{\pi}(x(k+\mathfrak{t}))) - \mathbf{r}(x(k+\mathfrak{t}), \boldsymbol{\pi}(x(k+\mathfrak{t})))| \leqslant c_{\boldsymbol{\pi}} + \epsilon_{\pi}(\gamma), \tag{1.44}$$

with $c_{\boldsymbol{\pi}} = \sup_{x \in \mathcal{C}} |\hat{\mathbf{r}}(x, \boldsymbol{\pi}(x)) - \mathbf{r}(x, \boldsymbol{\pi}(x))|$. By combining (1.44) and (1.43), we obtain

$$\begin{aligned} \mathbf{J}_{\boldsymbol{\pi}}(x) &\leqslant c_{\boldsymbol{\pi}} + \epsilon_{\boldsymbol{\pi}}(\gamma) + \hat{\mathbf{J}}_{\boldsymbol{\pi}}(x) \\ \hat{\mathbf{J}}_{\boldsymbol{\pi}}(x) &\leqslant c_{\boldsymbol{\pi}} + \epsilon_{\boldsymbol{\pi}}(\gamma) + \mathbf{J}_{\boldsymbol{\pi}}(x). \end{aligned}$$

Then, by proceeding as in the proof of Proposition 1.3 and combining (1.45) with (1.40) we obtain

$$\begin{aligned} \mathbf{J}_{\hat{\boldsymbol{\pi}}^\star}(x) &\leqslant c_{\hat{\boldsymbol{\pi}}^\star} + \epsilon_{\hat{\boldsymbol{\pi}}^\star}(\gamma) + \hat{\mathbf{J}}_{\hat{\boldsymbol{\pi}}^\star}(x) \leqslant c_{\hat{\boldsymbol{\pi}}^\star} + \epsilon_{\hat{\boldsymbol{\pi}}^\star}(\gamma) + \hat{\mathbf{J}}_{\boldsymbol{\pi}^\star}(x) \\ &\leqslant c_{\hat{\boldsymbol{\pi}}^\star} + c_{\boldsymbol{\pi}^\star} + \epsilon_{\hat{\boldsymbol{\pi}}^\star}(\gamma) + \epsilon_{\pi^\star}(\gamma) + \mathbf{J}_{\boldsymbol{\pi}^\star}(x) = \mathbf{J}_{\boldsymbol{\pi}^\star}(x) + \varepsilon(\gamma), \end{aligned}$$

thus concluding the proof. □

Based on the aforementioned findings, it is evident that making local adjustments to the reward function has a limited impact on system performance, as long as the resulting optimal policy stabilizes the system towards the desired equilibrium point. Consequently, our objective now is to devise a locally quadratic reward function $\hat{\mathbf{r}}$ that ensures the generation of locally stabilizing solutions regardless of the value of the discount factor γ. In particular, inspired by the approach in Section 1.1, we pick a blending function $h_{\mathbf{r}} : \mathcal{X} \times \mathcal{U} \to \mathbb{R}_{\geqslant 0}$ such that $h_{\mathbf{r}} \in \mathcal{H}_{\mathbf{r}}$ with

$$\mathcal{H}_{\mathbf{r}} := \Big\{ h \in C^3 : \lim_{s \to 0} \tfrac{h(sz)}{s^j |z|^j} = 0, \lim_{s \to 0} \tfrac{h(sz)}{s^3 |z|^3} \neq 0, \lim_{|s| \to \infty} h(sz) = 1,\ j = 0, 1, 2 \Big\}.$$

and $z = \mathrm{col}(x, u)$. Then, we can select the approximate reward as

$$\hat{\mathbf{r}}(x, u) := z^\top W z + h_3(x, u) \left(\mathbf{r}(x, u) - z^\top W z \right), \qquad W := \begin{pmatrix} Q_\gamma & 0 \\ 0 & R_\gamma \end{pmatrix}, \tag{1.46}$$

with Q_γ, R_γ designed as in (1.12) according to some $Q \in \mathbb{S}^{n_x}_{\succ 0}, R \in \mathbb{S}^{n_u}_{\succ 0}$ and $P \in \mathbb{S}^{n_x}_{\succ 0}$ solution to the undiscounted DARE (1.8) with

$$A := \frac{\partial f}{\partial x}(0, 0), \qquad B := \frac{\partial f}{\partial u}(0, 0).$$

As per LQR theory presented at the beginning of Section 1.1, we assume the pair (A, B) to be controllable and the pair $(A, \sqrt{Q})$ to be observable. Then, as (1.46) is locally quadratic and due to the choice of matrices Q_γ, R_γ, the optimal local solution is (1.7), which is locally exponentially stabilizing independently from γ. Henceforth, all techniques proposed in Section 1.1 apply and provide a learned parametrized policy $\boldsymbol{\pi}^\theta$ as in (1.19) which is locally exponentially stabilizing for all $\theta \in \mathbb{R}^{n_\theta}$. In conclusion, the blending approach presented in this chapter can be used in a general context and can equip neural controllers trained on a wide class of reward functions with local stabilization properties.

2 Integral action for discrete-time nonlinear systems

It is well-known that model uncertainties have a notable impact on the regulation performance of controllers in real-world applications. Applying a control law derived from a simplified model to the actual plant may not precisely reach the desired reference, often resulting in a non-zero offset from the setpoint. This observation indicates that stability properties alone might be insufficient in certain applications. Then, following the scope of this first part of the manuscript, our next step involves identifying the factors that enable the transfer of stability properties between sufficiently similar discrete-time nonlinear systems. This is essential to address the critical challenge of transitioning from simulation, where the behavior of the controller is known, to the real-world, where stability of the closed-loop must be guaranteed. In the first sections of the chapter, we assume that a model of the system is known, and our primary focus is to analyze the robust stability properties of its equilibria. In the context of the feedback design application of the last section, the assumed knowledge of the model will reflect the knowledge of the simulated closed-loop behavior.

Our focus is on robust output regulation, where the goal is to steer the output to a constant reference or, equivalently, to reject constant disturbances. These problems encompass stabilization to equilibria as a special case, where the output corresponds to the full state. The fundamental tool in the following analysis is the concept of *total stability*, which was first introduced for continuous-time systems. One of the earliest appearance of total stability dates back to the works of Dubosin, Gorsin and Malkin [138, pp. 316] (see also [180] and references therein), and it was more recently studied in [17, 108, 180, 189]. In this context, robustness properties are analyzed directly via unstructured nonlinear model differences. This allows inferring the preservation of a stable equilibrium point for sufficiently similar plants by means of simple model comparison [17, Lemma 4, 5]. However, to the best of our knowledge, results mimicking these well-established general approaches are still missing in the discrete-time nonlinear context.

Some early necessary local conditions linked to discrete-time robustness appeared in [132]. Therein, the authors state that a necessary condition for local stability of nonlinear discrete-time autonomous systems comes from invertibility of the vector field at the equilibrium and close to it. The aforementioned condition can be seen as requiring the existence of a fixed point for a perturbed system that does not differ too much in norm from the initial one. In other words, the existence of a stable equilibrium has to be robust to small bounded variations of the local

dynamics. Also, results on robust stability appeared in [2, Chapter 5], where it is shown that if the origin of the nominal system is locally stable and Lipschitz, it is also locally robustly stable for bounded disturbances. More recent research in the field of converse Lyapunov theorems for discrete-time systems has established that the existence of a smooth Lyapunov function is a necessary condition for robust stability [111, 112]. Subsequently, [92] extended these results to the stochastic framework. However, it is worth noting that these works primarily focused on perturbations that vanish at the equilibrium. Furthermore, when considering persistent constant disturbances (e.g., as discussed in [92, Section 7.1]), the authors only provide practical stability guarantees without characterizing the system behavior inside the attractive set. This lack of characterization may not ensure the correct functioning of specific tools classically associated to output regulation. One common example is the integral action, which relies on the existence of equilibria, as highlighted in [17].

By translating the continuous-time framework results to the discrete-time scenario, we will be able to conclude that a neural controller trained on a simulator reproducing the fundamental modes of the real plant will generate reliable closed-loop trajectories once applied on the real system. This is the goal of this chapter. We draw conclusions similar to the continuous-time case, yet under some fundamental differences, given by the discrete nature of the system. In particular, we show that stability properties of the equilibrium of a nominal model imply the existence and stability of an equilibrium (possibly different from the former) for any perturbed system "sufficiently close" to the nominal one. The result is proved under some regularity assumptions and bounded mismatches. Moreover, we provide a counterexample highlighting that some results from the continuous-time scenario may not apply in the discretized framework, disproving some arguments of [132]. In the second section of the chapter, we link the obtained results on robust stability to the robust output regulation problem, building on recent forwarding techniques [142]. We justify the addition of an integral action for rejecting constant disturbances or tracking constant references. More specifically, we show that if the true model of the plant to be controlled is sufficiently close to one used for controller design, then output regulation is still achieved. Motivated by these findings, we combine the results of this chapter with the ones presented in Chapter 1. We derive a local optimal controller guaranteeing exponential stability of a cascade of a discrete-time linear system and an integrator, and we propose to globalize it to the nonlinear scenario via learning. Finally, we build on the presented theoretical results to train a robust DNN-based controller via model-free DRL methods on a simplified simulated nuclear fusion reactor. The learned neural control law shows a robust behavior in tracking constant references and in compensating modeling errors with respect to the precise simulator it is tested on. The results of this chapter are covered in [S3, S5].

2.1 Total stability for autonomous systems

In this section, we study how the stability properties of the origin of a given discrete-time autonomous nonlinear system

$$x^+ = f(x), \tag{2.1}$$

transfer to systems described by a sufficiently similar difference equation

$$x^+ = \hat{f}(x), \tag{2.2}$$

where $f : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$, $\hat{f} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ are continuous functions. As previously stated, the findings on autonomous systems will be exploited in the following section for the analysis of

robustness properties of the closed-loop. We propose two different results. The first one links the existence of an equilibrium for system (2.1) to the existence of an equilibrium for the perturbed system (2.2). Without loss of generality, we assume such an equilibrium for (2.1) to be the origin. We show that an equilibrium for (2.2) exists, provided that the two models are locally close enough. More precisely, the result holds if the functions f and $\hat{f}$ are not too different in the C^0 norm, and system (2.1) presents a forward invariant set (trajectories starting in the set remain in the set) containing its equilibrium and which is homeomorphic to the unit ball. The second result considers the case where both the dynamics and the Jacobians of the two systems (2.1), (2.2) are sufficiently similar. Under such conditions, we show that the existence of a locally exponentially stable equilibrium for (2.1) implies the existence and uniqueness of a locally exponentially stable equilibrium for (2.2) close to it. Moreover, we present a lower bound on the size of the domain of attraction of the equilibrium for (2.2).

The definition of total stability historically involves quantitative properties of the solution initiated from an equilibrium point under constantly acting disturbances [138, pp. 303]. Inspired by these seminal works, we propose the following qualitative version of such a definition, which is related to the preservation of an equilibrium. In particular, we define local similarities in the phase plots between the vector fields of systems whose dynamics are close enough. A corresponding delta-epsilon condition is given by Theorem 2.1.

Definition 2.1 (Total stability)**.** *The origin of system* (2.1) *is said to be totally stable if, for any $\hat{f}$ close enough to f, we can associate an equilibrium x_e which is (locally) asymptotically stable for system* (2.2)*.*

2.1.1 Existence of equilibria

We now present the minimal assumption required to show the existence of an equilibrium for (2.2). To this end, we introduce the following notation. Given a positive function $V : \mathcal{A} \subseteq \mathbb{R}^{n_x} \to \mathbb{R}_{\geqslant 0}$ and a positive real number $c > 0$, we denote the sublevel set of such a function as

$$\mathcal{V}_c(V) := \{x \in \mathcal{A} : V(x) \leqslant c\}. \tag{2.3}$$

We also recall two useful definitions. Given two topological spaces $\mathcal{X}, \mathcal{Y}$, a function $\phi : \mathcal{X} \to \mathcal{Y}$ is said to be a homeomorphism if it is continuous, bijective (i.e. its inverse exists and it is unique), and its inverse is continuous. Moreover, $\mathcal{X}$ and $\mathcal{Y}$ are said to be homeomorphic if there exists a function $\phi : \mathcal{X} \to \mathcal{Y}$ which is a homeomorphism.

Assumption 2.1. *Let $\mathcal{A}$ be an open subset of $\mathbb{R}^{n_x}$. There exists a C^0 function $V : \mathcal{A} \to \mathbb{R}_{\geqslant 0}$ satisfying $V(0) = 0$ and such that the following holds:*

1. *there exists a positive real number $\bar{c}$ such that the set $\mathcal{V}_{\bar{c}}(V)$ is homeomorphic to the unit ball;*
2. *there exists $\rho \in (0, 1)$ such that*

$$V(f(x)) \leqslant \rho V(x), \qquad \forall\, x \in \mathcal{V}_{\bar{c}}(V). \tag{2.4}$$

The first result is formalized by the following proposition. It states that, if the conditions of Assumption 2.1 hold, system (2.2) admits (at least) one equilibrium inside the set $\mathcal{V}_{\bar{c}}(V)$ if the functions f and $\hat{f}$ are "close enough".

Proposition 2.1. *Let Assumption 2.1 hold. Then, for any positive $\underline{c} \leqslant \bar{c}$ there exists a positive real number δ such that, for any continuous function $\hat{f} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ satisfying*

$$|\hat{f}(x) - f(x)| < \delta, \qquad \forall x \in \mathcal{V}_{\bar{c}}(V) \tag{2.5}$$

the corresponding system (2.2) *admits an equilibrium point $x_e \in \mathcal{V}_{\underline{c}}(V)$. Moreover, such system has no other equilibrium in the set $\mathcal{V}_{\bar{c}}(V) \setminus \mathcal{V}_{\underline{c}}(V)$.*

Proof. Consider $\underline{c} \leqslant \bar{c}$ and let $\hat{\rho}$ be any positive real number satisfying $\rho < \hat{\rho} < 1$. Since the set $\mathcal{V}_{\bar{c}}(V)$ is homeomorphic to the unit ball, it is bounded. Moreover, the function V being continuous, $\mathcal{V}_{\underline{c}}(V)$ is a compact subset. Next, we define the function $p : \mathbb{R}_{\geqslant 0} \to \mathbb{R}$ as

$$p(s) = \max_{\substack{x \in \mathcal{V}_{\bar{c}}(V) \\ v \in \mathbb{R}^{n_x} : |v| = 1}} \big\{ V(f(x) + sv) - \bar{c} \big\}, \tag{2.6}$$

with s a positive real number. Recalling item 2 of Assumption 2.1, we obtain

$$p(0) = \max_{\substack{x \in \mathcal{V}_{\bar{c}}(V) \\ v \in \mathbb{R}^{n_x} : |v| = 1}} \big\{ V(f(x)) - \bar{c} \big\} \leqslant \max_{\substack{x \in \mathcal{V}_{\bar{c}}(V) \\ v \in \mathbb{R}^{n_x} : |v| = 1}} \big\{ \rho V(x) - \bar{c} \big\} \leqslant (\rho - 1)\bar{c} < 0. \tag{2.7}$$

Then, we define the function $q : \mathbb{R}_{\geqslant 0} \to \mathbb{R}$ as

$$q(s) = \max_{\substack{x \in \mathcal{V}_{\bar{c}}(V) \setminus \mathcal{V}_{\underline{c}}(V) \\ v \in \mathbb{R}^{n_x} : |v| = 1}} \big\{ V(f(x) + sv) - \hat{\rho} V(x) \big\}.$$

It satisfies

$$\begin{aligned} q(0) &= \max_{x \in \mathcal{V}_{\bar{c}}(V) \setminus \mathcal{V}_{\underline{c}}(V)} \{ V(f(x)) - \hat{\rho} V(x) \} \leqslant \max_{x \in \mathcal{V}_{\bar{c}}(V) \setminus \mathcal{V}_{\underline{c}}(V)} \{ (\rho - \hat{\rho}) V(x) \} \\ &\leqslant (\rho - \hat{\rho})\underline{c} < 0. \end{aligned} \tag{2.8}$$

As a consequence, since p, q are continuous functions satisfying (2.7), (2.8), there exists $\delta > 0$ such that

$$p(s) < 0, \quad q(s) < 0, \qquad \forall\, s \in [0, \delta]. \tag{2.9}$$

Now, pick any continuous function $\hat{f}$ satisfying (2.5). Note that for all x in $\mathcal{V}_{\bar{c}}(V)$ such that $f(x) \neq \hat{f}(x)$,

$$V(\hat{f}(x)) - \bar{c} = V(f(x) + sv) - \bar{c}$$

with $s = |f(x) - \hat{f}(x)|$, $v = \frac{\hat{f}(x) - f(x)}{|f(x) - \hat{f}(x)|}$. Consequently, for all x in $\mathcal{V}_{\bar{c}}(V)$ inequalities (2.9) and (2.5) imply

$$V(\hat{f}(x)) - \bar{c} \leqslant p(|f(x) - \hat{f}(x)|) < 0. \tag{2.10}$$

Hence, $\hat{f}(x) \in \mathcal{V}_{\bar{c}}(V)$. By assumption, $\mathcal{V}_{\bar{c}}(V)$ is homeomorphic to a unitary ball, which we denote as $\mathbb{B} = \{z \in \mathbb{R}^{n_x} : |z| \leqslant 1\}$. Hence, there exists two continuous mappings $T : \mathcal{V}_{\bar{c}}(V) \to \mathbb{B}$ and $T^{-1} : \mathbb{B} \to \mathcal{V}_{\bar{c}}(V)$ such that $T \circ T^{-1}(z) = z$ for all z in $\mathbb{B}$. Hence, the mapping $T \circ \hat{f} \circ T^{-1} : \mathbb{B} \to \mathbb{B}$ is a continuous function. Hence, by Brouwer's fixed point theorem, there exists $z^* \in \mathbb{B}$ such that $T \circ \hat{f} \circ T^{-1}(z^*) = z^*$. Thus, it implies $\hat{f} \circ T^{-1}(z^*) = T^{-1}(z^*)$. We deduce that $x_e = T^{-1}(z^*)$ is a fixed point belonging to $\mathcal{V}_{\bar{c}}(V)$. Now, let us consider the

set $\mathcal{V}_{\bar{c}}(V) \setminus \mathcal{V}_{\underline{c}}(V)$. As before, with the same definitions of s and v, inequalities (2.9) and (2.5) imply that, for all $x \in \mathcal{V}_{\bar{c}}(V) \setminus \mathcal{V}_{\underline{c}}(V)$, it holds

$$V(\hat{f}(x)) - \hat{\rho}V(x) = V(f(x) + sv) - \hat{\rho}V(x) \leqslant q(s) < 0.$$

Hence, $\hat{f}(x) \neq x$ for all $x \in \mathcal{V}_{\bar{c}}(V) \setminus \mathcal{V}_{\underline{c}}(V)$. Consequently, x_e belongs to $\mathcal{V}_{\underline{c}}(V)$ and this concludes the proof. □

Comments about Assumption 2.1. We remark that Assumption 2.1 and Proposition 2.1 do not imply that equilibria of systems (2.1) and (2.2) are unique or attractive. Indeed, Assumption 2.1 is not requiring the nominal system (2.1) to be asymptotically stable. It solely assumes the existence of a forward invariant compact set that is homeomorphic to the unit ball. As a matter of fact, since V is not strictly positive outside of the origin (i.e. we are not asking the function V to be lower-bounded by a class $\mathcal{K}$ function of the norm of x), V may have multiple local minima. This does not allow concluding asymptotic stability of the origin. In order to clarify this aspect, we give the following simple example. Consider a system of the form

$$x^+ = f(x) := \begin{cases} \frac{1}{2}(x+c) & \text{if } x \geqslant c \\ \frac{1}{2}(x-c) & \text{if } x \leqslant -c \\ x & \text{otherwise} \end{cases}$$

where $x \in \mathbb{R}$ and $f : \mathbb{R} \to \mathbb{R}$ is a piecewise continuous linear function and c is a strictly positive constant. It can be seen that all the points satisfying $|x| \leqslant c$ are equilibrium points, yet, none of them is attractive. Now, select V as

$$V(x) := \begin{cases} 0 & \text{if } |x| \leqslant c\,, \\ |x| - c & \text{otherwise}\,. \end{cases}$$

It is non-negative, continuous, and all level sets identified by $\bar{c} > 0$ are homeomorphic to the unit ball. Moreover, by the definition of f and since $c > 0$ we obtain

$$V(x^+) = \begin{cases} 0 \leqslant \frac{1}{2}V(x) & \text{if } |x| \leqslant c \\ \frac{1}{2}|x| - \frac{1}{2}c = \frac{1}{2}V(x) & \text{otherwise} \end{cases}$$

This shows that Assumption 2.1 holds and Proposition 2.1 applies. For instance, consider $\hat{f}(x) := f(x) + \delta$, satisfying (2.5). Pick $\bar{c} = 2c$ and $\underline{c} = c$. If $2|\delta| \leqslant c$ then $\hat{f}$ admits a unique fixed point x_e satisfying $|x_e| = c + 2|\delta| \leqslant 2c$ and no other equilibrium exists inside the set $\mathcal{V}_{\bar{c}}(V) \setminus \mathcal{V}_{\underline{c}}(V) := \{2c < |x_e| \leqslant 3c\}$. Finally, as a further example, by selecting any $0 < \delta < \frac{1}{2}c$ and

$$\hat{f}(x) = \begin{cases} \delta & \text{if } 0 \leqslant x \leqslant \delta \\ \frac{cx - \delta^2}{c - \delta} & \text{if } \delta \leqslant x \leqslant c \\ f(x) + \delta & \text{otherwise} \end{cases}$$

it can be verified that $x_e^* = \delta$ is an additional equilibrium point with respect to the previously defined x_e. This new equilibrium lies inside $\mathcal{V}_0(V)$. The above example is represented in Figure 2.1.

Proposition 2.1 establishes the conditions under which the existence of an equilibrium for the nominal model (2.1) implies the existence of an equilibrium for any perturbed system (2.2) sufficiently close to the nominal one. This result parallels the continuous-time one presented

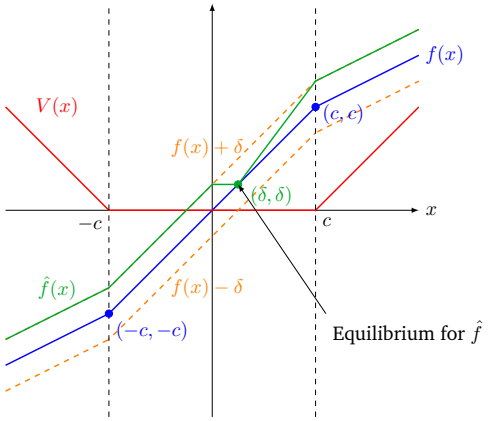


Figure 2.1: Example of perturbed dynamics preserving equilibria.

in [17, Lemma 4]. However, different assumptions are needed. In the continuous-time case, the origin of the nominal system is supposed to be asymptotically stable. In turn, for any forward invariant set, this ensures the existence of a Lyapunov function whose level sets are homeomorphic to a sphere, see [231, Theorem 1.2]. Unfortunately, in the discrete-time case this fact is in general not true. As a matter of fact, Lyapunov level sets may be non-homeomorphic to spheres, contrarily to what is stated in [132, Proof of Theorem 2.7]. As an example, see [90]. This phenomenon is due to the nature of such systems, as the presence of jumps doesn't allow an easy translation of continuous-time results. As a consequence, in Assumption 2.1 we ask for the existence of an invariant sublevel set $\mathcal{V}_{\bar{c}}(V)$ homeomorphic to a ball. At the same time, we do not require asymptotic stability of the origin.

In the following, we present an example showing the aforementioned behavior. In particular, we focus on a simple and stable linear system. We carefully craft a non-trivial Lyapunov function guaranteeing asymptotic stability of the origin. Successively, we exploit the structure of such a function to show that it has no sublevel set which is homeomorphic to a ball. Consider the linear system

$$x^+ = f(x) := \frac{1}{2}\,\mathrm{I}_{n_x}\, x,$$

where $x \in \mathbb{R}^{n_x}$, and the candidate Lyapunov function

$$V(x) = \begin{cases} 0 & x = 0 \\ 6|x| - 5 \cdot 2^i & 2^i \leqslant |x| < 2^i + 2^{i-1} \\ -4|x| + 5 \cdot 2^{i+1} & 2^i + 2^{i-1} \leqslant |x| < 2^{i+1} \end{cases}$$

where $i \in \mathbb{Z} \cup \{-\infty, +\infty\}$ and it is uniquely defined by $|x|$. It can be verified that $V(x)$ is continuous. Consider the case where $2^i \leqslant |x| < 2^i + 2^{i-1}$ for some i in $\mathbb{Z}$. We have $2^{i-1} \leqslant |f(x)| < 2^{i-1} + 2^{i-2}$ and

$$V(f(x)) - V(x) = 6|f(x)| - 5 \cdot 2^{i-1} - 6|x| + 5 \cdot 2^i = -3|x| + 5 \cdot 2^{i-1} < -2^{i-1} < 0.$$

Similarly, for $2^i + 2^{i-1} \leqslant |x| < 2^{i+1}$ for some i in $\mathbb{Z}$ we obtain $2^{i-1} + 2^{i-2} \leqslant |x| < 2^i$ and

$$V(f(x)) - V(x) = -4|f(x)| + 5 \cdot 2^i + 4|x| - 5 \cdot 2^{i+1} = 2|x| - 5 \cdot 2^i < -2^i < 0.$$

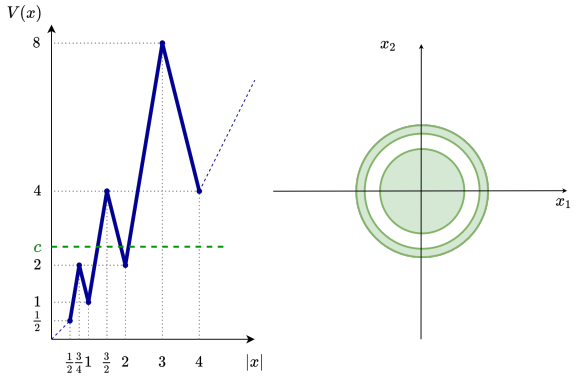


Figure 2.2: Lyapunov sublevel set for $x \in \mathbb{R}^2$

Hence, V is a continuous Lyapunov function for the system. However, it does not exist any $c > 0$ such that the corresponding sublevel set $\mathcal{V}_c(V)$ defined in (2.3) is homeomorphic to a ball, since each sublevel set is not path-connected. Figure 2.2 shows such a behavior for the planar case $x \in \mathbb{R}^2$.

Clearly, one could have picked a quadratic Lyapunov function for such a linear system. Yet, this example shows that discrete-time Lyapunov functions do not always possess the desired homeomorphicity property. As a consequence, we cannot guarantee that Lyapunov functions provided by converse Lyapunov theorems (e.g., [106]) satisfy such a property.

2.1.2 Existence of an exponentially stable equilibrium

We now present the main result on total stability of this section, which is formalized in the next theorem. This result parallels [17, Lemma 5]. We exploit the asymptotic properties of the equilibrium for the nominal system (2.1) to prove existence, uniqueness and stability of an equilibrium for (2.2).

Theorem 2.1. *Assume the origin of the system* (2.1) *is asymptotically stable with domain of attraction $\mathcal{A}$ and locally exponentially stable. Let $\underline{\mathcal{C}}$ be an arbitrary compact set satisfying $\{0\} \subset \underline{\mathcal{C}} \subset \mathcal{A}$ and suppose the function f is C^0 for all $x \in \mathcal{A}$ and C^1 for all $x \in \underline{\mathcal{C}}$. Then, for any forward invariant compact set $\overline{\mathcal{C}}$ verifying*

$$\{0\} \subset \underline{\mathcal{C}} \subset \overline{\mathcal{C}} \subset \mathcal{A},$$

there exists a positive scalar $\delta > 0$ such that, for any function $\hat{f}$ which is C^0 for all $x \in \mathcal{A}$ and C^1 for all $x \in \underline{\mathcal{C}}$ and satisfying

$$|\hat{f}(x) - f(x)| \leqslant \delta, \qquad \forall x \in \overline{\mathcal{C}}, \tag{2.11}$$

$$\left| \frac{\partial \hat{f}}{\partial x}(x) - \frac{\partial f}{\partial x}(x) \right| \leqslant \delta, \qquad \forall x \in \underline{\mathcal{C}}, \tag{2.12}$$

the corresponding system (2.2) *admits an equilibrium point $x_e \in \underline{\mathcal{C}}$, which is asymptotically stable with a domain of attraction containing $\overline{\mathcal{C}}$ and locally exponentially stable. In other words, the origin of system* (2.1) *is totally stable.*

Note that, differently from [17, Lemma 5], the nominal and perturbed models are required to be C^1 inside $\underline{\mathcal{C}}$ solely, while C^0 outside. This allows considering interesting continuous functions, such as saturation functions, which are not differentiable everywhere.

Proof. The proof is organized in several steps. We start by establishing the existence, uniqueness and local exponential stability of an equilibrium for the perturbed system (2.2). Then, we characterize the size of its domain of attraction by combining the converse Lyapunov theorems established in [106] and techniques similar to the ones used in the proof of Proposition 2.1. Before showing the proof we recall a useful relation. Given a symmetric positive definite matrix $P \in \mathbb{S}^{n_x}_{\succ 0}$, for any $A \in \mathbb{R}^{n_x \times n_x}$ and for any arbitrary scalar $r := r_1 r_2$ with $r_1, r_2 > 0$, the generalized Schur's complement implies that the following inequalities are equivalent

$$\begin{cases} A^\top P A - rP \prec 0 \\ -P \prec 0 \end{cases} \iff \begin{pmatrix} -r_1 P & A^\top P \\ PA & -r_2 P \end{pmatrix} \prec 0. \tag{2.13}$$

Step 1: Local analysis Let P be a positive definite symmetric matrix and $a \in (0,1)$ a real scalar satisfying

$$\frac{\partial f}{\partial x}(0)^\top P \frac{\partial f}{\partial x}(0) \preceq aP.$$

Since the origin is a locally exponentially stable equilibrium for system (2.1), its linearization around the origin is stable. Hence, the existence of P is guaranteed by discrete-time Lyapunov inequality. In the following, given any $c > 0$, we denote with $\mathcal{V}_c(P)$ the subset of $\mathbb{R}^{n_x}$ defined as

$$\mathcal{V}_c(P) := \{x \in \mathbb{R}^{n_x} : x^\top P x \leqslant c\} \subset \mathbb{R}^{n_x}. \tag{2.14}$$

Now, note that the quadratic Lyapunov function defined by P is a local Lyapunov function for system (2.1). Moreover, note that for any $a \in (0,1)$ we have $a < \frac{1+a}{2} < 1$. Then, by continuity there exists a real number $\varepsilon > 0$ such that $\mathcal{V}_\varepsilon(P) \subseteq \underline{\mathcal{C}}$ and

$$\frac{\partial f}{\partial x}(x)^\top P \frac{\partial f}{\partial x}(x) \prec \frac{1+a}{2} P \qquad \forall x \in \mathcal{V}_\varepsilon(P).$$

Equivalently, it holds

$$\Psi(x) := \begin{pmatrix} -\frac{1+a}{2} P & \frac{\partial f}{\partial x}(x)^\top P \\ P \frac{\partial f}{\partial x}(x) & -P \end{pmatrix} \prec 0 \tag{2.15}$$

for all $x \in \mathcal{V}_\varepsilon(P)$. Consider the candidate Lyapunov function

$$V(x) = x^\top P x, \qquad V(x^+) = f(x)^\top P f(x).$$

By defining a function $F : \mathbb{R} \to \mathbb{R}^{n_x}$ as $F(s) := f(sx)$, and since we assumed the origin to be an equilibrium point for f, we have

$$f(x) = f(x) - f(0) = F(1) - F(0) = \int_0^1 \frac{\partial F}{\partial s}(s)\, ds = \int_0^1 \frac{\partial f}{\partial x}(sx) x\, ds.$$

Then, recalling that $\frac{1+a}{2} < 1$ and via the addition and subtraction of $V(x^+)$, we compute

$$\begin{aligned} V(x^+) - \tfrac{1+a}{2} V(x) &= -\frac{1+a}{2} x^\top P x + \mathrm{He}\left\{ \int_0^1 x^\top \frac{\partial f}{\partial x}(sx)^\top P f(x)\, ds \right\} - f(x)^\top P f(x) \\ &= -\left(\frac{1+a}{2} x^\top P x + f(x)^\top P f(x) \right) \int_0^1 ds + \mathrm{He}\left\{ \int_0^1 x^\top \frac{\partial f}{\partial x}(sx)^\top P f(x)\, ds \right\} \\ &= \int_0^1 \left[-\frac{1+a}{2} x^\top P x + \mathrm{He}\left\{ x^\top \frac{\partial f}{\partial x}(sx)^\top P f(x) \right\} - f(x)^\top P f(x) \right] ds \\ &= \begin{pmatrix} x^\top & f(x)^\top \end{pmatrix} \int_0^1 \Psi(sx) ds \begin{pmatrix} x \\ f(x) \end{pmatrix} < 0 \end{aligned} \tag{2.16}$$

for all $x \in \mathcal{V}_\varepsilon(P)$ and $s \in [0,1]$, where in the last step we used the definition of Ψ in (2.15).

Step 2: Existence of an equilibrium for the perturbed system Now, let $\hat{f}$ satisfy

$$|\hat{f}(x) - f(x)| \leqslant \delta_1 := \sqrt{\frac{\varepsilon(1-a)^2}{8\bar{\lambda}(P)(3+a)}}, \tag{2.17}$$

for all $x \in \mathcal{V}_{\frac{\varepsilon}{2}}(P)$, where $\bar{\lambda}(\cdot)$ represents the maximum eigenvalue of its argument. In the following, we show that the existence of such a bound and the local stability of system (2.1) imply the existence of an equilibrium point $x_e \in \mathcal{V}_{\frac{\varepsilon}{2}}(P)$ for the perturbed system (2.2). To this aim, we leverage on Brouwer's fixed point theorem. Hence, we need to show that the set $\mathcal{V}_{\frac{\varepsilon}{2}}(P)$ is forward invariant for system (2.2) and that it is homeomorphic to a unit ball.

Let us choose $\bar{x} \in \mathbb{R}^{n_x}$ satisfying $\bar{x} \in \partial\mathcal{V}_{\frac{\varepsilon}{2}}(P)$ and note that for all $x \in \partial\mathcal{V}_{\frac{\varepsilon}{2}}(P)$

$$\hat{f}(x)^\top P \hat{f}(x) = \left[\hat{f}(x) - f(x)\right]^\top P \left[\hat{f}(x) - f(x)\right] + f(x)^\top P f(x) + 2\left[\hat{f}(x) - f(x)\right]^\top P f(x).$$

Then, using the generalized Young's inequality $2\alpha\beta \leqslant \nu^{-1}\alpha^2 + \nu\beta^2$ with $\nu = \dfrac{1-a}{2(1+a)}$, $\alpha = \left[\hat{f}(x) - f(x)\right]^\top \sqrt{P}$, $\beta = \sqrt{P} f(x)$ on the last term, we obtain

$$\begin{aligned} \hat{f}(x)^\top P \hat{f}(x) &\leqslant \frac{3+a}{1-a}\left[\hat{f}(x) - f(x)\right]^\top P \left[\hat{f}(x) - f(x)\right] + \frac{3+a}{2(1+a)} f(x)^\top P f(x) \\ &\leqslant \frac{1-a}{4}\bar{x}^\top P \bar{x} + \frac{3+a}{4} x^\top P x \leqslant \frac{1-a}{4}\bar{x}^\top P \bar{x} + \frac{3+a}{4}\bar{x}^\top P \bar{x} \leqslant \bar{x}^\top P \bar{x}, \end{aligned}$$

where we used inequality (2.17) in the second step for bounding the first term on the right-hand side of the inequality.

Thus, $\mathcal{V}_{\frac{\varepsilon}{2}}(P)$ is forward invariant for the system (2.2). Moreover, since $\partial\mathcal{V}_{\frac{\varepsilon}{2}}(P)$ is the level surface of a quadratic Lyapunov function, it is homeomorphic to a sphere. Hence, the set $\mathcal{V}_{\frac{\varepsilon}{2}}(P)$ is homomorphic to a closed unit ball. Following the proof of Proposition 2.1 and employing Brouwer's fixed point theorem, there exists a point $x_e \in \mathcal{V}_{\frac{\varepsilon}{2}}(P)$ satisfying $\hat{f}(x_e) = x_e$.

Step 3: Local exponential stability Now we show that $x_e \in \mathcal{V}_{\frac{\varepsilon}{2}}(P)$ is locally exponentially stable for (2.2). In particular, we show that if $\hat{f}$ satisfies

$$\left| \frac{\partial \hat{f}}{\partial x}(x) - \frac{\partial f}{\partial x}(x) \right| \leqslant \delta_2 := \frac{1-a}{2\sqrt{10+6a}}, \tag{2.18}$$

for all $x \in \mathcal{V}_\varepsilon(P)$, then, by denoting

$$\tilde{x} := x - x_e, \qquad \bar{f}(\tilde{x}) := \hat{f}(\tilde{x} + x_e) - \hat{f}(x_e) = \hat{f}(x) - x_e,$$

the candidate Lyapunov function $\widehat{V}(\tilde{x}) = \tilde{x}^\top P \tilde{x}$ has to satisfy

$$\widehat{V}(\tilde{x}^+) = \bar{f}(\tilde{x})^\top P \bar{f}(\tilde{x}) \leqslant \frac{3+a}{4}\widehat{V}(\tilde{x}) < \widehat{V}(\tilde{x}) \tag{2.19}$$

for all $\tilde{x} \in \widetilde{\mathcal{V}}_\varepsilon(P)$, where

$$\widetilde{\mathcal{V}}_\varepsilon(P) = \{\tilde{x} \in \mathbb{R}^{n_x} : (x_e + s\tilde{x}) \in \mathcal{V}_\varepsilon(P),\ \forall s \in [0,1]\}.$$

To show such a property, first note that by defining a function $G : \mathbb{R} \to \mathbb{R}^{n_x}$ as $G(s) = \hat{f}(x_e + s\tilde{x})$, it holds

$$\bar{f}(\tilde{x}) = G(1) - G(0) = \int_0^1 \frac{\partial \hat{f}}{\partial x}(x_e + s\tilde{x})\, ds\ \tilde{x}.$$

Then, similarly to the previous part of the proof, we compute

$$\begin{aligned} \widehat{V}(\tilde{x}^+) - \frac{3+a}{4}V(\tilde{x}) &= -\frac{3+a}{4}\tilde{x}^\top P \tilde{x} - \bar{f}(\tilde{x})^\top P \bar{f}(\tilde{x}) + 2\bar{f}(\tilde{x})^\top P \int_0^1 \frac{\partial \hat{f}}{\partial x}(x_e + s\tilde{x})\tilde{x}\, ds \\ &= \begin{pmatrix} \tilde{x}^\top & \bar{f}(\tilde{x})^\top \end{pmatrix} \int_0^1 \widehat{\Psi}(x_e + s\tilde{x}) ds \begin{pmatrix} \tilde{x} \\ \bar{f}(\tilde{x}) \end{pmatrix}, \end{aligned} \tag{2.20}$$

where in the last step we defined

$$\widehat{\Psi}(x) := \begin{pmatrix} -\frac{3+a}{4}P & \frac{\partial \hat{f}}{\partial x}(x)^\top P \\ P\frac{\partial \hat{f}}{\partial x}(x) & -P \end{pmatrix}. \tag{2.21}$$

Recalling (2.13) and since $a \in (0,1)$, if

$$\Phi(x) := \begin{pmatrix} -\frac{5+3a}{8}P & \frac{\partial \hat{f}}{\partial x}(x)^\top P \\ P\frac{\partial \hat{f}}{\partial x}(x) & -\frac{2(3+a)}{5+3a}P \end{pmatrix} \prec 0$$

then $\widehat{\Psi}(x) \prec 0$. Thus, we can study the sign semi-definiteness of $\Phi(x)$ to conclude the sign semi-definiteness of $\widehat{\Psi}(x)$. By adding and subtracting $\Psi(x)$ to $\Phi(x)$ and recalling inequality (2.15), we obtain,

$$\Phi(x) = \Phi(x) - \Psi(x) + \Psi(x) \prec \Phi(x) - \Psi(x)$$

Then,

$$\Phi(x) \prec \begin{pmatrix} \frac{a-1}{8}P & \frac{\partial \tilde{f}}{\partial x}(x)^\top P \\ P\frac{\partial \tilde{f}}{\partial x}(x) & \frac{a-1}{5+3a}P \end{pmatrix},$$

with $\frac{\partial \tilde{f}}{\partial x}(x) := \frac{\partial \hat{f}}{\partial x}(x) - \frac{\partial f}{\partial x}(x)$. Via generalized Schur's complement and bound (2.18), by following the lines of Step 2 and since $a \in (0,1)$, we conclude that the matrix on the right-hand side is negative semi-definite. Consequently, it holds $\widehat{\Psi}(x) \prec 0$ for all $x \in \Omega_\varepsilon$. As a consequence, we conclude $\widehat{\Psi}(x_e + s\tilde{x}) \prec 0$ for all $\tilde{x} \in \widetilde{\Omega}$ showing (2.19). We proved that x_e is locally exponentially stable for (2.2) with a domain of attraction including the set $\{\tilde{x} \in \widetilde{\mathcal{V}}_\varepsilon(P)\}$. Note that by the definition of $\widetilde{\mathcal{V}}_\varepsilon(P)$ and since $\mathcal{V}_\varepsilon(P)$ is convex, the point $x_e + s\tilde{x} = (1-s)x_e + sx$ belongs to the set $\mathcal{V}_\varepsilon(P)$ for all $(x, x_e, s) \in \mathcal{V}_\varepsilon(P) \times \mathcal{V}_\varepsilon(P) \times [0,1]$. Hence, the domain of attraction of $x_e \in \mathcal{V}_{\frac{\varepsilon}{2}}(P)$ contains the set $\{x \in \mathcal{V}_\varepsilon(P)\}$.

Step 4: Domain of attraction We now provide a stronger lower bound on the size of the domain of attraction of the equilibrium point x_e for the perturbed system (2.2). In particular, we show that x_e has a domain of attraction that includes the set $\overline{\mathcal{C}}$. First, by picking

$$\delta_3 = \min\{\delta_1, \delta_2\} \tag{2.22}$$

where δ_1 comes from (2.17) and δ_2 is defined in (2.18), we obtain that the point x_e is a locally exponentially stable equilibrium point for (2.2). Moreover, it is contained in $\underline{\mathcal{C}}$ and its domain of attraction includes $\mathcal{V}_\varepsilon(P)$.

Now, since $\overline{\mathcal{C}}$ is forward invariant and since system (2.1) is time-invariant and described by a continuous function on $\mathcal{A}$, we can leverage on [106, Theorem 1] to claim the existence of smooth Lyapunov functions $V_0 : \mathcal{A} \to \mathbb{R}_{\geqslant 0}$ and $V_{\overline{\mathcal{C}}} : \mathcal{A} \to \mathbb{R}_{\geqslant 0}$ such that for all $x \in \mathcal{A}$ it holds

$$\alpha_1(|x|) \leqslant V_0(x) \leqslant \alpha_2(|x|), \quad V_0(f(x)) \leqslant \rho_0 V_0(x), \quad V_0(x) = 0 \iff x = 0,$$

and

$$\alpha_1(\mathrm{d}(x, \overline{\mathcal{C}})) \leqslant V_{\overline{\mathcal{C}}}(x) \leqslant \alpha_2(\mathrm{d}(x, \overline{\mathcal{C}})), \quad V_{\overline{\mathcal{C}}}(f(x)) \leqslant \rho_{\overline{\mathcal{C}}} V_{\overline{\mathcal{C}}}(x), \quad V_{\overline{\mathcal{C}}}(x) = 0 \iff x \in \overline{\mathcal{C}},$$

where $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ and $(\rho_0, \rho_{\overline{\mathcal{C}}}) \in (0,1) \times (0,1)$. Note that the two Lyapunov functions can be bounded by different $\mathcal{K}_\infty$ functions. Yet, we can select the minimum (maximum) between those and obtain the same lower (upper) bound for the two. Consider now the set $\overline{\mathcal{C}}$. Since it is compact and included in $\mathcal{A}$, there exists a strictly positive real number $\bar{d}$ such that the set $\mathcal{D} := \{x \in \mathcal{A} : \mathrm{d}(x, \overline{\mathcal{C}}) \in (0, \bar{d}]\}$ is a subset of $\mathcal{A}$, namely $\mathcal{D} \subset \mathcal{A}$. Let

$$\mathbf{V}(x) = V_{\overline{\mathcal{C}}}(x) + \sigma V_0(x), \tag{2.23}$$

with

$$\sigma = \frac{\alpha_1(\bar{d})}{2\nu}, \qquad \nu = \sup_{x \in \mathcal{A} : \mathrm{d}(x, \overline{\mathcal{C}}) \leqslant \bar{d}} V_0(x).$$

By picking $\rho = \max\{\rho_0, \rho_{\overline{\mathcal{C}}}\}$, for all $x \in \mathcal{A}$ it satisfies

$$\alpha_3(|x|) \leqslant \mathbf{V}(x) \leqslant \alpha_4(|x|), \quad \mathbf{V}(f(x)) \leqslant \rho \mathbf{V}(x), \quad \mathbf{V}(x) = 0 \iff x = 0$$

with $\alpha_3, \alpha_4 \in \mathcal{K}_\infty$. This implies that $\mathbf{V}$ is a Lyapunov function for system (2.1) on $\mathcal{A}$. Define the following sets

$$\partial \mathcal{V}_{\bar{d}}(\mathbf{V}) := \{x \in \mathcal{A} : \mathbf{V}(x) = \alpha_1(\bar{d})\}, \quad \mathcal{V}_{\bar{d}}(\mathbf{V}) := \{x \in \mathcal{A} : \mathbf{V}(x) \leqslant \alpha_1(\bar{d})\}. \tag{2.24}$$

Due to the definitions of $V_{\overline{\mathcal{C}}}$ in (2.23) and its lower bound, we have

$$\alpha_1(\mathrm{d}(x, \overline{\mathcal{C}})) \leqslant V_{\overline{\mathcal{C}}}(x) \leqslant \mathbf{V}(x).$$

Hence, $\mathrm{d}(x, \overline{\mathcal{C}}) \leqslant \bar{d}$ for all $x \in \mathcal{V}_{\bar{d}}(\mathbf{V})$. As a consequence, for all $x \in \mathcal{V}_{\bar{d}}(\mathbf{V})$, we have $\nu \geqslant V_0(x)$ and by (2.23) it holds

$$V_{\overline{\mathcal{C}}}(x) = \alpha_1(\bar{d}) \left(1 - \frac{V_0(x)}{2\nu}\right) > 0.$$

These last relations imply $\overline{\mathcal{C}} \subset \mathcal{V}_{\bar{d}}(\mathbf{V}) \subseteq \mathcal{D} \subset \mathcal{A}$. Now, let us identify by $\underline{v} > 0$ a scalar such that if $\mathbf{V}(x) \leqslant \underline{v}$ then $x \in \mathcal{V}_{\frac{\varepsilon}{2}}(P)$, namely $x^\top P x \leqslant \frac{\varepsilon}{2}$ for all $x \in \mathcal{A} : \mathbf{V}(x) \leqslant \underline{v}$. Define

$$\delta_4 := \min \left\{ \inf_{(y', y'') \in \partial \mathcal{V}_{\bar{d}}(\mathbf{V}) \times \partial \overline{\mathcal{C}}} |y' - y''|, \frac{(1-\rho)\underline{v}}{\sup\limits_{x \in \mathcal{V}_{\bar{d}}(\mathbf{V})} \left| \frac{\partial \mathbf{V}}{\partial x}(x) \right|} \right\} \tag{2.25}$$

and consider (2.11) with (2.25). We can define the function $p : \mathbb{R}_{\geqslant 0} \to \mathbb{R}$ as

$$p(s) = \max_{x \in \overline{\mathcal{C}}, |v|=1} \{V(f(x) + sv) - \alpha_1(\bar{d})\}.$$

Then, by the first argument of (2.25) and (2.24), for all $s \in [0, \delta_4]$ we have $p(s) < 0$. Note that for all x in $\overline{\mathcal{C}}$ such that $f(x) \neq \hat{f}(x)$,

$$\mathbf{V}(\hat{f}(x)) - \alpha_1(\bar{d}) = V(f(x) + sv) - \alpha_1(\bar{d}) \tag{2.26}$$

with $s = |\hat{f}(x) - f(x)|$, $v = \frac{\hat{f}(x) - f(x)}{|\hat{f}(x) - f(x)|}$. Consequently, for all x in $\overline{\mathcal{C}}$,

$$\mathbf{V}(\hat{f}(x)) - \alpha_1(\bar{d}) \leqslant p(|\hat{f}(x) - f(x)|) \leqslant 0. \tag{2.27}$$

Thus, $\hat{f}(x) \in \mathcal{V}_{\bar{d}}(\mathbf{V})$ for all $x \in \overline{\mathcal{C}}$ and

$$f(x) + s(\hat{f}(x) - f(x)) \in \mathcal{V}_{\bar{d}}(\mathbf{V}), \quad \forall (x, s) \in \overline{\mathcal{C}} \times [0, 1].$$

Then, by picking $s \in [0, 1]$ and a function $H : \mathbb{R} \to \mathbb{R}$ as $H(s) = V(f(x) + s(\hat{f}(x) - f(x)))$, we have

$$\mathbf{V}(\hat{f}(x)) - \mathbf{V}(f(x)) = H(1) - H(0) = \int_0^1 \frac{\partial \mathbf{V}}{\partial x}\Big(f(x) + s(\hat{f}(x) - f(x))\Big)\, ds\, \big[\hat{f}(x) - f(x)\big].$$

Consequently, for all $x \in \mathcal{V}_{\bar{d}}(\mathbf{V})$, we obtain

$$\begin{aligned}\mathbf{V}(\hat{f}(x)) &\leqslant \mathbf{V}(f(x)) + \int_0^1 \sup_{x\in\mathcal{V}_{\bar{d}}(\mathbf{V})} \left|\frac{\partial \mathbf{V}}{\partial x}(x)\right| ds \left|\hat{f}(x) - f(x)\right| \\ &< \rho\mathbf{V}(x) + \sup_{x\in\mathcal{V}_{\bar{d}}(\mathbf{V})} \left|\frac{\partial \mathbf{V}}{\partial x}(x)\right| \delta < \rho\mathbf{V}(x) + (1-\rho)\underline{v}.\end{aligned}$$

Since $\rho \in (0,1)$, it holds

$$\begin{cases} \mathbf{V}(\hat{f}(x)) < \mathbf{V}(x) & \forall x \in \overline{\mathcal{C}} \setminus \mathcal{V}_{\frac{\varepsilon}{2}}(P) \\ \mathbf{V}(\hat{f}(x)) < \underline{v} & \forall x \in \mathcal{V}_{\frac{\varepsilon}{2}}(P). \end{cases} \tag{2.28}$$

Therefore, we may pick any function $\hat{f}$ satisfying conditions (2.11), (2.12) with $\delta < \min\{\delta_3, \delta_4\}$. The analysis shows that bound (2.25) ensures trajectories of the perturbed system (2.2) initialized in $\overline{\mathcal{C}}$ converge to $\mathcal{V}_{\frac{\varepsilon}{2}}(P)$. Moreover, we previously proved $\mathcal{V}_{\varepsilon}(P)$ is included in the exponentially stable domain of attraction of x_e. Hence, x_e is an asymptotically stable equilibrium for (2.2) and it is locally exponentially stable with domain of attraction including $\overline{\mathcal{C}}$. □

Theorem 2.1 states that, in the presence of bounded mismatches, existence of a stable equilibrium is preserved for the perturbed system (2.2). This new equilibrium may not coincide with the original one, yet it is guaranteed to be in its neighborhood. As a particular example, we may study the case in which $x^+ = f(x) + \delta$ with δ being a constant small perturbation. In such scenario, the equilibrium of the perturbed system slightly shifts (it is actually computed as the solution x_e to $f(x_e) + \delta = x_e$), and we recover the result proposed in [132, Theorem 2.7]. Hence, even if we highlight that the proof proposed by the authors is either incorrect or missing some arguments, we believe that the general statement may be correct.

Finally, we highlight that the existence of a (possibly shifted) equilibrium may be of great value in some applications. A direct example is the control of systems via integral-action, see e.g., [1, 17, 86], as detailed in the next section.

2.2 Total stability motivates integral action

As stated at the end of the previous section, the results of Theorem 2.1 can be of interest for the output regulation problem and, in particular, in the context of integral-action controllers, see e.g. [198]. Such control schemes are commonly employed for (constant) perturbation rejection or (constant) reference tracking. The most trivial examples are PI and PID laws. In particular, consider the following system

$$\begin{aligned} w^+ &= w, \\ \xi^+ &= F(w, \xi, u), \qquad e = H(w, \xi), \end{aligned} \tag{2.29}$$

where $\xi \in \mathbb{R}^{n_\xi}$ is the state of the plant, $u \in \mathbb{R}^{n_u}$ is the control input and $e \in \mathbb{R}^{n_e}$ represents the output to be regulated to zero (without loss of generality), with $n_e \leqslant n_u$. For instance, in a tracking problem, one can set $e = y - y_{\text{ref}}$ where y is an output that must converge to a desired reference y_{ref}. In the representation (2.29), $w \in \mathbb{R}^{n_w}$ are constant signals representing references to be tracked or (unknown) perturbations to be rejected. Following the output regulation theory and the celebrated internal-model principle, asymptotic regulation of the output e can

be achieved robustly with respect to model uncertainties of the functions F and H only if the controller contains an integral action (able to perfectly compensate unknown constant perturbations), see, e.g. [17,77,86] for the continuous-time case. We take inspiration from the results of the previous section, where constant perturbations can be included in a more general context of perturbed functions (e.g. instead of considering $f(x)+w$ with $f(0)=0$ we can simply consider a new function $\hat{f}(x)$ possibly with $\hat{f}(0)\neq 0$). Then, we set a nominal value for the model (2.29) and for the signals w. We describe the nominal system, possibly after a change of coordinate, in the form

$$\xi^+ = \varphi(\xi) + g(\xi, u), \qquad e = h(\xi)$$

with $\varphi(0)=0$, $g(\xi,0)=0$ and $h(\xi)=0$, so that the origin represents the nominal desired equilibrium on which the regulation objective $e=0$ is satisfied. Our objective is to show that the use of an integral action allows to robustly preserve the desired asymptotic regulation property $e=0$ in the presence of model uncertainties in the nominal functions φ, g, h (that may come from different values of w or uncertainties in F, H). For this reason, the explicit use of the variable w in the state dynamics is not used anymore. Still, one can keep in mind that all the functions may depend on some nominal value $w^\star$. Then, consider the following extended system

$$\begin{cases} \xi^+ = \varphi(\xi) + g(\xi, u), \quad e = h(\xi), \\ z^+ = z + k(\xi, e) \end{cases} \tag{2.30}$$

where $z \in \mathbb{R}^{n_e}$ represents a generalized integral discrete-time action with k being a C^1 function satisfying the following set of conditions

$$\begin{aligned} k(\xi, e) = 0 \quad &\iff e = 0 \\ |k(\xi, y_a) - k(\xi, y_b)| \quad &\leqslant L_1(\xi)|y_a - y_b| \\ \left|\frac{\partial k}{\partial \xi}(\xi, y_a) - \frac{\partial k}{\partial \xi}(\xi, y_b)\right| \quad &\leqslant L_2(\xi)\,|y_a - y_b| \end{aligned} \tag{2.31}$$

for all $\xi \in \mathbb{R}^{n_\xi}$, for all $(y_a, y_b) \in \mathbb{R}^{n_e} \times \mathbb{R}^{n_e}$ and for some continuous functions $L_1, L_2 : \mathbb{R}^{n_x} \to \mathbb{R}_{\geqslant 0}$. It can be easily seen that by selecting $k(\xi, e) = e$ we recover a standard discrete-time integrator

$$z^+ = z + e$$

that trivially satisfies the previous requirements with $L_1(\xi)=1$, $L_2(\xi)=0$. Under a feedback controller $u=\alpha(\xi,z)$, we can define the extended dynamics for $x=\mathrm{col}(\xi,z)$ as

$$x^+ = f(x) := \begin{pmatrix} \varphi(\xi) + g(\xi, \alpha(\xi, z)) \\ z + k(\xi, h(\xi)) \end{pmatrix}. \tag{2.32}$$

As previously explained, variations of the nominal value of the signals w and model uncertainties of the functions f, g, h can be fully captured and represented with a closed-loop perturbed model

$$x^+ = \hat{f}(x) := \begin{pmatrix} \hat{\varphi}(\xi) + \hat{g}(\xi, \alpha(\xi, z)) \\ z + k(\xi, \hat{h}(\xi, \alpha(\xi, z))) \end{pmatrix}. \tag{2.33}$$

Thus, the goal is to study the robustness properties of the closed-loop $x^+ = f(x)$ and, in particular, to establish conditions under which the existence of an equilibrium is preserved. As a matter of fact, the properties of the generalized integral action (2.31) allow concluding the desired regulation property $e=0$ is still preserved at the equilibrium in the presence of model uncertainties.

To this end, we introduce the following compact notation.

$$\begin{aligned}
\Delta_\xi(x) &:= |\hat{\varphi}(\xi) - \varphi(\xi) + \hat{g}(\xi, \alpha(\xi, z)) - g(\xi, \alpha(\xi, z))|,\\
\Delta_e(x) &:= |\hat{h}(\xi, \alpha(\xi, z)) - h(\xi)|,\\
\Delta_z(x) &:= |k(\xi, \hat{h}(\xi, \alpha(\xi, z))) - k(\xi, h(\xi)|,\\
\Delta_{\partial\xi}(x) &:= \left|\frac{\partial\hat{\varphi}}{\partial\xi}(\xi) - \frac{\partial\varphi}{\partial\xi}(\xi) + \frac{\partial\hat{g}}{\partial\xi}(\xi, \alpha(\xi, z)) - \frac{\partial g}{\partial\xi}(\xi, \alpha(\xi, z))\right|,\\
\Delta_{\partial u}(x) &:= \left|\frac{\partial\hat{g}}{\partial u}(\xi, \alpha(\xi, z)) - \frac{\partial g}{\partial u}(\xi, \alpha(\xi, z))\right|,\\
\Delta_{\partial e}(x) &:= \left|\frac{\partial\hat{h}}{\partial\xi}(\xi, \alpha(\xi, z)) - \frac{\partial h}{\partial\xi}(\xi)\right| + \left|\frac{\partial\hat{h}}{\partial u}(\xi, \alpha(\xi, z))\right|.
\end{aligned} \tag{2.34}$$

2.2.1 Existence of equilibria

We first study the existence of equilibria where regulation is achieved for the perturbed system (2.33). Hence, we assume φ, g, α, h to be continuous functions.

Proposition 2.2. *Let Assumption 2.1 hold for the closed-loop system* (2.32)*. Moreover, assume* $h(0) = 0$*,* $g(\xi, 0) = 0$ *for all* $\xi \in \mathbb{R}^{n_\xi}$ *and let* k *satisfy* (2.31)*. Then, for any positive* $\underline{c} \leqslant \bar{c}$ *there exists a positive real number* $\bar{\delta} > 0$ *such that, for any continuous functions* $\hat{\varphi} : \mathbb{R}^{n_\xi} \to \mathbb{R}^{n_\xi}$, $\hat{g} : \mathbb{R}^{n_\xi} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_\xi}$, $\hat{h} : \mathbb{R}^{n_\xi} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_e}$ *satisfying*

$$\Delta_\xi(x) + \Delta_e(x) \leqslant \bar{\delta}, \qquad \forall\, x \in \mathcal{V}_{\bar{c}}(V), \tag{2.35}$$

with Δ_ξ, Δ_e *defined in* (2.34)*, the corresponding system* (2.33) *admits an equilibrium point* $(\xi_e, z_e) \in \mathcal{V}_{\underline{c}}(V)$ *on which output regulation is achieved, i.e.* $\hat{h}(\xi_e, \alpha(\xi_e, z_e)) = 0$*. Furthermore, system* (2.33) *has no other equilibrium in the set* $\mathcal{V}_{\bar{c}}(V) \setminus \mathcal{V}_{\underline{c}}(V)$*.*

Proof. The proof relies on Proposition 2.1. It can be easily checked that (2.35) implies (2.11) for the extended system (2.32). Since $\mathbb{R}^{n_\xi} \times \mathbb{R}^{n_e}$ is a finite dimensional space, all norms are equivalent. Hence, there exists a strictly positive constant ℓ such that $|x| \leqslant \ell \sum_{i=1}^{n} |x_i|$ where x_i denotes the i-th component of x. Hence, recalling the compact notation (2.34), we have $|\hat{f}(x) - f(x)| \leqslant \ell(\Delta_\xi(x) + \Delta_z(x))$ for all $x \in \mathbb{R}^{q+p}$. By defining $L := \sup_{\xi \in \mathcal{V}_{\bar{c}}(V)} L_1(\xi) \geqslant 0$, we obtain $|\Delta_z(x)| \leqslant L|\Delta_e(x)|$ for all $x \in \mathcal{V}_{\bar{c}}(V)$ and therefore equation (2.31) yields $|\hat{f}(x) - f(x)| \leqslant \ell(1 + L)(\Delta_\xi(x) + \Delta_e(x))$ for all $x \in \mathcal{V}_{\bar{c}}(V)$. Hence, (2.35) implies (2.5) if $\bar{\delta} < \frac{\delta}{\ell(1+L)}$. Then, Proposition 2.1 guarantees the existence of an equilibrium $(\xi_e, z_e) \in \mathcal{V}_{\underline{c}}(V)$. In such an equilibrium, $z_e^+ = z_e$ and consequently $k(\xi, \hat{h}(\xi_e, \alpha(\xi_e, z_e))) = 0$. By (2.31), the last relation implies $\hat{h}(\xi_e, \alpha(\xi_e, z_e)) = 0$ and this concludes the proof. □

Proposition 2.2 is a direct application of Proposition 2.1 to the extended system (2.32). Hence, it guarantees the existence of (at least one) equilibrium for the perturbed closed-loop dynamics (2.33). In turn, due to the presence of the integral action, in such equilibrium the outputs e are identically 0, even in presence of model mismatches of the function φ, g. Thus, Proposition 2.2 shows that the addition of discrete-time generalized integrator dynamics to discrete-time nonlinear systems guarantees the existence of equilibria where (constant) reference tracking and (constant) disturbance rejection is achieved. In other words, if the controller (designed for a nominal system) is able to make the trajectories of the perturbed model converge to an equilibrium, then asymptotic regulation $e = 0$ is achieved.

2.2.2 Robust regulation

Proposition 2.2 does not provide any characterization of the attractivity properties of the equilibria. In other words, it does not characterize the asymptotic regulation objective when starting from an initial condition different from any of the equilibria. As a matter of fact, the generation of attractive and stable equilibria is often the main objective of control designs. Hence, we build on Theorem 2.1 to show that the addition of a discrete-time integral component to stabilizing controllers allows for robust asymptotic (constant) reference tracking and (constant) disturbance rejection.

Assumption 2.2. *There exists an open set $\mathcal{A} \subseteq \mathbb{R}^{n_\xi} \times \mathbb{R}^{n_e}$ and a C^1 function $\alpha : \mathbb{R}^{n_\xi} \times \mathbb{R}^{n_e} \to \mathbb{R}^{n_u}$ such that the control law $u = \alpha(\xi, z)$ makes the origin of system* (2.32) *asymptotically stable with domain of attraction $\mathcal{A}$ and locally exponentially stable.*

Under Assumption 2.2, we can infer robust setpoint-tracking properties for the extended system (2.32) by means of the results in the previous section.

Proposition 2.3. *Let Assumption 2.2 hold and let $\underline{\mathcal{C}}$ be an arbitrary compact subset of $\mathcal{A}$ including the origin. Moreover, let functions φ, g, h be C^0 for all $(\xi, z) \in \mathcal{A}$ and C^1 for all $(\xi, z) \in \underline{\mathcal{C}}$. Finally, assume $h(0) = 0$, $g(\xi, 0) = 0$ for all $\xi \in \mathbb{R}^{n_\xi}$ and let k satisfy* (2.31)*. Then, for any forward invariant compact set $\overline{\mathcal{C}}$ verifying $\{0\} \subset \underline{\mathcal{C}} \subset \overline{\mathcal{C}} \subset \mathcal{A}$, there exists a strictly positive scalar $\bar{\delta}$ such that, for functions $\hat{\varphi}, \hat{g}, \hat{h}$ that are C^0 for all $(\xi, z) \in \mathcal{A}$ and C^1 for all $(\xi, z) \in \underline{\mathcal{C}}$, and satisfy*

$$\Delta_\xi(x) + \Delta_e(x) \leqslant \bar{\delta}, \qquad \forall x \in \overline{\mathcal{C}}, \tag{2.36}$$

$$\Delta_{\partial\xi}(x) + \Delta_{\partial u}(x) + \Delta_{\partial e}(x) \leqslant \bar{\delta}, \qquad \forall\, x \in \underline{\mathcal{C}}, \tag{2.37}$$

with $\Delta_\xi, \Delta_e, \Delta_{\partial\xi}, \Delta_{\partial u}, \Delta_{\partial e}$ defined in (2.34)*, the corresponding system* (2.33) *admits an equilibrium point $(\xi_e, z_e) \in \underline{\mathcal{C}}$ which is locally exponentially stable and asymptotically stable with domain of attraction containing $\overline{\mathcal{C}}$. Moreover, on such an equilibrium, $\hat{h}(\xi_e, \alpha(\xi_e, z_e)) = 0$.*

Proof. To prove the existence of an asymptotically stable equilibrium for (2.33) with a locally exponential behavior, we rely on Theorem 2.1. As in the proof of Proposition 2.2, it can be easily checked that (2.36) implies (2.11) for the extended system (2.32). By following the same steps, we define $L := \sup_{\xi \in \overline{\mathcal{C}}} L_1(\xi) \geqslant 0$. With the same definition of ℓ as in the proof of Proposition 2.2, (2.36) implies (2.11) for all $x \in \overline{\mathcal{C}}$ if $\delta < \frac{\min\{\delta_3, \delta_4\}}{\ell(1+L)}$, with δ_3, δ_4 defined in Step 4 of the proof of Theorem 2.1. Similarly, we can check that (2.37) implies (2.12) for the extended system (2.32). Let us define

$$L_\alpha := \sup_{(\xi,z) \in \underline{\mathcal{C}}} \left\{ \left|\tfrac{\partial \alpha}{\partial \xi}(\xi, z)\right|, \left|\tfrac{\partial \alpha}{\partial z}(\xi, z)\right| \right\}, \quad L_k := \sup_{\xi \in \underline{\mathcal{C}}} \{L_1(\xi), L_2(\xi)\} \geqslant L.$$

To show this, let $\bar{\ell} > 0$ such that $|A| \leqslant \bar{\ell} \sum_{i=1}^{n} |A_i|$ where A is any (rectangular) matrix and A_i denotes the i-th row. Then, by recalling the notation introduced in (2.34) and using the previous inequalities, we obtain

$$\begin{aligned} \left|\tfrac{\partial \hat{f}}{\partial x}(x) - \tfrac{\partial f}{\partial x}(x)\right| &\leqslant \bar{\ell}\Delta_{\partial\xi}(x) + 2\bar{\ell}L_\alpha\Delta_{\partial u}(x) + 2\bar{\ell}L_\alpha L_k \left|\tfrac{\partial \hat{h}}{\partial u}(\xi, \alpha(\xi, z))\right| \\ &\quad + 2\bar{\ell}L_k \left|\tfrac{\partial \hat{h}}{\partial \xi}(\xi, \alpha(\xi, z)) - \tfrac{\partial h}{\partial \xi}(\xi)\right| \\ &\leqslant \bar{\ell}(1 + 2(L_\alpha + L_k + L_\alpha L_k))(\Delta_{\partial\xi} + \Delta_{\partial u} + \Delta_{\partial e}) \end{aligned}$$

for all $x \in \underline{\mathcal{C}}$. Then, by picking

$$\mu = \max\left\{\ell(1+L), \bar{\ell}(1+2(L_\alpha + L_k + L_\alpha L_k))\right\}$$

and $\bar{\delta} < \frac{\min\{\delta_3,\delta_4\}}{\mu}$. Theorem 2.1 guarantees the existence of an asymptotically stable equilibrium $x_e = (\xi_e, z_e)$ which is close to the origin and locally exponentially stable. In such an equilibrium, $z_e^+ = z_e$ and consequently $k(\xi, \hat{h}(\xi_e, \alpha(\xi_e, z_e))) = 0$. By (2.31), $\hat{h}(\xi_e, \alpha(\xi_e, z_e)) = 0$ and this concludes the proof. □

Proposition 2.3 states that discrete-time stabilizing controllers exploiting integral actions achieve robust output regulation to constant setpoints. As an intuitive example, we may consider controllers designed for a nonlinear model, whose constant parameters are identified through data. Then, Proposition 2.3 guarantees that, if the approximation error is sufficiently small, the controller will still stabilize the system and the output will reach the desired constant setpoint.

We finally remark that Proposition 2.3 holds for small model perturbations, i.e. for small variations of the value of w from its nominal value $w^\star$, when considering the original problem (2.29). Such a result parallels the works [17] in continuous-time. If one wants to guarantee output regulation in the presence of large variations of the values of w, it is generally needed to rely on incremental properties, e.g., [86].

About Assumption 2.2 and forwarding approach. It is not straightforward to see when and how Assumption 2.2 can be satisfied, namely how to design the feedback law $\alpha(x, z)$ for the extended dynamics (2.30). An elegant solution comes from discrete-time forwarding technique [142]. If the origin of the autonomous system $\xi^+ = \varphi(\xi)$ is globally asymptotically stable and locally exponentially stable, the origin of the extended system (2.32) can be stabilized via a feedback controller. By considering the SISO scenario $n_u = n_e = 1$, the result in [142, Theorem 4.2] provides a feedback law $u = \alpha(\xi, z)$ achieving global asymptotic and local exponential stability of the origin of the extended system (2.32).

In particular, suppose that $W : \mathbb{R}^{n_\xi} \to \mathbb{R}_{\geqslant 0}$ is a Lyapunov function for system (2.1) which is locally quadratic. Assume also that the linearization of (2.32) around the origin is stabilizable and suppose to know a mapping $M : \mathbb{R}^{n_\xi} \to \mathbb{R}$ satisfying $M(\xi^+) = M(\xi) + k(\xi, h(\xi))$. Then, we can perform a change of coordinates $\eta = z - M(\xi)$ and consider $\zeta = \mathrm{col}(\xi, \eta)$. This leads to the stabilizing controller $u = \bar{\alpha}(\zeta)$ given by the (implicit) solution to

$$u = -\frac{1}{u}\int_0^u \frac{\partial V}{\partial \zeta} G(\zeta^+(v), v) dv$$

with

$$V(\zeta) = W(\xi) + \eta^2,\ G(\zeta, u)^\top = \left(\frac{\partial g}{\partial u}(\xi, u), -\frac{\partial M}{\partial \xi}\frac{\partial g}{\partial u}(\xi, u)\right).$$

In the original coordinates, we obtain $\alpha(\xi, z) = \bar{\alpha}(\xi, z - M(\xi))$. The function V above defined is a Lyapunov function for the closed-loop systems. Moreover, note that if the function W has the desired homeomorphicity properties, so has V by construction.

We refer to [142] and references therein for more details about the existence of such a mapping M, and the extension to the MIMO scenario.

2.2.3 Globalizing local integral action-based controllers

By combining results of Section 1.1 and Section 2.2, one can learn a nonlinear controller with local stability guarantees and robustness to constant perturbations by solving the DARE (1.8) for the extended system composed by the linearized plant and the integrator. In what follows, we show that the choice of a locally quadratic objective as in Section 1.2.4 provides a local control policy which is solution to the forwarding problem for the linearized framework [142]. In turn, this guarantees local exponential stability of the closed-loop cascade.

Consider a discrete time nonlinear system evolving according to the difference equation

$$x^+ = f(x,u), \quad y = h(x)$$

where $f, h \in C^0$ and such that the open-loop dynamics are locally exponentially stable, namely $0 = f(0,0)$ and $A = \frac{\partial f}{\partial x}(0,0)$ is Schur stable. We remark that one can pre-stabilize the linearized model and consider f as the locally stable closed-loop. Our aim is to design a robust feedback controller solving the output regulation problem for constant signals. Without loss of generality, we assume such a constant reference to be 0. Hence, we extend the system with a discrete-time integrator of the error

$$\begin{cases} x^+ &= f(x,u), \quad y = h(x) \\ z^+ &= z + y \end{cases} \tag{2.38}$$

where $z \in \mathbb{R}^{n_y}$, and we look for a controller $u = \alpha(x,z)$ which is locally exponentially stabilizing for the cascade (2.38) and optimal with respect to the cost function

$$\begin{aligned} J(\zeta(\mathfrak{t}), u(\mathfrak{t})) \quad &:= \sum_{k=0}^{\infty} \gamma^k \left(\zeta(k+\mathfrak{t})^\top \bar{Q}_\gamma \zeta(k+\mathfrak{t}) + u(k+\mathfrak{t})^\top R u(k+\mathfrak{t}) \right), \\ &= \sum_{k=0}^{\infty} \gamma^k \left(\zeta(k+\mathfrak{t})^\top \begin{pmatrix} Q_{x,\gamma} & Q_{xz,\gamma}^\top \\ Q_{xz,\gamma} & Q_{z,\gamma} \end{pmatrix} \zeta(k+\mathfrak{t}) + u(k+\mathfrak{t})^\top R u(k+\mathfrak{t}) \right), \end{aligned} \tag{2.39}$$

where $\zeta := \mathrm{col}(x,z)$, $Q_{x,\gamma} \in \mathbb{S}^{n_x}_{\succ 0}, Q_{z,\gamma} \in \mathbb{S}^{n_y}_{\succ 0}, R_\gamma \in \mathbb{S}^{n_u}_{\succ 0}$ and $Q_{xz,\gamma} \in \mathbb{R}^{n_y \times n_x}$ is a coupling term. Due to the complexity of finding an optimal solution of (2.39) in the nonlinear framework, we rely on learning and the results of Section 1.1. Then, following the strategy proposed in to Section 1.1, we propose to blend a local forwarding-based controller with a global learned policy to prescribe a local guaranteed behavior of the extended system (see [24] for a continuous-time counterpart). As per Section 1.1.2, we structure the control law as

$$u = K_x x + K_z z + \bar{\alpha}(x,z),$$

where K_x, K_z are derived from forwarding and $\bar{\alpha}(x,z)$ is a nonlinear function to be learned. Our first steps involves the study of the local closed-loop behavior and we show how to select the local gains K_x, K_z.

Forwarding for integral action in discrete-time linear systems. We start our analysis by specializing the discrete-time forwarding technique to the case of linear time-invariant cascades. For our purposes, we will focus on the case where the second system is described by a discrete-time integrator. However, the results can be generalized to more general dynamics by following techniques similar to [18]. Consider the cascade system

$$\begin{cases} x^+ &= Ax + Bu, \quad y = Cx \\ z^+ &= z + y \end{cases} \tag{2.40}$$

where $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$ and $z \in \mathbb{R}^{n_y}$ with $n_y \leqslant n_u$. Without loss of generality, we assume the pair (A, B) to be controllable and matrix A to be Schur stable. As noted above, such an assumption is not restrictive, as we can consider A as the pre-stabilized closed-loop $A = \bar{A} + B\bar{K}$ with $\bar{A}$ possibly unstable and a properly selected $\bar{K}$. Moreover, we assume the following.

Assumption 2.3. *The matrix*

$$\Sigma := \begin{pmatrix} A - \mathrm{I}_{n_x} & B \\ C & 0 \end{pmatrix}$$

is full rank. Namely, $\operatorname{rank}(\Sigma) = n_x + n_y$.

Assumption 2.3 is related to the so called *non-resonance condition* [16, Chapter 4], and allows for the design of a control input u stabilizing the cascade. Let us define a matrix M as

$$M := C(A - \mathrm{I}_{n_x})^{-1}. \tag{2.41}$$

Since A is assumed to be Schur stable, such a matrix is well-defined and it is the unique solution to the Sylvester equation

$$MA = M + C. \tag{2.42}$$

We are now ready to present the closed-form solution to the forwarding problem defined by the cascade (2.40).

Proposition 2.4. *Let Assumption 2.3 hold and let* $R \in \mathbb{S}^{n_u}_{\succ 0}, Q \in \mathbb{S}^{n_x}_{\succ 0}$. *Moreover, let* $P \in \mathbb{S}^{n_x}_{\succ 0}$ *be solution of* $A^\top P A = P - Q$ *and* M *be defined as in* (2.41). *Then, the control law*

$$u = -(R + B^\top \bar{P} B)^{-1} B^\top ((PA + M^\top M)x - M^\top z), \quad \bar{P} := P + M^\top M \tag{2.43}$$

makes the closed-loop (2.40)-(2.43) *asymptotically stable.*

Proof. Consider the change of coordinates $\eta := z - Mx$. Then, the cascade (2.40) rewrites as

$$\begin{cases} x^+ &= Ax + Bu, \\ \eta^+ &= \eta - MBu \end{cases} \tag{2.44}$$

and the control law becomes

$$u = -(R + B^\top \bar{P} B)^{-1} B^\top (PAx - M^\top \eta) =: K_x x + K_\eta \eta. \tag{2.45}$$

Consider now the candidate Lyapunov function $V(x, \eta) := x^\top P x + \eta^\top \eta$. According to (2.44) and (2.45), we have

$$V(x^+, \eta^+) = x^\top T_{xx}\, x + \eta^\top T_{\eta\eta}\, \eta + x^\top T_{x\eta}\, \eta + \eta^\top T_{\eta x}\, x, \tag{2.46}$$

where we defined

$$T_{xx} := A^\top P A + A^\top P B K_x + K_x^\top B^\top P A + K_x^\top B^\top \bar{P} B K_x, \tag{2.47a}$$
$$T_{\eta\eta} := \mathrm{I}_{n_y} - MBK_\eta - K_\eta^\top B^\top M^\top + K_\eta^\top B^\top \bar{P} B K_\eta, \tag{2.47b}$$
$$T_{x\eta} := A^\top P B K_\eta - K_x^\top B^\top M^\top - K_x^\top B^\top \bar{P} B K_\eta, \tag{2.47c}$$
$$T_{\eta x} := -MBK_x + K_\eta^\top B^\top P A + K_\eta^\top B^\top \bar{P} B K_x. \tag{2.47d}$$

By the definitions of K_x, K_η in (2.45), we derive

$$B^\top PA = -(R + B^\top \bar{P}B)K_x, \quad B^\top M^\top = (R + B^\top \bar{P}B)K_\eta,$$

showing that (2.47) is equivalent to

$$\begin{aligned} T_{xx} &:= A^\top PA - K_x^\top (2R + B^\top \bar{P}B)K_x, & T_{\eta\eta} &:= \mathrm{I}_{n_y} - K_\eta^\top (2R + B^\top \bar{P}B)K_\eta, \\ T_{x\eta} &:= -K_x^\top (2R + B^\top \bar{P}B)K_\eta, & T_{\eta x} &:= -K_\eta^\top (2R + B^\top \bar{P}B)K_x. \end{aligned}$$

Hence, by (2.46) and the fact that $A^\top PA = \rho P$, we obtain the equivalence

$$\begin{aligned} V(x^+, \eta^+) &= x^\top A^\top PAx + \eta^\top \eta - (x^\top K_x^\top (2R + B^\top \bar{P}B)K_x x + \eta K_\eta^\top (2R + B^\top \bar{P}B)K_\eta \\ &\quad + 2x^\top K_x^\top (2R + B^\top \bar{P}B)K_\eta \eta) \\ &= x^\top A^\top PAx + \eta^\top \eta - (K_x x + K_\eta \eta)^\top (2R + B^\top \bar{P}B)(K_x x + K_\eta \eta) \\ &= V(x, \eta) - x^\top Qx - u^\top (2R + B^\top \bar{P}B)u. \end{aligned}$$

Then, since $Q \in \mathbb{S}^{n_x}_{\succ 0}$, by using discrete-time LaSalle arguments [146] system (2.44) in closed-loop with (2.45) can be proved to converge to the set

$$\{(x, \eta) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} : x = 0, u = 0\} = \{0\} \times \{(R + B^\top \bar{P}B)^{-1} B^\top M^\top \eta = 0\}.$$

Consider now Assumption 2.3 and select a vector $v = \mathrm{col}(-(A - \mathrm{I}_{n_x})^{-T} C^\top \bar{v}, \bar{v}) \in \mathbb{R}^{n_x + n_y}$ with $\bar{v} \in \mathbb{R}^{n_x}$. We have

$$v^\top \Sigma = \begin{pmatrix} \mathbf{0} & -\bar{v}^\top C(A - \mathrm{I}_{n_x})^{-1} B \end{pmatrix}.$$

Since Σ is full row-rank, $v^\top \Sigma = 0$ if and only if $v = \mathbf{0}$, namely, if $\bar{v} = \mathbf{0}$. Then, it must hold that $C(A - \mathrm{I}_{n_x})^{-1} B = MB \neq 0$. Consequently, by the invertibility of $R + B^\top \bar{P}B$, $\eta = 0$ is the only admissible solution, thus showing asymptotic convergence to the origin. □

Optimality of forwarding-based controller for linear systems. Bearing in mind the results of the above paragraph, we now aim at showing that (2.43) is an optimal solution for an LQR problem over the extended system (2.40). In doing so, we demonstrate that such a controller can provide an optimal local solution to be subsequently globalized via the approach of Section 1.1. Consequently, the goal is to prove that there exists a choice of the discount factor γ and the cost matrices $Q_{x,\gamma}, Q_{xz,\gamma}, Q_{z,\gamma}$ and R_γ in (2.39) for which (2.43) is the optimal solution in the linear framework. Similarly to Section 1.1, we start by analyzing the undiscounted framework, namely $\gamma = 1$, yielding the minimization objective

$$\begin{aligned} J_{\text{lin}}(x(0), z(0), u(0)) &:= \sum_{k=0}^{\infty} \begin{pmatrix} x(k)^\top & z(k)^\top \end{pmatrix} \bar{Q} \begin{pmatrix} x(k) \\ z(k) \end{pmatrix} + u(k)^\top Ru(k), \\ &= \sum_{k=0}^{\infty} \begin{pmatrix} x(k)^\top & z(k)^\top \end{pmatrix} \begin{pmatrix} Q_x & Q_{xz}^\top \\ Q_{xz} & Q_z \end{pmatrix} \begin{pmatrix} x(k) \\ z(k) \end{pmatrix} + u(k)^\top Ru(k), \end{aligned} \tag{2.48}$$

where $Q_x \in \mathbb{S}^{n_x}_{\succ 0}, Q_z \in \mathbb{S}^{n_y}_{\succ 0}, R \in \mathbb{S}^{n_u}_{\succ 0}$, $Q_{xz} \in \mathbb{R}^{n_y \times n_x}$ and the initial time $\mathfrak{t} = 0$, without loss of generality.

Proposition 2.5. *Let Assumption 2.3 hold and let $Q \in \mathbb{S}^{n_x}_{\succ 0}, R \in \mathbb{S}^{n_u}_{\succ 0}$. Moreover, let $P \in \mathbb{S}^{n_x}_{\succ 0}$ be solution of $A^\top PA = P - Q$ and M be defined as in* (2.41)*. Then, the control law* (2.43) *is the*

optimal solution to the minimization problem of (2.48) *subject to dynamics* (2.40) *with*

$$\begin{aligned} Q_x &= Q + \begin{pmatrix} A^\top P & M^\top M \end{pmatrix} \begin{pmatrix} BYB^\top & BYB^\top \\ BYB^\top & BYB^\top \end{pmatrix} \begin{pmatrix} PA \\ M^\top M \end{pmatrix}, \\ Q_z &= MBYB^\top M^\top, \\ Q_{xz} &= -MBYB^\top (PA + M^\top M), \end{aligned} \tag{2.49}$$

where we defined $Y = Y^\top := (R + B^\top P B + B^\top M^\top M B)^{-1}$.

Proof. By performing the change of coordinates $\eta = z - Mx$ and defining the aggregate state $\xi = \mathrm{col}(x, \eta)$, the extended system (2.40) can be written in the compact form

$$\xi^+ = F\xi + Gu, \qquad F = \begin{pmatrix} A & 0 \\ 0 & \mathrm{I}_{n_y} \end{pmatrix}, \qquad G = \begin{pmatrix} \mathrm{I}_{n_x} \\ -M \end{pmatrix} B. \tag{2.50}$$

Similarly, the control law (2.43) becomes

$$u = \begin{pmatrix} K_x & K_\eta \end{pmatrix} \xi, \tag{2.51}$$

where K_x, K_η are defined as in (2.45). As shown in the proof of Proposition 2.4, $V(\xi) = \xi^\top \mathbf{P} \xi$ with $\mathbf{P} = \mathrm{diag}(P, \mathrm{I}_{n_y})$ is a Lyapunov function for the closed-loop system (2.50)-(2.51). Then, by LQR theory (see Section 1.1), the Lyapunov function is also the optimal value function for an LQR problem. Consequently, by defining the state weight in (x, η)-coordinates as $\mathbf{Q} := \left(\begin{smallmatrix} \mathbf{Q}_x & \mathbf{Q}_{xz}^\top \\ \mathbf{Q}_{xz} & \mathbf{Q}_z \end{smallmatrix} \right)$ and by selecting $\mathbf{P} := \mathrm{diag}(P, \mathrm{I}_{n_y})$, the DARE (1.8) for the cascade (2.50) reads

$$\begin{aligned} F^\top \mathbf{P} F - F^\top \mathbf{P} G (R + G^\top \mathbf{P} G)^{-1} G^\top \mathbf{P} F &= \mathbf{P} - \mathbf{Q} \\ \begin{pmatrix} A^\top (P - PBYB^\top P) A & A^\top PBYB^\top M^\top \\ MBYB^\top PA & \mathrm{I}_{n_y} - MBYB^\top M^\top \end{pmatrix} &= \tilde{\mathbf{P}} \end{aligned} \tag{2.52}$$

where

$$\tilde{\mathbf{P}} := \begin{pmatrix} P - \mathbf{Q}_x & -\mathbf{Q}_{xz}^\top \\ -\mathbf{Q}_{xz} & \mathrm{I}_{n_y} - \mathbf{Q}_z \end{pmatrix}.$$

Recalling that $A^\top P A = P - Q$, equality (2.52) can be rewritten as

$$\begin{pmatrix} \mathbf{Q}_x - (Q + A^\top PBYB^\top PA) & \mathbf{Q}_{xz}^\top + A^\top PBYB^\top M^\top \\ \mathbf{Q}_{xz} + MBYB^\top PA & \mathbf{Q}_z - MBYB^\top M^\top \end{pmatrix} = 0. \tag{2.53}$$

With the change of coordinates $z = \eta + Mx$, the extended state cost under the selection (2.49) becomes

$$(x^\top z^\top) \bar{Q} \begin{pmatrix} x \\ z \end{pmatrix} = (x^\top \eta^\top) \begin{pmatrix} \mathrm{I}_{n_x} & M^\top \\ 0 & \mathrm{I}_{n_y} \end{pmatrix} \begin{pmatrix} Q_x & Q_{xz}^\top \\ Q_{xz} & Q_z \end{pmatrix} \begin{pmatrix} \mathrm{I}_{n_x} & 0 \\ M & \mathrm{I}_{n_y} \end{pmatrix} \begin{pmatrix} x \\ \eta \end{pmatrix} = (x^\top \eta^\top) \mathbf{Q} \begin{pmatrix} x \\ \eta \end{pmatrix}.$$

By expanding the product, we obtain

$$\mathbf{Q}_x = Q_x + M^\top Q_{xz} + Q_{xz}^\top M + M^\top Q_z M, \quad \mathbf{Q}_{xz} = Q_{xz} + Q_z M, \quad \mathbf{Q}_z = Q_z,$$

which leads to

$$\mathbf{Q} = \begin{pmatrix} Q + A^\top PBYB^\top PA & -A^\top PBYB^\top M^\top \\ -MBYB^\top PA & MBYB^\top M^\top \end{pmatrix},$$

thus verifying (2.52). Note that $\mathbf{Q} \in \mathbb{S}^{n_x+n_y}_{\succeq 0}$ since it can be rewritten as

$$\mathbf{Q} = \begin{pmatrix} Q & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} K_x^\top & K_\eta^\top \end{pmatrix} \begin{pmatrix} Y^{-1} & Y^{-1} \\ Y^{-1} & Y^{-1} \end{pmatrix} \begin{pmatrix} K_x & K_\eta \end{pmatrix},$$

where we used the definitions of K_x, K_η in (2.45) and the fact that $Y = YY^{-1}Y$. Moreover, since M is solution to the Sylvester equation (2.44), the change of coordinates is always well defined and $\mathbf{Q} \in \mathbb{S}^{n_x+n_y}_{\succeq 0}$ implies $\bar{Q} \in \mathbb{S}^{n_x+n_y}_{\succeq 0}$. Then, $\mathbf{P}$ is solution to the DARE (2.52) for the extended system and the optimal controller gain is

$$K^\star \;= -(R + G^\top \mathbf{P} G)^{-1} G^\top \mathbf{P} F = -YB^\top \begin{pmatrix} PA & -M^\top \end{pmatrix} = \begin{pmatrix} K_x & K_\eta \end{pmatrix}.$$

Consequently, the control law (2.51) is the optimal solution to the minimization problem (2.48)-(2.49) and this concludes the proof. □

Remark 2.1. *Notice that, while the weight on the x state can be independently controlled via Q, the cost on the integral state z is regulated solely by the input weight matrix R. This strong interconnection is due to the cascade structure, which inevitably intertwines the behavior of z to the one of u and x.*

Discounted locally quadratic rewards for locally stabilizing neural policies. The results of the above paragraph does not seem to allow for a free choice of the extended state weight matrix $\bar{Q} = \left(\begin{smallmatrix} Q_x & Q_{xz}^\top \\ Q_{xz} & Q_z \end{smallmatrix}\right)$. However, by recalling that the definition of DC-gain for discrete-time linear systems $K_0 = C(A - \mathrm{I}_{n_y})^{-1}B = MB$, the selection (2.49) can be rewritten as

$$\begin{aligned} Q_x &= Q + \begin{pmatrix} A^\top P & M^\top \end{pmatrix} \begin{pmatrix} B(R+Y)^{-1}B^\top & BYK_0^\top \\ K_0 YB^\top & K_0 YK_0^\top \end{pmatrix} \begin{pmatrix} PA \\ M \end{pmatrix}, \\ Q_z &= K_0 YK_0^\top, \\ Q_{xz} &= -(K_0 YB^\top PA + K_0 YK_0^\top M), \end{aligned}$$

where $Y = Y^\top := (R + B^\top PB + K_0^\top K_0)^{-1}$. Since K_0 (and consequently M) depend solely on the pre-stabilized matrix A, an arbitrary selection of the matrix $\bar{Q}$ generates a specific pre-stabilizing behavior, without impacting the exponential stability properties of the closed-loop. Then, following Section 1.1, we can globalize the local controller by choosing a locally quadratic cost according to Section 1.2.4, and selecting a local discounted quadratic reward in the (x, η)-coordinates following Lemma 1.1, with $\mathbf{Q}_\gamma = \left(\begin{smallmatrix} \mathbf{Q}_{x,\gamma} & \mathbf{Q}_{xz,\gamma}^\top \\ \mathbf{Q}_{xz,\gamma} & \mathbf{Q}_{z,\gamma} \end{smallmatrix}\right)$ and $\mathbf{P} = \mathrm{diag}(P, \mathrm{I}_{n_y})$.

2.3 Robust deep reinforcement learning for tokamak reactors

Equipped with the robustness results of Section 2.2, we now aim at robustifying DNN-based control laws. We focus on the complex problem of controlling the shape profile of plasma in tokamak reactors. Tokamak reactors are toroidal nuclear fusion reactors where the plasma is controlled and contained via strong magnetic fields [224, 229]. Because of the high uncertainties in the measurements and estimations of plasma profiles, as well as in the modeling of

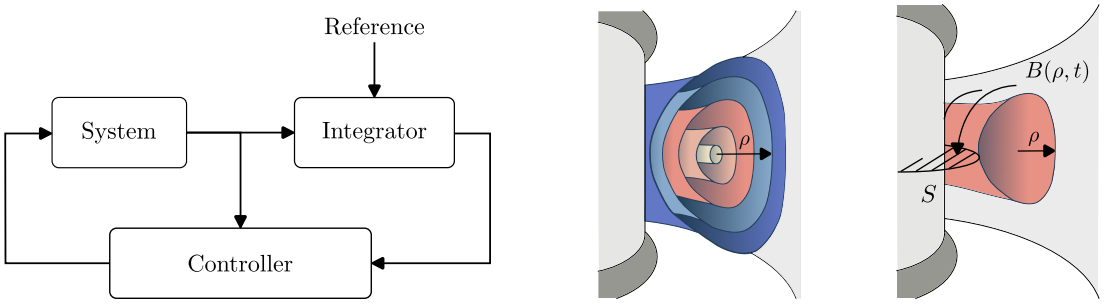
kinetic and magnetic dynamics, robust feedback control is crucial to obtain high-performance operations.

In the plasma control literature, it is common to interchangeably speak about the current profile, safety factor q (and its inverse ι-profile), magnetic flux gradient profile, and magnetic flux profile. In tokamak reactors, the safety factor is strongly linked to the plasma shape profile and it has been found to be strictly related to Magnetohydrodynamic (MHD) activities [229]. Therefore, robustly controlling the safety factor to the desired profile becomes an essential step towards obtaining long-time discharges [100]. In this section, we will consider the safety factor profile control problem during the so-called flat-top phase.

From an operative point of view, a lot of contributions have been made by the plasma physics and control communities working together. For instance, an overview of the plasma control in the Tore Supra tokamak can be found in [149]. A first attempt to model and numerically simulate the plasma profile evolution has been made in [29]. An early contribution in control of tokamak reactors appeared in [230], where the authors show experimental results using proportional feedback for the control of the internal inductance on Tore Supra. Then, throughout the years, consistent effort has been put in proposing different strategies for modeling and controlling such complex systems , see e.g., [20, 21, 32, 34, 81, 82, 119, 137, 144, 148, 156, 157, 159, 225]. However, the main challenge in model-based control of safety factor profile for advanced mode operations is the derivation of dynamical models that are complex enough to retain the main physical properties, and simple enough to be used for feedback design. Clearly, this requirement impacts the set of achievable closed-loop performances, due to the complexity of controllers computation. Hence, recent machine learning architectures such as DNNs stand out as valuable performance-oriented tool for addressing the problem in a data-driven fashion. In particular, the model-agnostic approach of many DRL algorithms has been shown to be effective on a wide variety of complex tasks [15, 125].

Unfortunately, few works have so far been dedicated to controlling PDEs using DRL, with the exception of some specific methods such as [68, 69, 160]. Only more recently, some DRL algorithms have been used in the context of nuclear fusion control. In [222], the authors proposed a DRL technique to control the safety factor during the ramp-up phase, while in [190] the authors developed a DRL algorithm to train a feed-forward controller for the kinetic profiles. In [223], the authors present a DRL-based algorithm for simultaneous control of the safety factor and normalized beta in the JT-60SA tokamak. Finally, a DRL controller has been proposed in [61] for plasma shape control.

In this section, we propose to design a robust dynamic DNN-based controller to be trained on a simplified model of a TCV tokamak reactor and successively tested on a control-oriented plasma transport simulator (RAPTOR) [70,136,145,217]. The idea is to rely on the results of Section 2.2 and complement the neural controller with a discrete-time integrator. As such, we consider the problem of stabilizing the extended cascade system composed of the plant and the integrator, as in Figure 2.3a. As the learning agent is trained to stabilize the extended dynamics, it learns a robust controller. Then, we aim at showing that the inclusion of an error integrator during and after training robustifies DNN-based controllers learned via DRL, by improving generalization to unseen constant perturbations. Before moving to the control problem, we highlight that such a structure (DNN with an integrator) resembles the one of Recursive Neural Networks (RNNs), e.g. [99, 186]. As a matter of fact, the addition of an integrator can be seen as adding “memory" in the agent, by providing time-related information. However, the generalization and robustness properties are due to the specific structure of the integrator dynamics.



(a) Extended system and controller.

(b) Simplified representation of tokamak quantities

2.3.1 Control problem

Dynamic model. We start by presenting the tokamak dynamics and the simplified (and discretized) ones used for training. Additional details can be found in Appendix B. Consider $\psi(R, Z)$ the poloidal flux of the magnetic field $B(R, Z)$ passing through a disc centred at the toroidal axis at height Z and with surface $S = \pi R^2$ where R is the large plasma radius. Let the magnetic flux be defined as

$$\psi(R, Z) := \frac{1}{2\pi} \int_S B(R, Z) dS. \tag{2.54}$$

Relation (2.54) can also be expressed in terms of the spatial index ρ. The term ρ is the toroidal flux coefficient indexing the magnetic surfaces, defined as $\rho = (2\phi/B_{\phi,0})^{\frac{1}{2}}$, where ϕ is the toroidal magnetic flux and $B_{\phi,0}$ is the value of the toroidal magnetic flux at the plasma center. The spatial index belongs to the interval $[0, a_\rho]$ where a_ρ is the minor plasma radius corresponding to the Last Closed Flux Surface (LCFS). A simplified representation is shown in Figure 2.3b, where the different colors correspond to different magnetic surfaces identified by ρ. Given the above quantities, the dynamics of (2.54) can be expressed by the following reaction-diffusion equation [233]

$$\frac{\partial \psi}{\partial t}(s, t) = \frac{D(s, t)}{a_\rho^2} \frac{\partial^2 \psi}{\partial s^2}(s, t) + \frac{G(s, t)}{a_\rho} \frac{\partial \psi}{\partial s}(s, t) + S(s, t). \tag{2.55}$$

where $s = \rho/a_\rho$ identifies the normalized spatial variable, $D(s, t)$ and $G(s, t)$ are diffusion parameters, while $S(s, t)$ is the source term and are defined in (B.2). We suppose the magnetic flux to be controlled by two Electron Cyclotron Current Drive (ECCD) systems, each characterized by its input power $P_{eccd,i}$, where $i \in 1, 2$. Moreover, similarly to [145], we assume that the two inputs are applied to the same spatial point. Specifically, the first antenna $P_{eccd,1}$ has a positive effect on ξ, while the second antenna $P_{eccd,2}$ has a negative effect on it. Thus, from a theoretical point of view, the two inputs can be treated as a single one.

An important quantity of plasma control in tokamak devices is the *safety factor*. This distributed variable measures the toroidal over poloidal turns of a field line passing through a point (R, Z) in a toroidal plane. Since the magnetic field lines are assumed to be equal in the same magnetic surface, we can define the safety factor for each magnetic surface indexed by s. In particular, the safety factor is defined as the quotient between the toroidal and poloidal gradient, which reads as

$$q(s, t) := \frac{d\phi}{d\psi} = \frac{\partial \phi / \partial s}{\partial \psi / \partial s} = -\frac{B_{\phi,0} a_\rho^2 s}{\partial \psi / \partial s} \tag{2.56}$$

where $\phi(s, t)$ is the toroidal flux defined in (B.7). Another important quantity in plasma analysis

is the ι-profile, which is defined as the inverse of the safety factor

$$\iota(s,t) = \frac{1}{q(s,t)} = \frac{\partial\psi/\partial s}{B_{\phi,0}a_\rho^2 s}. \tag{2.57}$$

The ι-profile is a more natural control variable since it proportionally depends on the poloidal flux gradient. Additional details about the model are discussed in Appendix B.1. The following equation gives the plasma thermal energy dynamics

$$\begin{cases} \tau_{th} = e^{-5.7466} P_{oh}^{0.0214}(1+P_{eccd,1})^{0.0426}(1+P_{eccd,2})^{0.0012} \\ \frac{d}{dt}W_{th} = -\frac{1}{\tau_{th}}W_{th} + P_{tot}, \quad W_{th}(0) = P_{tot}(0)\tau_{th}(0) \end{cases} \tag{2.58}$$

where

$$P_{tot} = \sum_{i=1}^{N_{eccd}} P_{eccd,i} + P_{OH}, \tag{2.59}$$

with P_{OH} identifying the ohmic power and is defined in (B.8).

By means of an implicit-explicit time discretization and a fixed-step spatial discretization for equation (2.55), and an implicit-explicit time discretization for equation (2.58), we obtain the difference equation

$$\xi^+ = f(\xi, u) \tag{2.60}$$

in the state variable $\xi = \mathrm{col}(\psi,\, W_{th}) \in \mathbb{R}^{n_\psi+1}$ and input $u \in [0,1]$. The discrete vector field f is defined in (B.14). Here, n_ψ is the number of elements used in the spatial discretization of equation (2.55). Hence, at iteration ŧ, the vector $\psi \in \mathbb{R}^{n_\psi}$ denotes the discretized magnetic flux in n_ψ spatial points, while W_{th} represents the thermal energy. The relation between u and $P_{eccd,i}$ and further information on the simulation algorithm employed in this study can be found in Appendix B.2. There, we also discuss discrepancies between the simplified simulation and RAPTOR, which can intuitively appreciated in Figure B.1.

Control objective. We now move to the control problem. The objective is to regulate the ι-profile $\iota(s,t)$ defined in (2.57) to a desired $\iota^\star(s)$. As ι depends on the magnetic flux gradient $\frac{\partial\psi}{\partial s}(s,t)$, the control objective can be reformulated as the regulation to a desired magnetic flux gradient profile $\frac{\partial\psi^\star}{\partial s}(s)$. Therefore, we define the output corresponding to the discretized version of the magnetic flux gradient

$$y_g = \begin{pmatrix} \frac{\psi_2-\psi_1}{\delta s_1} \\ \frac{\psi_3-\psi_1}{\delta s_1+\delta s_2} \\ \vdots \\ \frac{\psi_N-\psi_{N-1}}{\delta s_N} \end{pmatrix} = \overbrace{\begin{pmatrix} -\frac{1}{\delta s_1} & \frac{1}{\delta s_1} & 0 & \cdots & 0 & 0 & 0 \\ -\frac{1}{\delta s_1+\delta s_2} & 0 & \frac{1}{\delta s_1+\delta s_2} & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -\frac{1}{\delta s_N} & \frac{1}{\delta s_N} & 0 \end{pmatrix}}^{C} \begin{pmatrix} \psi \\ W_{th} \end{pmatrix} \tag{2.61}$$

where $y_g \in \mathbb{R}^{n_\psi}$ and $C \in \mathbb{R}^{n_\psi \times n_\psi+1}$. In few words, y_g outputs an approximated version of the flux gradient profile at a finite amount of points s_i, $i = 1, \ldots, n_\psi$. Then, we select one element of the previously defined output to define the second output $y_r = Sy_g$, where $S \in \mathbb{R}^{1\times n_\psi}$ is the selection matrix with a single element equal to one and zero elsewhere. We assume the system parameters to be uncertain, yet to belong to a known range of values. This implies we do not know their value, and we can only rely on a rough estimate. Consider κ_n a generic system's parameter. According to the previous assumption, we know that $\kappa_n \in [\underline{\kappa}_n, \bar{\kappa}_n]$. We

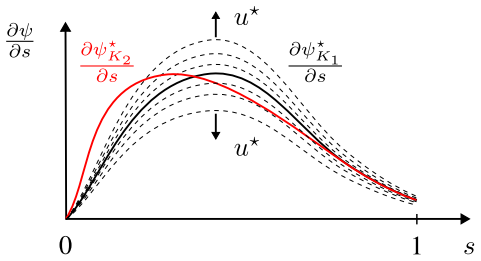


Figure 2.4: Different equilibrium solutions with different parameters.

define the vector K as the collection of all the system's parameters $K = [\kappa_1 \ \kappa_2 \ \dots \ \kappa_{n_\kappa}]^T \in \mathfrak{K} := [\underline{\kappa}_1, \bar{\kappa}_1] \times [\underline{\kappa}_2, \bar{\kappa}_2] \times \cdots \times [\underline{\kappa}_{n_\kappa}, \bar{\kappa}_{n_\kappa}] \subset \mathbb{R}^{n_\kappa}$, where n_κ is the total number of parameters. We denote by $f_K(\xi, u)$ the magnetic flux dynamics with parameters K

$$\begin{cases} \xi^+ = f_K(\xi, u) \\ y_g = C\xi, \quad y_r = Sy_g \end{cases} \tag{2.62}$$

Therefore, it is possible to obtain the equilibrium position $\xi^\star_K = [\partial \psi^{\star T}_K / \partial s \ W^\star_{th,K}]^T$ related to a certain constant input $u^\star$ and set of parameter K, solving the equation

$$f_K(\xi^\star_K, u^\star) = 0 \tag{2.63}$$

with respect to $\xi^\star_K$. Similarly, it can be experimentally obtained by simulating the system with constant input and extracting the state into which RAPTOR stabilizes.

Clearly, equilibria are parameters dependent. In other words, the equilibrium solution $\xi^\star_{K_1}$ of $f_{K_1}(\xi^\star, u^\star)$ may not be a solution of $f_{K_2}(\xi^\star, u^\star)$, as shown in Figure 2.4. Since parameters are assumed to be uncertain, integral action is vital to converge to the desired equilibrium. In addition, it robustifies the controller to mismatches between the training model and RAPTOR (and, possibly, real-world applications). Note that, due to the aforementioned mismatches, a desired reference $y^\star_{\bar{K}}$ for the RAPTOR simulator under parameter vector $\bar{K}$ may be unreachable for the simplified model.

While integral action allows the regulation of the integrated quantity to zero, asymptotic regulation can be achieved only for a number of output which is smaller or equal to the number of inputs [103]. As such, having at our disposal only one input (the two antennas act in opposite directions), we can only regulate one point in the spatial domain, corresponding to y_r. Then, we formulate the following control problem.

Problem 2.1. *Stabilize the output y_g as close as possible to the (potentially unreachable) desired set point $y^\star_{\bar{K}}$, and, at the same time, guarantee that the selected output $y_r = Sy_g$ converges to the desired value $y^\star_{r,\bar{K}} = Sy^\star_{\bar{K}}$.*

As recalled in Section 2.2, the use of integral action guarantees regulation to zero of the error $y_r - y^\star_{r,\bar{K}}$. We select the discrete-time integrator state dynamics as

$$z^+ = z + (y_r - y^\star_{r,\bar{K}})\delta t, \tag{2.64}$$

and the extended state $x = \mathrm{col}(\xi, z) \in \mathcal{X} := \mathbb{R}^{n_\psi + 2} = \mathbb{R}^{n_x}$. Therefore, the extended dynamics read as

$$x^+ = \begin{pmatrix} f_K(\xi, u) \\ z + (y_r - y^\star_{r,\bar{K}})\delta t \end{pmatrix} = \varphi_K(x, u). \tag{2.65}$$

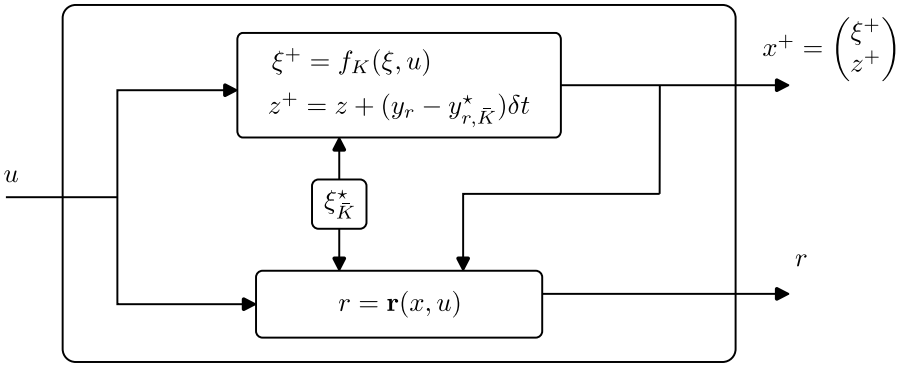


Figure 2.5: Extended tokamak environment.

Since we do not know a priori the equilibrium of the integral state (it depends on the unknown parameters K), we fix the state reference $x^\star_K = \mathrm{col}(\xi^\star_{\bar{K}}, 0)$. Then, the output of the extended system is set as $y = \mathrm{col}(y_g, z)$ and its reference to $y^\star_K = \mathrm{col}(y^\star_{\bar{K}}, 0)$.

Optimization problem. The goal is to address the problem with DRL methods. Consequently, we need to define an optimization objective. However, we do not use the blending technique proposed in Chapter 1 since even the derivation of a linear model turns out to be complex in such a framework. Nevertheless, to encourage a robust stabilizing solution, we choose a quadratic cost minimization problem. This ensures the exact local optimal solution is exponentially stabilizing, see Section 2.2.3. Then, we select a quadratic reward of the form

$$\mathbf{r}(y, u) = -(y - y^\star_K)^T Q (y - y^\star_K) - R(u - u^\star)^2, \tag{2.66}$$

where y^+ is the observed next output and $Q \in \mathbb{S}^{n_x}_{\succ 0}$ and R is a positive scalar. In particular, we choose

$$Q = \begin{pmatrix} Q_1 & \frac{\alpha_3}{2} S^\top \\ \frac{\alpha_3}{2} S & \alpha_4 \end{pmatrix} \in \mathbb{R}^{(n_\psi+1)\times(n_\psi+1)} \quad R \in \mathbb{R}. \tag{2.67}$$

with $W \in \mathbb{R}^{n_x}$ being a coupling vector and

$$Q_1 = \begin{pmatrix} \alpha_1 \mathrm{I}_{p_i - 1} & 0 & 0 \\ 0 & \alpha_2 & 0 \\ 0 & 0 & \alpha_1 \mathrm{I}_{n_\psi - p_i} \end{pmatrix} \tag{2.68}$$

being p_i the integration position and $\alpha_i > 0$ for all $i = 1, \ldots, 4$. Notice that Q_1 is built to have a different cost at the diagonal entry corresponding to the error integrated by the integral state. This gives an additional degree of freedom during the reward shaping procedure and can be used to induce the DNN to rely more on the integral state. Moreover, the cross-terms in (2.66) can be rewritten as

$$\alpha_3 \left(y - y^\star_K\right)^\top S^\top z = \alpha_3 (y_r - y^\star_{r,\bar{K}}) z = \alpha_3 \frac{z^+ - z}{\delta t} z. \tag{2.69}$$

Therefore, it penalizes the integral state variations. We recall that integral action is useful to achieve robust regulation only when the integral state reaches an equilibrium. The overall extended tokamak environment is sketched in Figure 2.5.

Algorithm 5 Tokamak DRL training algorithm

Input data:
- Initialize parameters for actor θ_0 and critic ϕ_0
- Initialize the evolution parameters $K \in \mathfrak{K}$
- select an actor-critic algorithm $\mathbf{a}$

for $k = 1$ **to** N_e **do**
  Randomly initialize the tokamak's state $\xi_0 \in \mathbb{R}^{n_\psi+1}$
  Initialize the integrator state $z_0 = 0$
  Randomly select a couple of compatible equilibrium and steady-state input $(\xi^\star_{\tilde{K}}, u^\star)$
  Compute the perturbed equilibrium $x^\star_{\tilde{K}}$ and $y^\star_{\tilde{K}}$ using (2.71)
  for $\mathrm{t} = 1$ **to** N_s **do**
    The agent draws an action $u = \boldsymbol{\pi}^{\theta_k}(x)$
    Update the state $x^+ = \varphi_K(x, u)$
    Compute the reward $\mathbf{r}(y, u)$
  end for
  Update actor parameters θ_k and critic parameters ϕ_k according to $\mathbf{a}$
end for

2.3.2 Training algorithm and simulation results

In this section, we describe the strategy used to train an agent to exploit the integral state and regulate the desired error to zero and successively present experimental results. We start from the training procedure. Its implementation is summarized in Algorithm 5. We refer to N_e and N_s as the number of episodes and episode steps in the training, respectively. This episodic learning framework is a practical necessity, allowing for a finer exploration of possible initial conditions and desired equilibria. Firstly, we select the training model parameters $K \in \mathfrak{K}$. Then, we define the *steady-state set* $\mathcal{S}^{ss}_K$ as

$$\mathcal{S}^{ss}_K = \{(\xi^\star_K, u^\star) \in \mathbb{R}^N \times [0,1] \mid f_K(\xi^\star_K, u^\star) = 0\}. \tag{2.70}$$

Then, we perturb the desired equilibrium as if it was generated by a different set of parameters $\tilde{K} \in \mathfrak{K}$. This procedure encourages generalization and the exploitation of the integral dynamics. By studying RAPTOR, we observed similarities in the shape of the flux gradient equilibria $\xi^\star$ for numerous equilibrium points, under different plasma parameters. In particular, variations in the plasma parameters corresponded to shifts of the maximum value of $\xi^\star$ while either flattening or sharpening the shape of the equilibrium function around it. Then, to avoid the time-consuming task of identifying all possible steady-state equilibria $(\xi^\star_{\tilde{K}}, u^\star) \in \mathcal{S}^{ss}_{\tilde{K}}$ for each $\tilde{K} \in \mathfrak{K}$, we reproduce a similar behavior by approximating $\xi^\star_{\tilde{K}}(s)$ by a Gaussian function having a randomly generated mean and variance

$$\xi^\star_{\tilde{K}}(s) \approx \xi^\star_K(s) + \frac{c_{\mathcal{N}}}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{s-\mu}{\sigma}\right)^2} \tag{2.71}$$

where $\mu \backsim U([0.4, 0.6])$, $\sigma \backsim U([0.8, 1.2])$ and $c_{\mathcal{N}} \backsim U([-1, +1])$ are continuous uniform random variable selected in different intervals. We choose a Gaussian function since it approaches zero at $s = 1, 0$, and we aimed to preserve the value of $\xi^\star_K$ at the plasma's center and the LCFS. This decision is influenced by the fact that the magnetic flux gradient is constantly zero at the center, while the flux gradient at the LCFS is primarily dependent on the magnetic central location, which experiences comparatively minimal variations compared to other plasma parameters. To

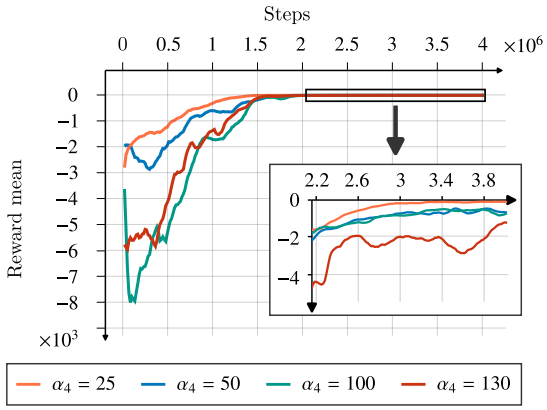


Figure 2.6: Integral weight training effect.

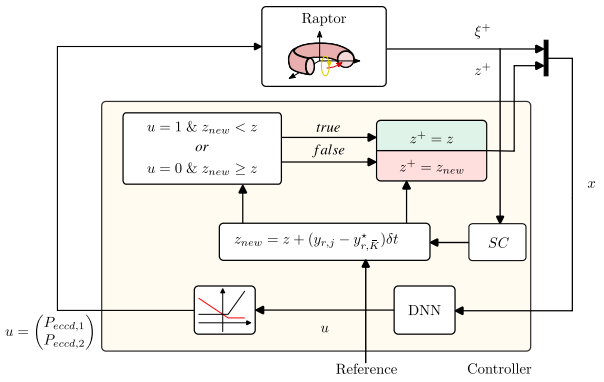


Figure 2.7: RAPTOR feedback control scheme.

conclude, the learning procedure is carried out using any actor-critic algorithm. In our experiments, we selected PPO [188]. The output of the training procedure is a trained DNN that takes as input the state x and the reference point $x^\star_K$ and returns the values of the input u to be applied.

We now move to the closed-loop simulations of the DRL feedback control law applied to the RAPTOR simulator. First, we explore the effect of the integral state cost on the training. Figure 2.6 shows the episode reward mean of four training runs with different integral cost parameters. We fix the cost parameters

$$\alpha_1 = 0.1, \quad \alpha_2 = 0.05, \quad \alpha_3 = 0.1, \quad \alpha_4 = a, \quad R = 0.01, \tag{2.72}$$

with the run-varying parameter $a \in \{25, 50, 100, 130\}$. The selection matrix S extracts the 5^{th} spatial point ($p_i = 5$), i.e., it takes the form

$$S = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & \dots & 0 \end{pmatrix}.$$

Practically, the 5^{th} spatial point corresponds to the $s = 0.2$ position. We select a position near to the plasma center since it is in the interval of the spatial domain where the measures are more reliable. Moreover, it is sufficiently near to the deposit of the ECCD current, that is $s = 0$.

We remark that α_4 is significantly bigger than $\alpha_i, \alpha_2, \alpha_3$ in all runs. Our studies showed that the controller learned to use the integrator dynamics only when the weight of integral states was significantly higher than the one of other states. Besides the fact that each component may have different magnitude since the terms are not normalized, our intuition is that when the integral state cost is not significant compared to the one of other states, the optimization process focuses on stabilizing those states. This also reduces the integrator cost. However, while a low running cost from the integral state may be tolerable for an approximate optimal solution, it indicates that the integrator remains unstable and fails in correctly regulating the desired output. Therefore, setting the weight of integral states higher than that of other states encourages learning of a forwarding-like controller, see Section 2.2. On the flip side, Figure 2.6 demonstrates that a higher weight assigned to integral states results in slower convergence. The intuition is that if the weight is too high, the learning process slows down because the controller focuses solely on the integrator. However, no direct action can be taken on the integrator dynamics due to the cascade structure of the extended system. As a result, the gradients of the cost with respect to the agent actions are small and the network requires more time to converge to an effective solution.

Once the DNN is trained to control the extended system, we test it on RAPTOR. We recall that the training model comes from the simulation algorithm described in Appendix B.2, and thus it is an approximation of the complex dynamics computed by RAPTOR. Moreover, as integral action paired with input saturation may generate overshoots, we add an anti-windup layer. This additional layer is not present at training time as the training procedure is run using the linear integrator dynamic described by (2.64). Consequently, the training and test models have non-negligible discrepancies.

The simple anti-windup scheme is implemented by modifying the integrator linear dynamics (2.64) into

$$z^+ = \begin{cases} z & \text{if } u = 1 \text{ and } y_{r,j} - y^\star_{r,\bar{K}} < 0 \\ z & \text{if } u = 0 \text{ and } y_{r,j} - y^\star_{r,\bar{K}} \geqslant 0 \\ z + (y_{r,j} - y^\star_{r,\bar{K}})\delta t. & \text{otherwise} \end{cases} \tag{2.73}$$

We refer to [213] for a survey on anti-windup technique. Figure 2.7 shows the closed-loop with the anti-windup scheme. We set the RAPTOR parameters as the ones in (B.15)-(B.16). Target ι-profiles, corresponding to steady-state plasma configurations $(\xi^\star_K, u^\star) \in \mathcal{S}^{ss}_K$, are generated by applying a constant input for a sufficiently long time in the RAPTOR simulator. During the simulations, the ramp-up phase lasts until $t = 0.02$ s bringing the central current from $I_p = 80$ kA to $I_p = 120$ kA, while the feedback controller is activated at $t = 0.1$ s. During the flat-top phase, four different target profiles are given to the controller: the first at $t = 0.1$ s, the second at $t = 2.5$ s, the third at $t = 5$ s, and the last at $t = 7.5$ s. To test the robustness of the DNN-based control feedback, we set up three different control scenarios:

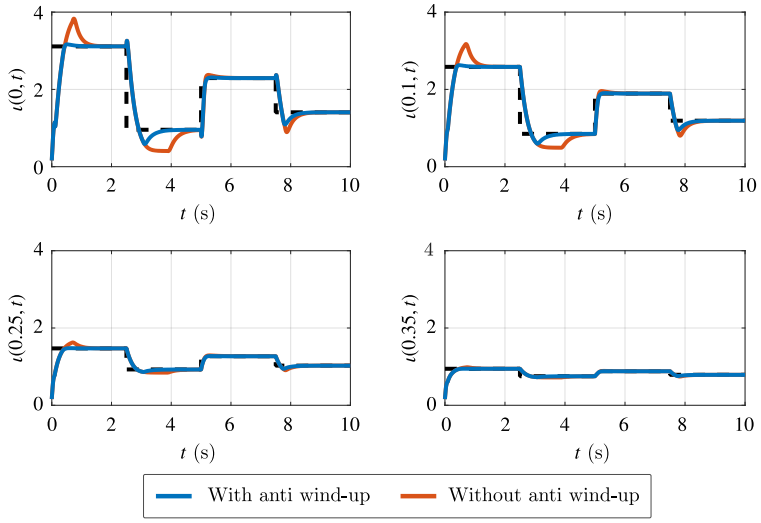
1st scenario. We compare neural controller performances with and without the anti-windup layer (implemented only in the testing environment). The integration position $s = 0.2$ is not changed between training and test. Results are shown in Figure 2.8a, which presents the ι-profile evolution at four locations of the spatial domain $s \in \{0,\ 0.1,\ 0.25,\ 0.35\}$ during a simulation time of length $T_{sim} = 10$ s. Figure 2.9a and 2.9c show the ι-profiles at different time instants $t \in \{2.5,\ 2.7,\ 2.9,\ 3.1,\ 3.3\}$ and Figures 2.9b and 2.9d present the feedback control input $u = \mathrm{col}(P_{eccd,1},\ P_{eccd,2})$ during the simulation interval.

2nd scenario. We compare the control performances of the anti-windup neural controller under input perturbations. The integration position $s = 0.2$ is not changed between training and test. We introduce an input disturbance at time $t = 5.3$ s, implemented in RAPTOR via a Neutral-Beam Injector with $\rho_{dep} = 0.4$ and $w_{cd} = 0.4$. Both parameters ρ_{dep}, w_{cd} are set differently from the ones of the ECCD antenna in (B.16). Figure 2.8b depicts the ι-trajectories in four points of the spatial domain. The corresponding profile is presented in Figure 2.9e and the input is depicted in Figure 2.9f.

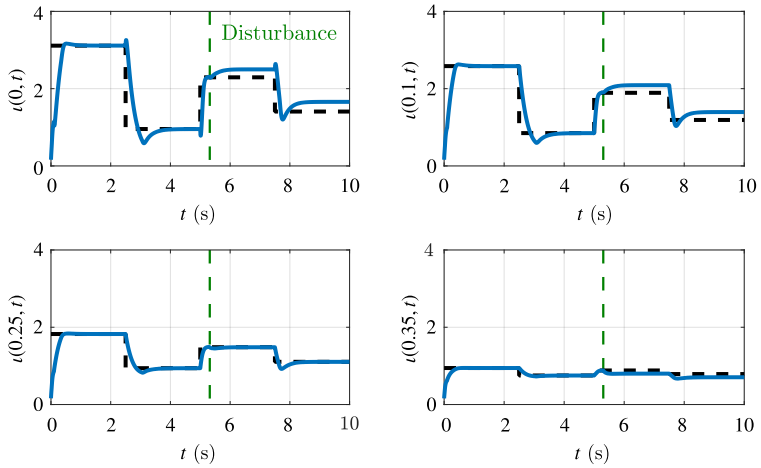
3rd scenario. We study the robustness with respect to integration point of the anti-windup neural controller. The integration position is changed from the 5^{th} element (training) to the 3^{rd} (test), i.e., from $s = 0.2$ to $s = 0.1$. Figure 2.8c shows the ι trajectories in four points of the spatial domain, and the corresponding input is depicted in Figure 2.9h. Figure 2.9g shows the ι-profiles at five different time instants.

In the 1^{st} scenario, we see from Figure 2.8a that the addition of anti-windup to the closed-loop dynamics does not hinder the tracking capabilities of the controller, thus showing its robustness.

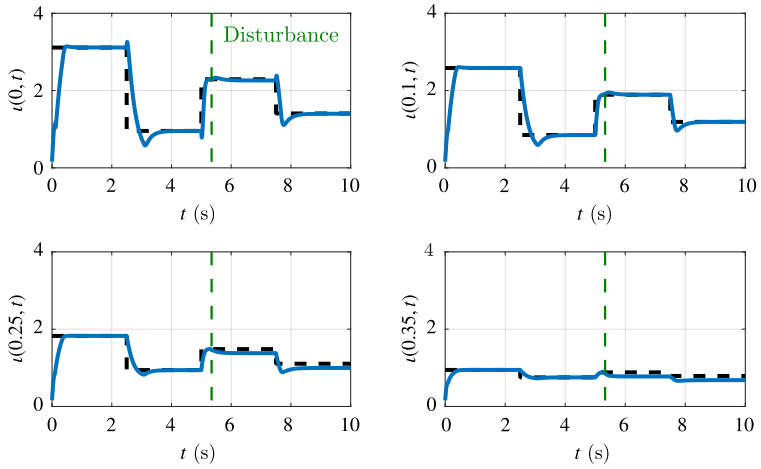
Moreover, we can observe strong overshoot when the anti-windup layer is turned off, corresponding to a slow integrator dynamics. This is expected, the integrator continues to evolve even if the input is saturated. Using the nonlinear integrator dynamics in (2.73), we prevent the integrator state from evolving in the wrong direction in case of input saturation. This leads to faster response and reduced overshoots. Comparing Figures 2.9a and 2.9c, we can remark that the ι-profile in the case of anti-windup implementation is converging faster to the desired profile than in the case of linear integrator dynamics. For the 2^{nd} scenario, we remark from Figure 2.8b that the error at the integration point $s = 0.2$ is regulated to zero, even in presence of the disturbance. As for the previous scenario, this shows the effectiveness of integral action in providing robust regulation. The 3^{rd} scenario shows the controller robustness to small variations in the integration point. Indeed, Figure 2.8c shows that, once the disturbance is injected into the system, the error is regulated to zero at the new position $s = 0.1$.



(a) ι-profile trajectories with and without anti-windup.

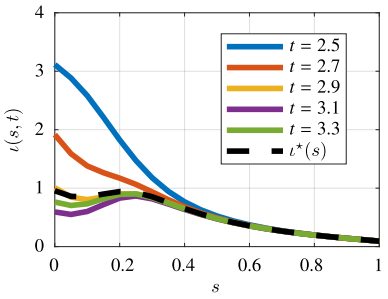


(b) ι-profile trajectories with input disturbance at $t = 5.3$ s.

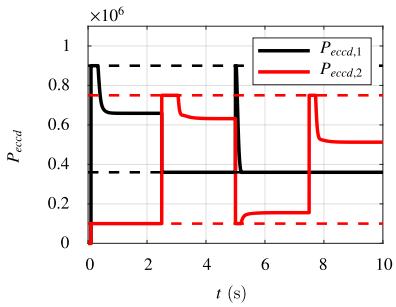


(c) ι-profile trajectories with disturbance and integration point $s = 0.1$.

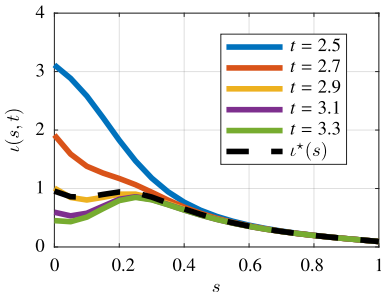
Figure 2.8: Evolution of ι-profiles in four points of the spatial domain



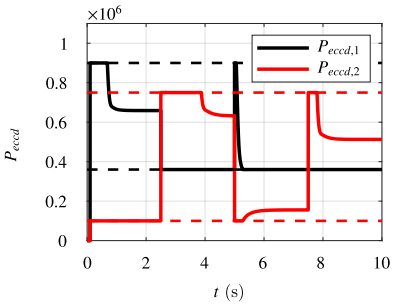
(a) ι-profile with anti-windup.



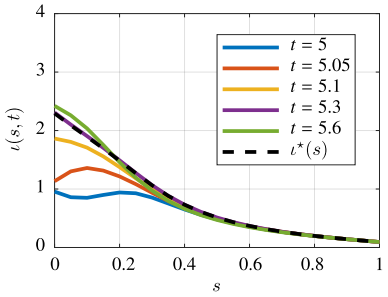
(b) Applied input with anti-windup.



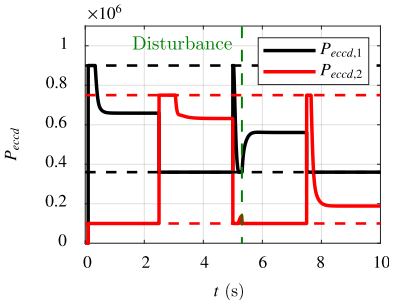
(c) ι-profile without anti-windup.



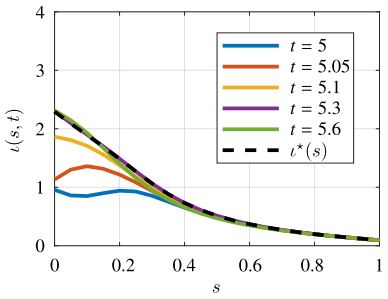
(d) Applied input without anti-windup.



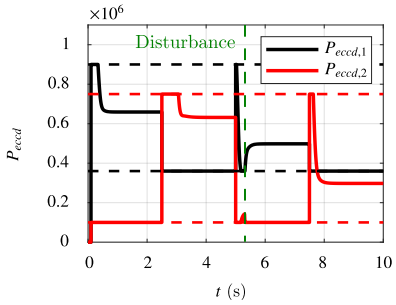
(e) ι-profile with disturbance.



(f) Applied input with disturbance.



(g) ι-profile with shifted integration point.



(h) Applied input with shifted integration point.

Figure 2.9: ι-profiles at five time instants and applied inputs.

Part II

Contraction as an optimization problem

Introduction

In Part I, we explored how control theoretic results can improve modern machine learning approaches for system control. As we proceed to the second part of this manuscript, our attention shifts towards the complementary perspective. In other words, we examine how optimization and machine learning tools can enhance control theoretic designs by providing approximate solutions to complex problems that would otherwise be intractable. In particular, we focus on results based on incremental stability obtained via contraction theory.

In simple terms, a system exhibits incremental stability properties if any two trajectories originating from different yet "close" initial conditions remain "close" throughout their time of existence. If the distance between these trajectories decreases over time, the system is incrementally asymptotically stable (see, e.g. [3, 5, 9, 10, 22, 57, 72, 109, 134, 166, 178, 197, 203, 215, 216]). Contraction theory, which involves the shrinkage of the length of *any* smooth curve between two points [6], was first proposed several decades ago and has recently gained significant attention within the control community [87, 107, 219]. Its applications have proven effective in various control problems, including observer design [181, 239], output regulation [166], and multi-agent synchronization [4, 7, 200]. Incremental stability is closely related to the notion of contraction and their strong connection has been established, e.g. [83, 216].

Chapter 3 marks this manuscript's transition from equilibrium stabilization to incremental stability in discrete-time. There, we introduce novel robust incremental stability tools based on discrete-time contraction and we address the multi-agent framework. The contractive control law is obtained via the solution of a convex generalized eigenvalue problem. While DNNs will not be the focus of this third chapter, we show that the design can be cast as an optimization problem, thus opening possible developments based on data-driven methods and machine learning tools. The results of this last chapter are partially covered in [S6].

Expanding on the ideas presented in Chapter 3, our subsequent focus revolves around discovering neural feedback designs that approximate control laws making the closed-loop incrementally stable via contraction theory. Among the numerous tools proposed to analyze contractivity properties of dynamical systems, such as matrix measures based on both Euclidean and non-Euclidean norms [3, 57, 203], Riemannian metrics have emerged as a noteworthy approach, offering a general and intuitive verification method [5, 6, 134, 197]. Very similar properties have been also studied in the context of convergent theory [163, 166, 178], resulting in conditions which are mainly equivalent to those studied in the context of Riemannian metrics (also denoted as Demidovich conditions [163]). More recently, some works focused on the design of feedback controllers making a closed-loop system a contraction, see, e.g. [83, 140, 181, 242]. Unfortunately, when applied on general nonlinear systems, most of these Riemannian metric-based designs require finding solutions to nonlinear partial differential inequalities. This task often poses challenges and typically necessitates the use of numerical methods to obtain approximate solutions. Consequently, only a limited number of studies have proposed general approaches to address this problem [58, 110, 139, 209, 219, 227, 228].

In Chapter 4, our primary objective is to develop practical methods for contractive feedback design through the use of DNNs. Given the more mature literature on contraction theory in the continuous-time domain, we will consider continuous-time nonlinear systems. Addressing the scenario of Riemannian metric-based conditions, we propose a DNNs-based algorithm to approximate controllers with infinite-gain margin properties, and show their effectiveness in multi-agent synchronization. Thanks to the strong theoretical foundations these controllers are

built on, they will preserve fundamental robustness guarantees even when approximated by DNNs. These results are part of [S2]. Successively, we improve the proposed technique and apply it the output tracking problem [S8].

It is worth noting that some works already proposed generalizations to the concept of contraction [151, 236]. However, related feedback designs have yet to appear in the literature, due to the complexity of the required conditions. Such a complexity also affects their practical solvability via optimization methods, as the number of variables to be estimated rapidly explodes for most pairs of system dimension and contraction order. Hence, we conclude this second part by introducing the general notion of k-contraction, which we subsequently reformulate and investigate to derive design-friendly conditions. Our aim is to provide a broader understanding of contraction analysis for dynamical systems, and opening new possibilities for future designs of guaranteed neural controllers. Compared to existing results, our conditions scale nicely with the system dimension and contraction order, thus allowing for an effective use of convex optimization approaches on a significantly wider range of situations. The results are presented in Chapter 5 and they are extracted from [S7].

3 Discrete-time contractive feedback design

In Part I, our focus was on studying the robust stability of equilibrium points, which is fundamental in many control applications. However, there exist numerous control scenarios where convergence to a trajectory is required instead of an equilibrium. Notable examples include output tracking, where a system must follow a desired trajectory, and multi-agent synchronization, where multiple agents need to coordinate and converge to the same solution [103, 166].

In this chapter, we shift our attention to studying robust convergence to trajectories in the discrete-time nonlinear framework. Our main analysis tool will be the concept of incremental stability, specifically, incremental Input-to-State Stability (δISS). Intuitively, incrementally stable dynamics exhibit the intriguing property of "forgetting" their initial conditions while offering robustness against signals entering in the input directions. This feature makes incremental stability a useful property, since the vanishing effect of initial conditions implies that distances between trajectories asymptotically decrease in time, ensuring robust convergence to a (possibly unbounded) unique solution.

The concept of discrete-time incrementally stable dynamics has garnered attention from the research community in the last two decades [22, 78, 88, 194, 215]. Typically, δISS is obtained via incremental Lyapunov functions. More recently, paralleling the continuous-time framework, the equivalence between δISS properties with exponential convergence rate and contractive dynamics has been shown also in the discrete-time scenario [216]. However, existing results on discrete-time contraction analysis focus on smooth dynamics [55,109,110,165,216]. In discrete-time, this limitation becomes particularly relevant when dealing with machine learning tools such as Deep Neural Networks (DNNs), where non-smooth functions (e.g., ReLU activations) are typically involved.

Hence, in this chapter, we extend the results to encompass dynamics that are time-varying and not differentiable everywhere. Additionally, we address the lack of *closed-form* solutions in existing feedback designs based on discrete-time contraction, which typically rely on optimization problems [110, 227, 228]. By providing closed-form state-feedback controllers, we offer a constructive design that ensures uniform exponential incremental stability properties for the closed-loop system. This approach not only enhances the practicality and efficiency of implementation (compared, e.g., to approaches relying on estimation of shortest-paths, i.e. geodesics [228]) but also provides insights into the optimality and gain margin properties of the proposed controllers. More specifically, we draw parallels to results in the continuous-time framework [84], and we

demonstrate the optimality of the proposed control law with respect to a quadratic cost. These findings lay a strong foundation for future developments based on learning approaches. We also highlight the gain margin properties of the controller, emphasizing the key differences from the continuous-time solution.

Lastly, building upon the aforementioned results, we address the problem of multi-agent synchronization. We start by generalizing the results for the linear framework, as it serves as an instrumental analysis for the nonlinear scenario. By establishing a clear link between linear and nonlinear approaches, we leverage the proposed incremental stability tools to derive a distributed control law that solves the synchronization problem for a general network of identical nonlinear agents. The results discussed in this chapter are partially covered in [S6].

3.1 Incremental stability via non-smooth contraction

As previously stated, the existing body of literature in the domain of discrete-time contraction analysis has primarily concentrated on the study of systems governed by smooth dynamics. However, practical discrete-time models frequently exhibit non-differentiable nonlinearities, such as saturations, which demand an extension of the analytical framework. Hence, in this section, we generalize existing results to a particular class of non-smooth dynamics. We show the link between such a class of systems and common non-smooth dynamics (e.g. piece-wise linear) and we proposed a contraction-based closed-form control law making the closed-loop δISS with exponential convergence rate. Furthermore, we analyze the gain margin properties and optimality of these controllers, providing valuable insights into their performance and robustness characteristics.

We start our analysis by recalling some definitions of incremental stability coming from the literature, e.g. [22, Definition 1], [215, Definition 1]. At first, we consider unforced time-varying systems and introduce the concept of asymptotic incremental stability. Consider a discrete-time nonlinear system evolving according to the difference equation

$$x^+ = f(x, \mathfrak{t}), \tag{3.1}$$

with $x \in \mathbb{R}^{n_x}$ and $f : \mathbb{R}^{n_x} \times \mathbb{N} \to \mathbb{R}^{n_x}$ a continuous function of the state.

Definition 3.1 (Asymptotic incremental stability)**.** *System* (3.1) *is globally incrementally asymptotically stable (δGAS) if there exists $\boldsymbol{\beta} \in \mathcal{KL}$ such that, for all $\mathfrak{t} \geqslant \mathfrak{t}_0$ with $\mathfrak{t}_0 \in \mathbb{N}$ and for any initial states $x_1(\mathfrak{t}_0), x_2(\mathfrak{t}_0)$, the resulting solutions $x_1(\mathfrak{t}), x_2(\mathfrak{t})$ of* (3.1) *satisfy*

$$|x_1(\mathfrak{t}) - x_2(\mathfrak{t})| \leqslant \boldsymbol{\beta}(|x_1(\mathfrak{t}_0) - x_2(\mathfrak{t}_0)|, \mathfrak{t}).$$

The function $\boldsymbol{\beta}$ describes the rate at which trajectories converge to one another. If $\boldsymbol{\beta}$ represents an exponential decay and it is uniform in the initial conditions, namely if there exists scalars $\beta > 0$ and $\rho \in (0, 1)$ such that $\boldsymbol{\beta} = \beta\rho^{\mathfrak{t}-\mathfrak{t}_0}|x_1(\mathfrak{t}_0) - x_2(\mathfrak{t}_0)|$, the system is said to be globally uniformly incrementally exponentially stable (δGES).

For differentiable system dynamics, δGES properties have recently been shown to be equivalent to contraction properties [216, Theorem 15]. In continuous-time, it is known that contractivity can be studied by exploiting sufficient Euclidean metric-based conditions, also known as Demidovich conditions [163]. Similar results recently appeared also for the discrete-time framework,

thus generalizing the well-known Lyapunov stability inequality for linear systems to the nonlinear scenario. These sufficient conditions are recalled in the following theorem [216, Theorem 14].

Theorem 3.1 (Discrete-time Demidovich conditions.)**.** *Consider system* (3.1) *and suppose f is C^1 in its first argument. Moreover, suppose there exists a positive definite matrix $P \in \mathbb{S}^{n_x}_{\succ 0}$ and a scalar $\rho \in (0,1)$ such that*

$$\frac{\partial f}{\partial x}(x,\mathfrak{t})^\top P \frac{\partial f}{\partial x}(x,\mathfrak{t}) \preceq \rho^2 P,$$

for all $(x,\mathfrak{t}) \in \mathbb{R}^{n_x} \times \mathbb{N}$. Then, system (3.2) *is δGES.*

We now move to the analysis of systems with inputs and we formally define incremental input-to-state stability. This definition clearly highlights the robustness properties required in δISS dynamics. Consider a system of the form

$$x^+ = f(x,w,\mathfrak{t}), \tag{3.2}$$

where $w \in \mathbb{R}^{n_w}$ is a disturbance input and $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_w} \times \mathbb{N} \to \mathbb{R}^{n_x}$ is a continuous function of state and input.

Definition 3.2 (Incremental Input-to-State Stability)**.** *System* (3.2) *is globally incrementally input-to-state stable (δISS) if there exists $\boldsymbol{\beta} \in \mathcal{KL}$ and $\boldsymbol{\gamma} \in \mathcal{K}_\infty$ such that, for all $\mathfrak{t} \geqslant \mathfrak{t}_0$ with $\mathfrak{t}_0 \in \mathbb{N}$, for any initial states $x_1(\mathfrak{t}_0), x_2(\mathfrak{t}_0)$ and any pair of disturbance sequences $\mathfrak{t} \mapsto w_1(\mathfrak{t}), \mathfrak{t} \mapsto w_2(\mathfrak{t})$, the resulting solutions $x_1(\mathfrak{t}), x_2(\mathfrak{t})$ of* (3.1) *satisfy*

$$|x_1(\mathfrak{t}) - x_2(\mathfrak{t})| \leqslant \boldsymbol{\beta}(|x_1(\mathfrak{t}_0) - x_2(\mathfrak{t}_0)|, \mathfrak{t}) + \boldsymbol{\gamma}(\sup_{t \in [\mathfrak{t}_0, \mathfrak{t}]} |w_1(t) - w_2(t)|).$$

As for the unforced case, we particularize the property to the case of exponential uniform convergence.

Definition 3.3 (Incremental exponential Input-to-State Stability)**.** *System* (3.2) *is globally uniformly incrementally input-to-state stable with exponential convergence rate (exponentially δISS) if there exist $\beta, \gamma > 0$ and $\rho \in (0,1)$ such that, for all $\mathfrak{t} \geqslant \mathfrak{t}_0$ with $\mathfrak{t}_0 \in \mathbb{N}$ and for any initial states $x_1(\mathfrak{t}_0), x_2(\mathfrak{t}_0)$ and any pair of disturbance sequences $\mathfrak{t} \mapsto w_1(\mathfrak{t}), \mathfrak{t} \mapsto w_2(\mathfrak{t})$, the resulting solutions $x_1(\mathfrak{t}), x_2(\mathfrak{t})$ of* (3.4) *satisfy*

$$|x_1(\mathfrak{t}) - x_2(\mathfrak{t})| \leqslant \beta \rho^{\mathfrak{t}-\mathfrak{t}_0} |x_1(\mathfrak{t}_0) - x_2(\mathfrak{t}_0)| + \sup_{t \in [\mathfrak{t}_0, \mathfrak{t}]} \gamma |w_1(t) - w_2(t)|\,. \tag{3.3}$$

3.1.1 Sufficient conditions for exponential δISS

While Theorem 3.1 represents a significant contribution to the field of discrete-time nonlinear system analysis, its applicability is limited to scenarios with continuously differentiable dynamics. This restriction excludes important classes of systems found in practice, such as those modeled by DNNs employing ReLU activation functions. These systems often exhibit non-differentiable behaviors, necessitating an extension of the existing results. Moreover, the result provides incremental stability properties only for the autonomous scenario. In light of these limitations, our objective is to propose a generalization that addresses the challenges posed by non-differentiable discrete-time nonlinear systems with inputs. We generalize the results of Theorem 3.1 to a class of non-differentiable and non-autonomous system dynamics. Then, we

establish a connection between the selected non-smooth representation and well-established non-smooth dynamics. Building upon these results, we develop a closed-form control law that guarantees exponential δISS properties for the closed-loop system. Consider time-varying discrete-time nonlinear system of the form

$$x^+ = \varphi(x, \mathfrak{t}) + w, \tag{3.4}$$

where the function $\varphi : \mathbb{R}^{n_x} \times \mathbb{N} \to \mathbb{R}^{n_x}$ is such that the following mild property holds.

Property 3.1. *Function $\varphi : \mathbb{R}^{n_x} \times \mathbb{N} \to \mathbb{R}^{n_x}$ is continuous in its first argument and there exists a (possibly unbounded) set of matrices $\mathcal{D}\varphi \subset \mathbb{R}^{n_x \times n_x}$ such that, for each $x_a, x_b \in \mathbb{R}^{n_x}$ and all $\mathfrak{t} \in \mathbb{N}$, there exists an integrable function $\psi : [0, 1] \to \mathbb{R}^{n_x \times n_x}$ satisfying*

$$\varphi(x_a, \mathfrak{t}) - \varphi(x_b, \mathfrak{t}) = \int_0^1 \psi(s)\, ds\, (x_a - x_b) \tag{3.5a}$$

$$\psi(s) \in \mathcal{D}\varphi, \quad \forall s \in [0, 1]. \tag{3.5b}$$

The above definition allows considering a wide class of dynamical systems. First, note that when $n_x = 1$, Property 3.1 boils down to the requirement of φ being absolutely continuous. Trivially, such a class of systems includes continuously differentiable ones with $\mathcal{D}\varphi$ containing all of their Jacobians. Moreover, Property 3.1 includes functions that are differentiable almost everywhere (i.e., everywhere but on a set of measure zero), such as piecewise smooth and Lipschitz functions. In this case, $\mathcal{D}\varphi$ contains all the possible Clarke generalized gradients [50]. As a particular case, for linear systems of the form $x^+ = Ax + w$, $\mathcal{D}\varphi = \{A\}$. When moving to nonlinear systems, this allows the inclusion of some useful nonlinearities, such as saturations and arctangents, by selecting $\mathcal{D}\varphi$ as the vertices of the convex hull of all possible Jacobians. Finally, systems satisfying Property 3.1 are linked to the class of piece-wise smooth continuous (PWSC) dynamics, which saw the development of robust synchronization via contraction-based designs in the continuous-time framework [63, 64]. In what follows, we show the relation between these two classes of systems. Consider the unforced dynamics with $w = 0$. PWSC dynamics are defined via a *finite* collection of open, disjoint, and nonempty sets $\mathcal{X}_1, \ldots, \mathcal{X}_p$. By defining $\partial\mathcal{X}_i$ the boundary of the set $\mathcal{X}_i$ and by $\overline{\mathcal{X}}_i$ its closure, i.e., $\overline{\mathcal{X}}_i = \mathcal{X}_i \cup \partial\mathcal{X}_i$, a discrete-time PWSC system is described by the following dynamics

$$x^+ = \Phi(x, \mathfrak{t}) = \begin{cases} \varphi_1(x, \mathfrak{t}) & \forall x \in \overline{\mathcal{X}}_1 \\ \vdots & \\ \varphi_p(x, \mathfrak{t}) & \forall x \in \overline{\mathcal{X}}_p \end{cases}, \tag{3.6}$$

where $\varphi_i : \mathcal{X}_i \to \mathbb{R}^{n_x}$ is a differentiable function, $x \in \mathbb{R}^{n_x}$ is the state at time $\mathfrak{t} \in \mathbb{N}$. Moreover, these systems satisfy the following:

- The whole space $\mathbb{R}^{n_x}$ can be covered, namely $\cup_{i=1}^p \overline{\mathcal{X}}_i \subseteq \mathbb{R}^{n_x}$
- For any i, j, the intersection $\Sigma_{ij} := \overline{\mathcal{X}}_i \cap \overline{\mathcal{X}}_j$ is either an $\mathbb{R}^{n_x - 1}$ manifold included in $\partial\mathcal{X}_i$ and $\partial\mathcal{X}_j$ or the empty set.
- Each φ_i is C^1 for all $x \in \mathcal{X}_i$ and its Jacobian $\dfrac{\partial \varphi_i}{\partial x}$ can be continuously extended on the boundary $\partial\mathcal{X}_i$
- Φ is C^0 for all $x \in \mathbb{R}^{n_x}$

- For any i, j such that Σ_{ij} is not empty, it holds $\varphi_i(x, \mathfrak{t}) = \varphi_j(x, \mathfrak{t})$ for all $\mathfrak{t}$ and all $x \in \Sigma_{ij}$

Given these properties, the relation between PWSC dynamics and systems satisfying Property 3.1 is presented in the following proposition.

Proposition 3.1. *If, for any couple $(x^a, x^b) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_x}$, the dynamics of* (3.6) *switches a countable (possibly infinite) number of times $\tau \in \mathbb{N}$ along the line $\gamma(s) = sx^a - (1-s)x^b$, then system* (3.6) *satisfies Property 3.1.*

Proof. Consider the points x^a and x^b at time $\mathfrak{t} \in \mathbb{N}$. First, note that if the line $\gamma(s) = sx^a + (1-s)x^b$ lies completely on Σ_{ij}, $\Phi(\gamma(s)) = \varphi_i(\gamma(s))$ for all $s \in [0, 1]$. Indeed, since by assumption $\varphi_i \in C^1$ at the boundaries for all $i = 1, \dots, p$, we can arbitrarily choose the dynamics of γ. In other words, we can assume the whole line to belong to the same set. As such, the number of dynamics switches at time $\mathfrak{t}$ is $\tau = 0$, and the system behaves as a continuously differentiable one. Then, we focus on the case when $\tau > 0$. We identify by $x_q = \gamma(s_q)$ the states where $\gamma(s)$ is forced to change dynamics for the q-th time. In other words, $s_q \in (0, 1)$ is such that $\lim_{s \to s_q^-} \gamma(s) \in \overline{\mathcal{X}}_i$ and $\lim_{s \to s_q^+} \gamma(s) \in \mathcal{X}_j$. Note that, since x_q lies at the intersection between two sets, we can arbitrarily assume its evolution to follow either the i-th or the j-th dynamics. Then, by defining $x_0 := x_b$ and $x_{\tau+1} := x_a$, we can assume the following without loss of generality

$$\text{if } \tau = 0 : \gamma(s) \in \overline{\mathcal{X}}_i \ \forall s \in [0,1], \qquad \text{if } \tau > 0 : \begin{cases} x_0 \in \overline{\mathcal{X}}_{h_1} & \\ x_{\tau+1} \in \overline{\mathcal{X}}_{h_{\tau+1}} & \\ \gamma(s) \in \mathcal{X}_{h_q} & \forall s \in (s_{q-1}, s_q), \\ & q = 1, \dots, \tau+1 \\ x_q \in \Sigma_{h_q h_{q+1}} & q = 1, \dots, \tau \end{cases} \tag{3.7}$$

where we used the compact notation $h_i := h(x_i)$ and $h : \mathbb{R}^{n_x} \to \{1, \dots, p\}$ is a function linking each point of the state space to a set. Differently put, we can subdivide the line connecting x_0 to $x_{\tau+1}$ into a set of collinear and consecutive segments, each one belonging to a single set. Then, since $\tau \in \mathbb{N}$ (i.e., the set of segments is countable) the set of scalars $\mathcal{S} = \{s_1, \dots, s_\tau\}$ is of Lebesgue measure zero and, for any $q = 1, \dots, \tau+1$ and any $i = 1, \dots, p$, the following holds

$$x_a - x_b = \sum_{q=1}^{\tau+1} (x_q - x_{q-1}) \,, \tag{3.8a}$$

$$\lim_{(c_1, c_2) \to (s_{q-1}^+, s_q^-)} \int_{c_1}^{c_2} \frac{\partial \varphi_i}{\partial x}(\gamma(s), \mathfrak{t}) ds = \int_{s_{q-1}}^{s_q} \frac{\partial \varphi_i}{\partial x}(\gamma(s), \mathfrak{t}) ds \,, \tag{3.8b}$$

By the fundamental theorem of calculus applied to the function $G_q(s) = \varphi_{h_q}(\gamma(s), \mathfrak{t})$, equation (3.8b) leads to

$$\varphi_{h_q}(x_q, \mathfrak{t}) \varphi_{h_q}(x_{q-1}, \mathfrak{t}) = G_q(s_q) - G_q(s_{q-1}) = \int_{s_{q-1}}^{s_q} \frac{\partial \varphi_{h_q}}{\partial x}(\gamma(s), \mathfrak{t}) \, ds \; (x_a - x_b) \,. \tag{3.9}$$

Consider the points $x_a^+ = \Phi(x_a, \mathfrak{t})$ and $x_b^+ = \Phi(x_b, \mathfrak{t})$ at time $\mathfrak{t} + 1$. Since $\tau \in \mathbb{N}$, by (3.7) we

have

$$x_a^+ - x_b^+ = \sum_{q=1}^{\tau+1} [\Phi(x_q, \mathfrak{t}) - \Phi(x_{q-1}, \mathfrak{t})] = \sum_{q=1}^{\tau+1} \left[\varphi_{h_q}(x_q, \mathfrak{t}) - \varphi_{h_q}(x_{q-1}, k)\right]$$
$$= \left[\sum_{q=1}^{\tau+1} \int_{s_{q-1}}^{s_q} \frac{\partial \varphi_{h_q}}{\partial x}(\gamma(s), \mathfrak{t})\, ds\right] (x_a - x_b)\,. \tag{3.10}$$

For each $q = 1, \dots, \tau$, define the sets

$$\mathcal{D}\Phi_q := \left\{ \frac{\partial \varphi_{h_q}}{\partial x}(x_q, \mathfrak{t}), \frac{\partial \varphi_{h_{q+1}}}{\partial x}(x_q, \mathfrak{t}) \right\} \qquad \mathcal{D}\Phi := \left\{ \frac{\partial \varphi_1}{\partial x}, \dots, \frac{\partial \varphi_p}{\partial x} \right\}.$$

By choosing the function $\psi(s)$ in Proposition 3.1 such that, for all $\mathfrak{t} \in \mathbb{N}$, it satisfies

$$\begin{cases} \psi(s) = & \dfrac{\partial \varphi_{h_q}}{\partial x}(\gamma(s), \mathfrak{t}) \quad \forall s \in (s_{q-1}, s_q), \\ \psi(s_q) \in & \mathcal{D}\Phi_q \\ \psi_k(s_0) = & \dfrac{\partial \varphi_{h_1}}{\partial x}(x_b, \mathfrak{t}) \\ \psi(s_{\tau+1}) = & \dfrac{\partial \varphi_{h_{\tau+1}}}{\partial x}(x_a, \mathfrak{t}) \end{cases}$$

equality (3.10) reads

$$\Phi(x_a, \mathfrak{t}) - \Phi(x_b, \mathfrak{t}) = \int_0^1 \psi(s)\, ds\ (x_a - x_b)\,.$$

with $\psi(s) \in \mathcal{D}\Phi$ for all $s \in [0, 1]$, and this concludes the proof. □

We now generalize existing results to the framework of non-smooth dynamics whose vector fields satisfy Property 3.1. We prove below a sufficient δISS condition extending the results in [55, Theorem 2], [216, Theorem 14], [78, Theorem 6.1] to the case of time-varying non-smooth vector fields satisfying Property 3.1.

Proposition 3.2. *Consider system* (3.4) *and suppose that φ satisfies Property 3.1 with a specific set-valued map $\mathcal{D}\varphi$. Moreover, suppose that there exists $P \in \mathbb{S}^{n_x}_{\succ 0}$ and $\rho \in (0, 1)$ satisfying*

$$J^\top P J \preceq \rho^2 P\,, \quad \forall J \in \mathcal{D}\varphi. \tag{3.11}$$

Then, system (3.4) *is exponentially δISS according to Definition 3.3.*

Proof. Consider the candidate Lyapunov function $V : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbb{R}_{\geqslant 0}$ defined as

$$V(x_1, x_2) = (x_1 - x_2)^\top P (x_1 - x_2)\,,$$

for any two states $x_1, x_2 \in \mathbb{R}^{n_x}$. Given any selection of x_1, x_2 and $w_1, w_2 \in \mathbb{R}^{n_x}$, define function $F : \mathbb{R} \to \mathbb{R}^{n_x}$ as

$$F(s) = \varphi(sx_1 + (1-s)x_2, \mathfrak{t}) + (sw_1 + (1-s)w_2)\,.$$

We have

$$x_1^+ - x_2^+ = F(1) - F(0) = \varphi(x_1, \mathfrak{t}) - \varphi(x_2, \mathfrak{t}) + (w_1 - w_2)\,.$$

In view of (3.5), we obtain

$$V(x_1^+, x_2^+) = (x_1^+ - x_2^+)^\top P \left[\int_0^1 \psi(s)\, ds\, (x_1 - x_2) + (w_1 - w_2) \right] \tag{3.12}$$

for some $\psi(s) \in \mathcal{D}\varphi$, for all $s \in [0,1]$. Then, adding and subtracting $\rho^2 V(x_1, x_2)$ and $V(x_1^+, x_2^+)$ to the right-hand side of (3.12) yields

$$\begin{aligned} V(x_1^+, x_2^+) - \rho^2 V(x_1, x_2) &= 2(x_1^+ - x_2^+)^\top P \left[\int_0^1 \psi(s)\, ds\, (x_1 - x_2) + (w_1 - w_2) \right] \\ &\quad - (x_1^+ - x_2^+)^\top P (x_1^+ - x_2^+) \int_0^1 ds - \rho^2 (x_1 - x_2)^\top P (x_1 - x_2) \int_0^1 ds \\ &= \int_0^1 \xi^\top \Upsilon(s) \xi \, ds + 2(x_1^+ - x_2^+)^\top P (w_1 - w_2) \end{aligned}$$

where we defined $\xi := \operatorname{col}(x_1 - x_2, x_1^+ - x_2^+)$ and

$$\Upsilon(s) := \begin{pmatrix} -\rho^2 P & \psi^\top(s) P \\ P\psi(s) & -P \end{pmatrix}.$$

By performing steps similar to the ones in Theorem 2.1, due to (3.11) and a Schur complement, $\Upsilon(s) \preceq 0$ for all $s \in [0,1]$. As a consequence, since $\rho \in (0,1)$, we obtain

$$V(x_1^+, x_2^+) - \rho^2 V(x_1, x_2) \leqslant 2(x_1^+ - x_2^+)^\top P (w_1 - w_2)\,.$$

By the generalized Young's inequality and by considering the decomposition $P = \sqrt{P}^\top \sqrt{P}$, we have

$$2(x_1^+ - x_2^+)^\top P (w_1 - w_2) \leqslant (1-\rho) V(x_1^+, x_2^+) + \frac{1}{1-\rho} (w_1 - w_2)^\top P (w_1 - w_2)\,.$$

Then, by combining the previous inequalities we obtain

$$V(x_1^+, x_2^+) - \rho V(x_1, x_2) \leqslant \frac{1}{\rho(1-\rho)} (w_1 - w_2)^\top P (w_1 - w_2)\,.$$

As $\rho \in (0,1)$ and $P \succ 0$, the function V is a dissipation-form incremental Lyapunov function [215, Definition 7]. Then, the result follows by [215, Theorem 8]. Finally, by using standard arguments (i.e. [216, Theorem 14]) one can conclude the exponential behavior of solutions. □

Equipped with sufficient conditions for contraction of non-smooth dynamics, we conclude the subsection with some pedagogical examples, providing a useful insight into the applicability of the result.

Example 3.1. *Consider a system of the form*

$$x^+ = \varphi(x) = \operatorname{sat}_r(Ax)\,, \tag{3.13}$$

where $r \in \mathbb{R}^n_{>0}$ and the vector saturation function $\mathrm{sat}(\cdot)$ has components $\mathrm{sat}_i(\cdot) := \max(\min(\cdot, r_i), -r_i)$. It can be easily verified that, for all $x \in \mathbb{R}^n$, the generalized Jacobian of φ [50] satisfies

$$\frac{\partial \varphi}{\partial x}(x) \subset \boldsymbol{co}\{\Delta A, \quad \Delta \in \boldsymbol{\Delta}\}$$

where $\boldsymbol{\Delta} := \{\Delta = \mathrm{diag}(\delta_1, \dots, \delta_n) : \delta_i \in \{0, 1\}, \forall i = 1, \dots, n\}$ is a finite set of matrices representing the vertices of a polytope. More generally, let $\mathcal{V}$ be a set of matrices $\mathcal{V} := \{A_1, \dots, A_v\}$ with $v \in \mathbb{N}$, such that $\partial\varphi(x) \in \boldsymbol{co}\{\mathcal{V}\}$ for all $x \in \mathbb{R}^n$. Then, it suffices to verify (3.11) *on $\mathcal{V}$ and convexity of the equivalent formulation*

$$\begin{pmatrix} \rho^2 P & J^\top \\ J & P^{-1} \end{pmatrix} \succeq 0$$

(obtained via a Schur complement) ensures that (3.11) *holds for $\mathcal{D}\varphi = \boldsymbol{co}\,\mathcal{V}$. Similar reasonings can be followed for smooth monotone saturation-like functions, such as arctangents or hyperbolic tangents.*

Example 3.2. *The study of incremental stability properties of neural networks is gaining attention in the research community, e.g. [31, 55, 241]. The presented contraction analysis tools can be valuable to derive such properties. For example, a multilayer perceptron with $L \in \mathbb{N}$ layers and ReLU activation functions can be described by the following dynamics*

$$\begin{cases} x^+ = y_L \\ y_\ell = W_\ell\, \nu(y_{\ell-1}) + b_\ell, \quad \ell = 1, \dots, L \\ y_0 = x \end{cases} \tag{3.14}$$

with the ReLU function $\nu(\cdot)$ applied component-wise, i.e., $\nu(x)$ has components $\nu_i(x_i) := \max(0, x_i)$, $y_\ell, b_\ell \in \mathbb{R}^{n_\ell}$ and $W_\ell \in \mathbb{R}^{n_\ell \times n_{\ell-1}}$. Denoting by $x \mapsto \varphi(x)$ the function satisfying $y_L = \varphi(x)$ recursively defined in (3.14)*, by the chain rule [50, Theorem 2.6.6], for all $x \in \mathbb{R}^n$, we have*

$$\partial\varphi(x) \subset \boldsymbol{co}\{W_L \Delta_{L-1} W_{L-1} \dots \Delta_1 W_1, \Delta_i \in \boldsymbol{\Delta}\, \forall i = 1, \dots, L-1\},$$

with $\boldsymbol{\Delta}$ defined as in Example 3.1. Proceeding as in Example 3.1, we can conclude exponential δISS properties of (3.14) *by checking* (3.16) *on a set $\mathcal{V}$ satisfying $\partial\varphi(x) \in \boldsymbol{co}\,\mathcal{V}$ for all $x \in \mathbb{R}^n$. Similar results extend to more complex recurrent neural networks, as shown in [55].*

3.1.2 Nonlinear robust feedback design

We now exploit the results of Proposition 3.2 to design a feedback stabilizers $u = \alpha(x, \mathfrak{t})$ inducing exponential δISS of the closed-loop (as defined in Definition 3.3). Consider a nonlinear system of the form

$$x^+ = f(x, \mathfrak{t}) + Bu + w, \tag{3.15}$$

where $f : \mathbb{R}^{n_x} \times \mathbb{N} \to \mathbb{R}^{n_x}$ satisfies Property 3.1 and $B \in \mathbb{R}^{n_x \times n_u}$ is full column rank. The obtained result shows gain margin properties as defined below, which is inspired by [191, Definition 3.13].

Definition 3.4 (Gain margin of radius r)**.** *Consider system* (3.15)*. A function $\alpha : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_u}$ is a δISS feedback with gain margin of radius $r > 0$ if, for any real numbers $\kappa \in [1 - r, 1 + r]$, system* (3.15) *with $u = \kappa\alpha(x, t)$ is δISS with respect to w.*

We now present a result for the design of incrementally stabilizing feedback controllers with gain margin in the non-smooth nonlinear framework described by Property 3.1.

Theorem 3.2. *Let $R \in \mathbb{S}^{n_u}_{\succeq 0}$ and assume that f in* (3.15) *satisfies Property 3.1 for some $\mathcal{D}f \subset \mathbb{R}^{n_x \times n_x}$. Moreover, suppose that there exists $P \in \mathbb{S}^{n_x}_{\succ 0}$ satisfying*

$$J^\top Q J \preceq \rho^2 P, \quad \forall J \in \mathcal{D}f, \tag{3.16a}$$

$$Q := P - \sigma P B \left(R + B^\top P B\right)^{-1} B^\top P, \tag{3.16b}$$

for some $\rho \in (0,1)$ and $\sigma \in (0,1]$. Then for the system (3.15) *the function*

$$u = \alpha(x, \mathfrak{t}) = -\kappa \left(R + B^\top P B\right)^{-1} B^\top P f(x, \mathfrak{t}) \tag{3.17}$$

is a δISS feedback with gain margin of radius $r = \sqrt{1-\sigma}$.

Proof. For the sake of compactness, let us start by defining

$$Y = Y^\top := \left(R + B^\top P B\right)^{-1}, \qquad \Omega = \mathrm{I}_{n_x} - \kappa B Y B^\top P.$$

Since B is assumed to be full column rank, the matrix $R + B^\top P B$ is invertible, and Y exists. Then, Proposition 3.2 states that the closed-loop (3.15), (3.17), which can be written as (3.4) with

$$\varphi(x, \mathfrak{t}) = f(x, \mathfrak{t}) + B\alpha(x, \mathfrak{t}) = \Omega f(x, \mathfrak{t}), \tag{3.18}$$

is exponentially δISS if

$$J^\top \Omega^\top P \Omega J \preceq \rho^2 P, \qquad \forall J \in \mathcal{D}f. \tag{3.19}$$

By expanding the left-hand side in (3.19) and by adding and subtracting $\sigma J^\top P B Y B^\top P J$, due to (3.19) we obtain the equality

$$J^\top \Omega^\top P \Omega J = J^\top P J - \sigma J^\top P B Y B^\top P J + (\sigma - 2\kappa) J^\top P B Y B^\top P J + \kappa^2 J^\top P B Y B^\top P B Y B^\top P J,$$

where we note that the first two terms at the right-hand side coincide with $J^\top Q J$. Then inequality (3.16) implies, for all $J \in \mathcal{D}f$,

$$\begin{aligned} J^\top \Omega^\top P \Omega J &\preceq \rho^2 P + J^\top P B Y \left((\sigma - 2\kappa) Y^{-1} + \kappa^2 B^\top P B\right) Y B^\top P J \\ &\preceq \rho^2 P + (\kappa^2 - 2\kappa + \sigma) J^\top P B Y B^\top P J, \end{aligned}$$

where we expanded Y^{-1} and added $\kappa^2 R \succeq 0$ inside the brackets. Since Y is positive definite, (3.19) holds and Proposition 3.2 applies if $\kappa^2 - 2\kappa + \sigma \leqslant 0$, which holds if and only if

$$1 - \sqrt{1-\sigma} \leqslant \kappa \leqslant 1 + \sqrt{1-\sigma}\,,$$

as to be proven. □

Remark 3.1. *Consider the role of the parameter σ in* (3.16b). *On the one hand, a strictly positive σ implies that the system can be made exponentially δISS with the addition of an input acting in the correct directions. On the other hand, $\sigma = 0$ implies that the autonomous system $x^+ = f(x, \mathfrak{t})$ is already contracting, while a negative σ would mean that the autonomous system is sufficiently robust to withstand inputs in the wrong directions. Then, conditions* (3.16b) *can be seen as a generalization of the discrete-time Modified Algebraic Riccati Inequality (MARI) to the nonlinear framework [43, 221].*

Remark 3.2. *Note that* (3.16b) *implies* $Q \in \mathbb{S}^{n_x}_{\succ 0}$ *if* $R \in \mathbb{S}^{n_u}_{\succ 0}$. *Indeed, by* (3.16b) *and the block matrix inversion identity*

$$(\mathcal{A}+\mathcal{B}\mathcal{D}^{-1}\mathcal{C})^{-1} = \mathcal{A}^{-1} - \mathcal{A}^{-1}\mathcal{B}(\mathcal{D}+\mathcal{C}\mathcal{A}^{-1}\mathcal{B})^{-1}\mathcal{C}\mathcal{A}^{-1}$$

with $\mathcal{A} = P$, $\mathcal{B} = B$, $\mathcal{C} = B^\top$ *and* $\mathcal{D} = R$, *the invertibility of* R *yields the equivalent formulation*

$$Q = (1-\sigma)P + \sigma(P^{-1} + BR^{-1}B^\top)^{-1}.$$

Since $\sigma \in (0,1]$ *and* $P \succ 0$, *matrix* Q *is a* σ*-governed linear interpolation between positive definite matrices and, thus, it is positive.*

Despite lacking infinite gain margin properties, discrete-time controllers designed using Demidovich conditions can be reformulated to exhibit similar characteristics. Specifically, σ, κ in conditions (3.16),(3.17) can be expressed in terms of a pair of parameters σ_1 and σ_2, which demonstrate behavior akin to infinite gain margin. In what follows, we revisit Theorem 3.2 in terms of σ_1, σ_2.

Proposition 3.3. *Let* $R \in \mathbb{S}^{n_u}_{\succ 0}$ *and assume that* f *in* (3.15) *satisfies Property 3.1 for some* $\mathcal{D}f \subset \mathbb{R}^{n_x \times n_x}$. *Moreover, suppose that there exists* $P \in \mathbb{S}^{n_x}_{\succ 0}$ *satisfying*

$$\begin{aligned} J^\top Q J \preceq \rho^2 P, \quad \forall J \in \mathcal{D}f, \\ Q := P - \sigma_1 \tfrac{2\sigma_2-\sigma_1}{\sigma_2^2} PB\left(R + B^\top PB\right)^{-1} B^\top P, \end{aligned} \tag{3.20}$$

for some $\rho \in (0,1)$ *and some scalars* σ_1, σ_2 *satisfying* $\sigma_1(2\sigma_2 - \sigma_1) > 0$. *Then, system* (3.15) *in closed-loop with* $u = \alpha(x, \mathrm{t})$ *selected as* (3.17) *with*

$$\kappa = \frac{2\sigma_2 - \sigma_1}{\sigma_2} \tag{3.21}$$

is exponentially δ*ISS with respect to* w.

Proof. From Theorem 3.2, sufficient conditions for exponential δISS properties require the satisfaction of the bounds

$$\sigma_1 \frac{2\sigma_2 - \sigma_1}{\sigma_2^2} \in (0,1], \tag{3.22a}$$

$$\frac{2\sigma_2 - \sigma_1}{\sigma_2} \in \left[1 - \sqrt{1 - \sigma_1 \frac{2\sigma_2-\sigma_1}{\sigma_2^2}}, 1 + \sqrt{1 - \sigma_1 \frac{2\sigma_2-\sigma_1}{\sigma_2^2}}\right]. \tag{3.22b}$$

We start by considering the first constraint (3.22a). Trivially, positivity is verified for all (σ_1, σ_2) belonging to the set $\mathcal{S}_1 := \left\{(\sigma_1,\sigma_2) \in \mathbb{R}^2 : \sigma_1 > 0 \wedge \sigma_2 > \frac{\sigma_1}{2} \vee \sigma_1 < 0 \wedge \sigma_2 < \frac{\sigma_1}{2}\right\}$, which can be equivalently represented via the cone

$$\mathcal{S}_1 = \left\{(\sigma_1,\sigma_2) \in \mathbb{R}^2 : \sigma_1(2\sigma_2 - \sigma_1) > 0\right\}. \tag{3.23}$$

Similarly, it is easy to show that the upper bound in (3.22a) is satisfied if and only if

$$\sigma_2^2 - 2\sigma_2\sigma_1 + \sigma_1^2 \geqslant 0 \iff (\sigma_2 - \sigma_1)^2 \geqslant 0,$$

which holds for any σ_1, σ_2. Then (3.22a) is satisfied for all $(\sigma_1, \sigma_2) \in \mathcal{S}_1$. Consider now

(3.22b). First, notice that for any σ_1, σ_2 there exist real solutions to the square roots, since

$$1 - \sigma_1 \frac{2\sigma_2 - \sigma_1}{\sigma_2^2} = \frac{(\sigma_2 - \sigma_1)^2}{\sigma_2^2} \geqslant 0.$$

Then, we focus our attention to the satisfaction of the constraints. First, note that the lower bound in (3.22b) reads as

$$\frac{2\sigma_2 - \sigma_1}{\sigma_2} \geqslant 1 - \sqrt{\frac{(\sigma_2 - \sigma_1)^2}{\sigma_2^2}}.$$

Hence, by simple computations, it can be proven to be equivalent to

$$\sqrt{\frac{(\sigma_2 - \sigma_1)^2}{\sigma_2^2}} \geqslant -\frac{\sigma_2 - \sigma_1}{\sigma_2}.$$

On the one hand, if the pair (σ_1, σ_2) belongs to the set

$$\mathcal{S}_2 := \left\{(\sigma_1, \sigma_2) \in \mathbb{R}^2 : \sigma_2 - \sigma_1 \geqslant 0 \wedge \sigma_2 > 0 \vee \sigma_1 - \sigma_2 \leqslant 0 \wedge \sigma_2 < 0\right\} \tag{3.24}$$

the above inequality is satisfied, as the right-hand side becomes negative. On the other hand, if the pair $(\sigma_1, \sigma_2) \notin \mathcal{S}_2$, the right-hand side is positive and squaring of the terms yields $(\sigma_2 - \sigma_1)^2 \geqslant (\sigma_2 - \sigma_1)^2$, which is always verified. Hence, (3.22b) is satisfied for all $(\sigma_1, \sigma_2) \in \mathbb{R}^2$ and (3.22) holds for all $(\sigma_1, \sigma_2) \in \mathcal{S}_1$. Therefore, by Theorem 3.2, the closed-loop (3.15)-(3.17) with (3.21) is exponentially δISS. □

Remark 3.3. *The resemblance to infinite gain margin laws for continuous-time as in [84] is more evident by defining the matrix* $\mathbf{R} := \frac{\sigma_2^2}{2\sigma_2 - \sigma_1}(R + B^\top P B) \succ 0$. *In the continuous-time case, the contraction condition for smooth dynamics reads*

$$\frac{\partial f}{\partial x}(x,t)^\top P + P\frac{\partial f}{\partial x}(x,t) - \mu P B R^{-1} B^\top P \preceq -\lambda P,$$

where λ *and* μ *are positive scalars. Then, the control law* $u = -\kappa\beta(x,t)$ *with*

$$\beta(x,t) = R^{-1} B^\top P x$$

makes the closed-loop system δ*ISS for any* $k \geqslant \frac{\rho}{2}$ *[83]. Hence, by substituting* $\mathbf{R}$*, conditions* (3.16) *under* (3.20)*,*(3.21) *rewrite as*

$$J^\top (P - \sigma_1 P B \mathbf{R}^{-1} B^\top P) J \preceq \rho P$$

and

$$\alpha(x, \mathfrak{t}) = -\sigma_2 \mathbf{R}^{-1} B^\top P f(x).$$

This closely resembles the continuous-time scenario as it allows for $\sigma_1 > 0, \sigma_2 > \frac{\sigma_1}{2}$*. However, the weight matrix* $\mathbf{R}$ *is scaled by the parameters* σ_1, σ_2 *and, due to the quadratic form of* (3.16a) *and the coupling between* σ_1 *and* σ_2*, the symmetric solution* $\sigma_1 < 0, \sigma_2 < \frac{\sigma_1}{2}$ *is also valid.*

3.1.3 GEVPs for exponential δISS

We now discuss numerically efficient formulations of the results of Theorem 3.2. LMI-based conditions for robust stabilization are a valuable tool for control design for discrete-time nonlinear

systems, see e.g. [88, 110, 194, 247]. Hence, inspired by these works and recent LMI approaches for solving MARI inequalities [199, 221], we propose LMI-based conditions to obtain the solution of the MARI-like inequality (3.16). This numerically efficient formulation provides a viable solution to the design problem of robustly synchronizing controllers. First, we introduce an equivalent formulation for (3.16). The parameters of the proposed reformulation can be obtained by solving a generalized eigenvalue problem (GEVP). Then, we focus our attention on the case where the set of possible open-loop Jacobians of the system dynamics (3.15) is polytopic. Finally, we target the specific case of Lur'e systems and propose numerically tractable sufficient conditions.

Formulation as a GEVP. We start by reformulating Theorem 3.2 as an LMI problem. This provides convex analysis conditions for constructing matrix P. Given this new formulation, the convergence rate ρ and the parameter σ in (3.16) can be estimated as part of a GEVP.

Proposition 3.4. *Let $R \in \mathbb{S}^{n_u}_{\succ 0}, \sigma \in (0,1], \rho \in (0,1)$. The following LMIs in the decision variables W and Σ*

$$W \succ 0, \quad \Sigma \succ 0, \qquad \begin{pmatrix} \rho W & WJ^\top \\ JW & \rho\Sigma \end{pmatrix} \succeq 0, \tag{3.25a}$$

$$\begin{pmatrix} W + \sigma BR^{-1}B^\top - \Sigma & BR^{-1}B^\top \\ BR^{-1}B^\top & \frac{1}{\sigma}\left(\frac{W}{(1-\sigma)} + BR^{-1}B^\top\right) \end{pmatrix} \succeq 0, \tag{3.25b}$$

hold if and only if conditions (3.16) *hold with $P = W^{-1} \succ 0$. Moreover,* (3.25) *is a generalized eigenvalue problem in (σ, ρ, R), namely, if it is feasible for $(\bar{\sigma}, \bar{\rho}, \bar{R})$, then it is feasible for any (σ, ρ, R) such that $\sigma \geqslant \bar{\sigma}$, $\rho \geqslant \bar{\rho}$, $R \preceq \bar{R}$ and, conversely, if it is infeasible for $(\underline{\sigma}, \underline{\rho}, \underline{R})$, then it is infeasible for any (σ, ρ, R) such that $\sigma \leqslant \underline{\sigma}$, $\rho \leqslant \underline{\rho}$, $R \succeq \underline{R}$.*

Proof. Consider the last LMI of (3.25). Since $W \succ 0$ and $R^{-1} \succ 0$, then its $(2,2)$ entry is positive definite. Then, by the Schur complement (3.25b) holds if and only if

$$W + \sigma BR^{-1}B^\top - \Sigma \succeq \sigma(1-\sigma)BR^{-1}B^\top(W + (1-\sigma)BR^{-1}B^\top)^{-1}BR^{-1}B^\top,$$

which can be rearranged as

$$\Sigma \preceq W + \sigma B\Big(R^{-1} - (1-\sigma)R^{-1}B^\top(W + (1-\sigma)BR^{-1}B^\top)^{-1}BR^{-1}\Big)B^\top. \tag{3.26}$$

By the matrix inversion lemma (3.78) with $\mathcal{A} = R$, $\mathcal{B} = \sqrt{1-\sigma}B^\top$, $\mathcal{C} = \sqrt{1-\sigma}B$, $\mathcal{D} = W$, inequality (3.26) is equivalent to

$$\Sigma \preceq W + \sigma B(R + (1-\sigma)B^\top W^{-1}B)^{-1}B^\top.$$

Left and right multiplication of both sides by $W^{-1} \succ 0$ yields the equivalent condition

$$W^{-1}\Sigma W^{-1} \preceq W^{-1} - \sigma W^{-1}B(-(R + B^\top W^{-1}B) + \sigma B^\top W^{-1}B)^{-1}B^\top W^{-1}. \tag{3.27}$$

Once again, by the matrix inversion lemma (3.78) applied with $\mathcal{A} = W$, $\mathcal{B} = \sqrt{\sigma}B$, $\mathcal{C} = \sqrt{\sigma}B^\top$, $\mathcal{D} = -(R + B^\top W^{-1}B)$, inequality (3.27) is equivalent to

$$W^{-1}\Sigma W^{-1} \preceq (W - \sigma B(R + B^\top W^{-1}B)^{-1}B^\top)^{-1}.$$

By left and right multiplying both sides by W, we obtain the equivalent inequality

$$\Sigma \preceq W(W - \sigma B(R + B^\top W^{-1} B)^{-1} B^\top)^{-1} W. \tag{3.28}$$

Since $(\mathcal{A}\mathcal{B}\mathcal{C})^{-1} = \mathcal{C}^{-1}\mathcal{B}^{-1}\mathcal{A}^{-1}$ for any invertible matrices $\mathcal{A}, \mathcal{B}, \mathcal{C}$, inequality (3.28) is equivalent to

$$\begin{aligned} \Sigma &\preceq (W^{-1}(W - \sigma B(R + B^\top W^{-1} B)^{-1} B^\top) W^{-1})^{-1} \\ &\preceq (P - \sigma PB(R + B^\top PB)^{-1} B^\top P)^{-1} = Q^{-1}, \end{aligned} \tag{3.29}$$

where we used $P = W^{-1}$ and the definition of Q in (3.16b). Consider now the right LMI in (3.25a). By left and right multiplication by the matrix

$$T = \begin{pmatrix} W^{-1} & 0 \\ 0 & \mathrm{I}_{n_x} \end{pmatrix} = \begin{pmatrix} P & 0 \\ 0 & \mathrm{I}_{n_x} \end{pmatrix},$$

we have

$$\begin{pmatrix} \rho P & J^\top \\ J & \rho \Sigma \end{pmatrix} = \begin{pmatrix} \rho W^{-1} & J^\top \\ J & \rho \Sigma \end{pmatrix} \succeq 0.$$

Then, by the Schur complement, we obtain the equivalent condition $\rho W^{-1} - \rho^{-1} J^\top \Sigma^{-1} J \succeq 0$, which, with the selection $P = W^{-1}$, can be written as

$$J^\top \Sigma^{-1} J \preceq \rho^2 P. \tag{3.30}$$

Summarizing, we proved the equivalence of (3.25) with the four inequalities $P \succ 0$, $\Sigma \succ 0$, (3.29) and (3.30), where we emphasize that under (3.25) Q^{-1} exists due to the positive definiteness of Q implied by $R \succ 0$, as established in Remark 3.2. More specifically, (3.25) is equivalent to

$$P \succ 0, \quad \Sigma \succ 0, \quad \Sigma^{-1} \succeq Q, \quad J^\top \Sigma^{-1} J \preceq \rho^2 P. \tag{3.31}$$

To complete the first part of the proof, we show that (3.31) is equivalent to (3.16). If (3.16) holds, then $Q \succ 0$ and (3.31) holds with $\Sigma = Q^{-1}$. If (3.31) holds, then

$$J^\top Q J \preceq J^\top \Sigma^{-1} J \preceq \rho^2 P,$$

thus completing the first part of the proof. To prove that (3.25) is a GEVP in (σ, ρ, R), let us denote by $\bar{W}, \bar{\Sigma}$ the solution of (3.25) with $(\bar{\sigma}, \bar{\rho}, \bar{R})$. If $\sigma \geqslant \bar{\sigma} > 0$, $0 \prec R \preceq \bar{R}$, we obtain

$$\bar{Q} = \bar{P} - \bar{\sigma} \bar{P} B(\bar{R} + B^\top \bar{P} B)^{-1} B^\top \bar{P} \succeq \bar{P} - \sigma \bar{P} B(R + B^\top \bar{P} B)^{-1} B^\top \bar{P} = Q,$$

where $\bar{P} = \bar{W}^{-1} \succ 0$. Then, by (3.31) and since $\bar{\Sigma}^{-1} \succeq \bar{Q}$ and $\rho \geqslant \bar{\rho}$, the following inequalities hold

$$\bar{P} \succ 0, \quad \bar{\Sigma} \succ 0, \quad \bar{\Sigma}^{-1} \succeq Q, \quad J^\top \bar{\Sigma}^{-1} J \preceq \bar{\rho}^2 \bar{P}.$$

Due to the equivalence between inequalities (3.31) and (3.25), we conclude that $P = \bar{P}$ is solution to (3.25) with (σ, ρ, R). Similar reasonings prove infeasibility of (3.31) for any (σ, ρ, R) such that $\sigma \leqslant \underline{\sigma}$, $\rho \leqslant \underline{\rho}$, $R \succeq \underline{R}$ if (3.31) is infeasible for (σ, ρ, R). □

Polytopic conditions. Combined with Proposition 3.2, Proposition 3.4 requires the satisfaction of (3.25) for all $J \in \mathcal{D}f$. This may turn out to be impracticable, as $\mathcal{D}f$ could be infinite

dimensional. However, under some additional assumptions on system (3.15), we can follow a polytopic approach similar to the one in Examples 3.1 and 3.2. Hence, we propose the following result addressing the case where the open-loop system Jacobian belongs to a polytopic set defined by a finite number of vertices.

Corollary 3.1. *Let $R \in \mathbb{S}^{n_u}_{\succ 0}$ and assume that f in* (3.15) *satisfies Property 3.1 for some $\mathcal{D}f \subset \mathbb{R}^{n_x \times n_x}$. Moreover, suppose there exists a finite set of matrices $\mathcal{V} := \{A_1, \dots, A_v\} \subset \mathbb{R}^{n_x \times n_x}$ such that $\mathcal{D}f \subseteq$ **co** $\mathcal{V}$. If there exist matrices $W, \Sigma \in \mathbb{S}^{n_x}_{\succ 0}$ and scalars $\rho \in (0,1)$, $\sigma \in (0,1]$ satisfying* (3.25) *for all $J \in \mathcal{V}$, the control law $u = \alpha(x, \mathfrak{t})$ with α defined in* (3.17) *and $P = W^{-1}$ makes the closed-loop exponentially δISS with respect to w with gain margin of radius $r = \sqrt{1-\sigma}$.*

Lur'e systems. We further specialize our result to the case of Lur'e systems. Namely, we now consider nonlinear discrete-time systems of the form

$$x^+ = f(x) + Bu = Ax + F\phi(Cx) + Bu, \tag{3.32}$$

where $C \in \mathbb{R}^{n_y \times n_x}$, $F \in \mathbb{R}^{n_x \times n_y}$ and the square nonlinearity $\phi : \mathbb{R}^{n_y} \to \mathbb{R}^{n_y}$ is a pool of n_y, possibly different, feedback nonlinear elements $\phi(y) := \operatorname{diag}(\phi_1(y_1), \dots, \phi_{n_y}(y_{n_y}))$ whose components $\phi_i, i = 1, \dots, n_y$, satisfy Property 3.1 for some intervals $\mathcal{D}\phi_i \subset \mathbb{R}$, $i = 1, \dots, n_y$. We assume that each function ψ_i belongs to an incremental sector $[0, \omega_i]$, with $\omega_i \geqslant 0$, in the following classical sense:

$$(\phi_i(s_1) - \phi_i(s_2))(\phi_i(s_1) - \phi_i(s_2) - \omega_i(s_1 - s_2)) \leqslant 0, \ \forall s_1, s_2 \in \mathbb{R}. \tag{3.33}$$

By the non-smooth mean value theorem [50, Theorem 2.3.7], we may combine bounds (3.33) into

$$\operatorname{He}\{J_\phi S(J_\phi - \Omega)\} \preceq 0 \tag{3.34}$$

which holds for all diagonal $J_\phi \in \mathcal{D}\phi = \operatorname{diag}(\mathcal{D}\phi_1, \dots, \mathcal{D}\phi_{n_y})$, for any diagonal $S \in \mathbb{S}^{n_y}_{\succeq 0}$ and for some diagonal $\Omega = \operatorname{diag}(\omega_1, \dots, \omega_{n_y}) \in \mathbb{S}^{n_y}_{\succeq 0}$. We then have the following result.

Proposition 3.5. *Let $R \in \mathbb{S}^{n_u}_{\succ 0}$ and suppose that ϕ in* (3.32) *satisfies* (3.33) *for some $\Omega = \operatorname{diag}(\omega_1, \dots, \omega_{n_y})$. If there exist symmetric matrices W, Σ, a diagonal matrix $S \in \mathbb{S}^{n_y}_{\succeq 0}$ and scalars $\rho \in (0,1)$, $\sigma \in (0,1]$ satisfying* (3.25b) *and*

$$W \succeq 0, \quad \Sigma \succeq 0, \quad \begin{pmatrix} \rho W & WA^\top & WC^\top \\ AW & \rho\Sigma & -F\Omega^\top S \\ CW & -S\Omega F^\top & 2S \end{pmatrix} \succeq 0, \tag{3.35}$$

then the control law $u = \alpha(x, \mathfrak{t})$ with α defined in (3.17) *and $P = W^{-1}$ makes the closed-loop* (3.32)-(3.17) *exponentially δISS with respect to w with gain margin of radius $r = \sqrt{1-\sigma}$.*

Proof. Due to [50, Theorem 2.3.7], (3.34) holds for any diagonal $S \in \mathbb{S}^{n_y}_{\succeq 0}$ and any $J_\phi \in \mathcal{D}\phi$. Define the matrix $\Lambda^\top := \begin{pmatrix} \mathrm{I}_{n_x} & 0 & 0 \\ 0 & \mathrm{I}_{n_x} & FJ_\phi \end{pmatrix}$ with any diagonal $J_\phi \in \mathcal{D}\phi$. It is easy to verify that (3.34) implies

$$\Lambda^\top \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -F\Omega^\top S \\ 0 & -S\Omega F^\top & 2S \end{pmatrix} \Lambda = \Lambda^\top \Pi \Lambda \preceq 0. \tag{3.36}$$

Consider now (3.35), which implies

$$\Lambda^\top \begin{pmatrix} \rho W & WA^\top & WC^\top \\ AW & \rho\Sigma & -F\Omega^\top S \\ CW & -S\Omega F^\top & 2S \end{pmatrix} \Lambda = \Lambda^\top (\Xi + \Pi)\, \Lambda \succeq 0,$$

where we defined

$$\Xi := \begin{pmatrix} \rho W & WA^\top & WC^\top \\ AW & \rho\Sigma & 0 \\ CW & 0 & 0 \end{pmatrix}. \tag{3.37}$$

By (3.36), we then have

$$\Lambda^\top \Xi\, \Lambda \succeq \Lambda^\top (\Xi + \Pi)\, \Lambda \succeq 0,$$

thus showing $\Lambda^\top \Xi\, \Lambda \succeq 0$. Then, the expansion of the product leads to

$$\Lambda^\top \Xi\, \Lambda = \begin{pmatrix} \rho W & W(A + FJ_\phi C)^\top \\ (A + FJ_\phi C)W & \rho\Sigma \end{pmatrix} \succeq 0, \forall J_\phi \in \mathcal{D}\phi.$$

By Proposition 3.4, the assumption (3.25b) implies that conditions (3.16) hold with $P = W^{-1} \succ 0$ for any $J \in \mathcal{D}f = A + F\mathcal{D}\phi C$. The proof is concluded by Proposition 3.2. □

3.1.4 Optimality of discrete-time contractive feedbacks

In Section 3.1.2, we propose a fundamental result to derive method for designing robust incrementally stabilizing controllers through optimization problems. In order to provide further insight into the properties of these controllers and their potential for application via machine learning tools, we aim to demonstrate their optimality, similar to the continuous-time case [84]. To achieve this goal, we introduce a generalized version of Theorem 3.2, which provides sufficient conditions for δISS based on Demidovich-like conditions with a discount factor. This generalization lays the foundation for exploring the possibility of learning these robust controllers using data-driven methods such as DRL. These results can be linked to the findings in [41].

Proposition 3.6. *Let $R \in \mathbb{S}^{n_u}_{\succ 0}$ and assume that f in* (3.15) *satisfies Property 3.1 for some $\mathcal{D}f \subset \mathbb{R}^{n_x \times n_x}$. Moreover, suppose that there exists $P \in \mathbb{S}^{n_x}_{\succ 0}$ satisfying*

$$\begin{aligned} J^\top Q J &\preceq \rho^2 P, \quad \forall J \in \mathcal{D}f, \\ Q &:= \eta P - \sigma\eta^2 PB \left(R + \eta B^\top P B\right)^{-1} B^\top P, \end{aligned} \tag{3.38}$$

for some $\sigma, \eta \in (0, 1]$ and $\rho \in (0, \eta)$. Then, system (3.15) *in closed-loop with $u = \alpha(x, \mathfrak{t})$ selected as*

$$\alpha(x, \mathfrak{t}) = -\kappa \left(R + \eta B^\top P B\right)^{-1} B^\top P f(x, \mathfrak{t}) \tag{3.39}$$

is exponentially δISS with respect to w for any $\kappa \in \left[\eta(1 - \sqrt{1-\sigma}), \eta(1 + \sqrt{1-\sigma})\right]$.

Proof. Let us define $P_\eta := \eta P$ and $\rho_\eta := \frac{\rho}{\eta}$. By substituting P_η, ρ_η in (3.38), we obtain a form equivalent to (3.16b). As a consequence, since $\rho < \eta$, by Theorem 3.2 the control law

$$\alpha(x, \mathfrak{t}) = -\kappa_\eta \left(R + B^\top P_\eta B\right)^{-1} B^\top P_\eta f(x, \mathfrak{t}) \tag{3.40}$$

makes the closed-loop (3.15)-(3.40) exponentially δISS with respect to w for any scalar gain $\kappa_\eta \in \left[1-\sqrt{1-\sigma}, 1+\sqrt{1-\sigma}\right]$. By the definition of P_η, function (3.40) is equivalent to

$$\alpha(x, \mathrm{t}) = -\eta\kappa_\eta \left(R + \eta B^\top P B\right)^{-1} B^\top P f(x, \mathrm{t}).$$

By defining $\kappa := \eta\kappa_\eta$, the bounds on κ_η translate to $\kappa \in \left[\eta(1-\sqrt{1-\sigma}), \eta(1+\sqrt{1-\sigma})\right]$, thus concluding the proof. □

Remark 3.4. *It is interesting to notice that the addition of a discount factor $\eta \in (0, 1]$ can drastically affect the gain margin properties of the controller. Indeed, η scales the radius of the circle of allowed gains and shifts its center towards the origin.*

In order to demonstrate the optimality of the proposed control law, we will draw on concepts from the dynamic programming literature, see Appendix A.1. Hence, throughout the remainder of this section, we will refer to control policies rather than control laws. Furthermore, as we are considering cooperative tasks, we can focus on deterministic policies that directly map states to actions, without loss of generality. For simplicity, we will now restrict our analysis to smooth, time-invariant system dynamics of the form (3.15), where f is a C^1 function of the state only. However, the results can be generalized to systems satisfying Property 3.1. As the objective is to prove the optimality of a controller that achieves closed-loop incremental stability, the objective function will involve a distance metric between two arbitrary trajectories of the system, rather than a measure of the distance to the origin. In other words, the problem can be reformulated as a multi-agent synchronization one between homogeneous agents of the form

$$\begin{cases} x_1^+ = f(x_1) + Bu_1 \\ x_2^+ = f(x_2) + Bu_2 \end{cases} \tag{3.41}$$

where $u_1, u_2 \in \mathbb{R}^{n_u}$ are arbitrary control actions. Then, we aim at showing that $u_1 = \alpha(x_1)$, $u_2 = \alpha(x_2)$ with α as in (3.40) is a pair of optimal actions with respect to some synchronizing cost, under a proper selection of the gain κ. Let us define the error coordinates

$$\begin{aligned} \tilde{x}(\mathrm{t}) &:= x_1(\mathrm{t}) - x_2(\mathrm{t}) \\ \tilde{f}(\mathrm{t}) &:= f(x_1(\mathrm{t})) - f(x_2(\mathrm{t})) \\ \tilde{u}(\mathrm{t}) &= u_1(\mathrm{t}) - u_2(\mathrm{t}). \end{aligned}$$

We now present the optimality result for discrete-time contraction-based control laws. Consider a discrete-time nonlinear system described by the following difference equation

$$x^+ = f(x) + Bu, \tag{3.42}$$

with $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$ and $f \in C^1$.

Theorem 3.3. *Let $R \in \mathbb{S}^{n_u}_{\succ 0}$, and assume there exists $P \in \mathbb{S}^{n_x}_{\succ 0}$ solution of*

$$\begin{aligned} \frac{\partial f}{\partial x}(x)^\top Q \frac{\partial f}{\partial x}(x) \prec P, \qquad \forall x \in \mathbb{R}^{n_x} \\ Q := \eta P - \sigma\eta^2 P B \left(R + \eta B^\top P B\right)^{-1} B^\top P, \end{aligned} \tag{3.43}$$

for some $\sigma, \eta \in (0, 1]$. Then

$$\alpha(x) = -\eta\sigma \left(R + \eta B^\top P B\right)^{-1} B^\top P f(x) \tag{3.44}$$

makes the closed-loop (3.42)-(3.44) *exponentially δISS. Moreover, for any pair on initial conditions* $x_1(\mathfrak{t}), x_2(\mathfrak{t})$ *evolving according to* (3.41)*, the control policy* (3.44) *is an optimal policy with respect to the incremental minimization problem described by the cost function*

$$J(x_1(\mathfrak{t}), x_2(\mathfrak{t}), \tilde{u}(\mathfrak{t})) = \sum_{k=0}^{\infty} \eta^k \left(\tilde{x}(k+\mathfrak{t})^\top S(\bar{x}(k+\mathfrak{t}))\tilde{x}(k+\mathfrak{t}) + \tilde{u}(k+\mathfrak{t})\mathbf{R}\tilde{u}(k+\mathfrak{t}) \right), \tag{3.45}$$

where we defined

$$S(x) := P - \frac{\partial f}{\partial x}(x)^\top Q \frac{\partial f}{\partial x}(x), \qquad \mathbf{R} := R + \frac{1-\sigma}{\sigma}(R + \eta B^\top P B) \tag{3.46}$$

and $\bar{x}(k+\mathfrak{t}) := \bar{s}x_1(k+\mathfrak{t}) + (1-\bar{s})x_2(k+\mathfrak{t})$ *with* $\bar{s} \in [0,1]$ *such that*

$$f(x_1) - f(x_2) = \frac{\partial f}{\partial x}\left(\bar{s}x_1 + (1-\bar{s})x_2\right)(x_1 - x_2).$$

Proof. Since $1 - \sqrt{1-\sigma} \leqslant \sigma \leqslant 1 + \sqrt{1-\sigma}$ for all $\sigma \in (0,1]$, exponential δISS properties are easily proven via Proposition 3.6. Hence, we focus on the optimality of policy (3.44). Note that, since $R \in \mathbb{S}^{n_u}_{\succ 0}$ and $\eta \in (0,1]$, the matrix Q in (3.38) can be proven to be positive definite via steps similar to the ones in Remark 3.2, thus showing that the matrix valued function $S : \mathbb{R}^{n_x} \to \mathbb{S}^{n_x}_{\succ 0}$ satisfies

$$P \succeq S(x) \succ 0, \qquad \forall x \in \mathbb{R}^{n_x}.$$

Consider the Lyapunov function

$$V(x_1, x_2) := (x_1 - x_2)^\top P (x_1 - x_2) = \tilde{x}^\top P \tilde{x}.$$

We now show that it is the state-value function for problem (3.45) under policy (3.44), namely $V(x_1, x_2) = \mathbf{J}_{\alpha,\alpha}(x_1, x_2)$, being $\mathbf{J}_{\alpha,\alpha}$ the state-value function obtained under the choice $u_1 = \alpha(x_1)$ and $u_2 = \alpha(x_2)$ for all $(x_1, x_2) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_x}$. To do so, let $z = x_2$ and $x_1 = z + \tilde{x}$. At the successive timestep, we have

$$V(x_1^+, x_2^+) = (\tilde{x}^+)^\top P \tilde{x}^+ = \left(\tilde{f} + B\left(\alpha(z+\tilde{x}) - \alpha(z)\right)\right)^\top P \left(\tilde{f} + B\left(\alpha(z+\tilde{x}) - \alpha(z)\right)\right).$$

By defining the change of coordinates

$$\tilde{\alpha} := \alpha(x_1) - \alpha(x_2) = \alpha(z+\tilde{x}) - \alpha(z) = -\eta\sigma(R + B^\top P B)^{-1} B^\top P \tilde{f}, \tag{3.47}$$

definition (3.46) and the fact that $-(R + B^\top P B)\tilde{\alpha} = \eta\sigma B^\top P \tilde{f}$ yield

$$\begin{aligned} V(x_1^+, x_2^+) &= \tilde{f}^\top P \tilde{f} + \tilde{\alpha}^\top B^\top P \tilde{f} + \tilde{f}^\top P B \tilde{\alpha} + \tilde{\alpha}^\top B^\top P B \tilde{\alpha} \\ &= \tilde{f}^\top \left(P - \eta\sigma P B (R + \eta B^\top P B)^{-1} B^\top P\right) \tilde{f} - \frac{1}{\eta\sigma}\tilde{\alpha}^\top (R + \eta B^\top P B)\tilde{\alpha} + \tilde{\alpha}^\top B^\top P B \tilde{\alpha} \\ &= \frac{1}{\eta}\tilde{f}^\top Q \tilde{f} + \tilde{\alpha}^\top \left(B^\top P B - \frac{1}{\eta\sigma}(R + \eta B^\top P B)\right)\tilde{\alpha}, \end{aligned}$$

where Q is defined as in (3.38). Then, by adding and subtracting $\eta^{-1}R$ inside the parenthesis of the last term, we obtain the equivalence

$$V(x_1^+, x_2^+) = \frac{1}{\eta}\tilde{f}^\top Q\tilde{f} + \tilde{\alpha}^\top \left(\frac{-1+\sigma}{\eta\sigma}(R + \eta B^\top P B) - \frac{1}{\eta}R\right)\tilde{\alpha} = \frac{1}{\eta}\left(\tilde{f}^\top Q\tilde{f} - \tilde{\alpha}\mathbf{R}\tilde{\alpha}\right). \tag{3.48}$$

Let us define $F : \mathbb{R} \to \mathbb{R}^{n_x}$ as

$$F(s) := f(z + s\tilde{x}) - f(z).$$

It can be easily verified that $F(0) = 0$ and $\tilde{f} = F(1) - F(0)$. Hence, by the mean value theorem there exists a scalar $\bar{s} \in [0, 1]$ such that

$$F(1) - F(0) = \frac{dF}{ds}(\bar{s}) = \frac{\partial f}{\partial x}(z + \bar{s}\tilde{x})\tilde{x}.$$

Consequently, in view of (3.43) and by defining $\bar{x} := z + \bar{s}\tilde{x}$, equation (3.48) is equivalent to

$$\begin{aligned}\eta V(x_1^+, x_2^+) &= V(x_1, x_2) - \tilde{x}^\top S(\bar{x})\tilde{x} - \tilde{x}^\top \frac{\partial f}{\partial x}(\bar{x})^\top Q \frac{\partial f}{\partial x}(\bar{x})\tilde{x} + \tilde{f}^\top Q\tilde{f} - \alpha\mathbf{R}\tilde{\alpha} \\ &= -(\tilde{x}^\top S(\bar{x})\tilde{x} + \tilde{\alpha}^\top \mathbf{R}\tilde{\alpha}) + V(x_1, x_2).\end{aligned} \tag{3.49}$$

Hence, we showed V can be rewritten in the recursive form of a Bellman's equation and that $V = \mathbf{J}_{\alpha,\alpha}$. We now show that it also is the optimal value function under the objective (3.45). Consider two policies $\boldsymbol{\pi}_1, \boldsymbol{\pi}_2$ which differ from α only in the first input. In other words, for $i = 1, 2$, let control policy $\boldsymbol{\pi}_i$ be defined as

$$\pi_i(x_1(k+\mathfrak{t}), x_2(k+\mathfrak{t})) = \begin{cases} \alpha(x_i(k+\mathfrak{t})) - \delta_i(x_1(k+\mathfrak{t}), x_2(k+\mathfrak{t})) & k = 0, \\ \alpha(x_i(k+\mathfrak{t})) & k \geqslant 1. \end{cases} \tag{3.50}$$

Consequently, the state-value function can be written as

$$\mathbf{J}_{\boldsymbol{\pi}_1,\boldsymbol{\pi}_2}(x_1, x_2) = \tilde{x}^\top S(\bar{x})\tilde{x} + \tilde{\boldsymbol{\pi}}^\top \mathbf{R}\tilde{\boldsymbol{\pi}} + \eta \mathbf{J}_{\alpha,\alpha}(x_{1,\boldsymbol{\pi}_1}^+, x_{2,\boldsymbol{\pi}_2}^+), \tag{3.51}$$

where $x_{i,\boldsymbol{\pi}_i}^+$ is the next state obtained using $\boldsymbol{\pi}_i$ with $i = 1, 2$ and $\tilde{\boldsymbol{\pi}} := \boldsymbol{\pi}_1(x_1, x_2) - \boldsymbol{\pi}_2(x_1, x_2)$. Note that (3.49) implies

$$\tilde{x}^\top S(\bar{x})\tilde{x} + \tilde{\alpha}^\top \mathbf{R}\tilde{\alpha} = V(x_1, x_2) - \eta V(x_{1,\alpha}^+, x_{2,\alpha}^+) = \mathbf{J}_{\alpha,\alpha}(x_1, x_2) - \eta \mathbf{J}_{\alpha,\alpha}(x_{1,\alpha}^+, x_{2,\alpha}^+),$$

where $x_{i,\alpha}^+$ with $i = 1, 2$ is the next state obtained following α. Then, recalling (3.51), we obtain

$$\mathbf{J}_{\boldsymbol{\pi}_1,\boldsymbol{\pi}_2}(x_1, x_2) = \mathbf{J}_{\alpha,\alpha}(x_1, x_2) - \eta \mathbf{J}_{\alpha,\alpha}(x_{1,\alpha}^+, x_{2,\alpha}^+) - \tilde{\alpha}^\top \mathbf{R}\tilde{\alpha} + \tilde{\boldsymbol{\pi}}^\top \mathbf{R}\tilde{\boldsymbol{\pi}} + \eta \mathbf{J}_{\alpha,\alpha}(x_{1,\boldsymbol{\pi}_1}^+, x_{2,\boldsymbol{\pi}_2}^+).$$

Expansion of the term $\tilde{\boldsymbol{\pi}}$ paired with (3.50) yields

$$\mathbf{J}_{\boldsymbol{\pi}_1,\boldsymbol{\pi}_2}(x_1, x_2) = \mathbf{J}_{\alpha,\alpha}(x_1, x_2) - 2\tilde{\alpha}^\top \mathbf{R}\tilde{\delta} + \tilde{\delta}^\top \mathbf{R}\tilde{\delta} + \eta \tilde{\mathbf{J}}_{\boldsymbol{\pi},\alpha}(x_1^+, x_2^+), \tag{3.52}$$

where we defined $\tilde{\delta} := \delta_1(x_1, x_2) - \delta_2(x_1, x_2)$ and the state-value function error term $\tilde{\mathbf{J}}_{\boldsymbol{\pi},\alpha}(x_1^+, x_2^+) := \mathbf{J}_{\alpha,\alpha}(x^+_{1,\boldsymbol{\pi}_1}, x^+_{2,\boldsymbol{\pi}_2}) - \mathbf{J}_{\alpha,\alpha}(x^+_{1,\alpha}, x^+_{2,\alpha})$. Hence, using the fact that $\mathbf{J}_{\alpha,\alpha}(x_1, x_2) = V(x_1, x_2) = \tilde{x}^\top P \tilde{x}$ and (3.50), we obtain

$$\begin{aligned}\tilde{\mathbf{J}}_{\boldsymbol{\pi},\alpha}(x_1^+, x_2^+) &= \left(\tilde{f} + B\tilde{\boldsymbol{\pi}}\right)^\top P \left(\tilde{f} + B\tilde{\boldsymbol{\pi}}\right) - \left(\tilde{f} + B\tilde{\alpha}\right)^\top P \left(\tilde{f} + B\tilde{\alpha}\right) \\ &= \tilde{\boldsymbol{\pi}}^\top B^\top P \tilde{f} + \tilde{f}^\top P B \tilde{\boldsymbol{\pi}} + \tilde{\boldsymbol{\pi}}^\top B^\top P B \tilde{\boldsymbol{\pi}} - \tilde{\alpha}^\top B^\top P \tilde{f} - \tilde{f}^\top P B \tilde{\alpha} - \tilde{\alpha}^\top B^\top P B \tilde{\alpha} \\ &= -\tilde{\delta}^\top B^\top P \tilde{f} - \tilde{f}^\top P B \tilde{\delta} - \tilde{\delta}^\top B^\top P B \tilde{\alpha} - \tilde{\alpha}^\top B^\top P B \tilde{\delta} + \tilde{\delta}^\top B^\top P B \tilde{\delta}.\end{aligned}$$

By the definition of $\tilde{\alpha}$ in (3.47) and the fact that $-(R + B^\top P B)\tilde{\alpha} = \eta\sigma B^\top P \tilde{f}$, the above relation is equivalent to

$$\begin{aligned}\tilde{\mathbf{J}}_{\boldsymbol{\pi},\alpha}(x_1^+, x_2^+) &= \frac{2}{\eta\sigma}\tilde{\delta}^\top (R + \eta B^\top P B)\tilde{\alpha} - 2\tilde{\delta}^\top B^\top P B \tilde{\alpha} + \tilde{\delta}^\top B^\top P B \tilde{\delta} \\ &= \frac{2(1-\sigma)}{\eta\sigma}\tilde{\delta}^\top (R + \eta B^\top P B)\tilde{\alpha} + \frac{2}{\eta}\tilde{\delta}^\top R \tilde{\alpha} + \tilde{\delta}^\top B^\top P B \tilde{\delta} \\ &= \frac{2}{\eta}\tilde{\delta}^\top \left(\frac{1-\sigma}{\sigma}(R + B^\top P B) + R\right)\tilde{\alpha} + \tilde{\delta}^\top B^\top P B \tilde{\delta} \\ &= \frac{2}{\eta}\tilde{\delta}^\top \mathbf{R} \tilde{\alpha} + \tilde{\delta}^\top B^\top P B \tilde{\delta}. \end{aligned} \tag{3.53}$$

Hence, by substituting (3.53) in (3.52) we obtain

$$\begin{aligned}\mathbf{J}_{\boldsymbol{\pi}_1,\boldsymbol{\pi}_2}(x_1, x_2) &= \mathbf{J}_{\alpha,\alpha}(x_1, x_2) - 2\tilde{\alpha}^\top R \tilde{\delta} + \tilde{\delta}^\top \mathbf{R} \tilde{\delta} + 2\tilde{\delta}^\top \mathbf{R} \tilde{\alpha} + \eta \tilde{\delta}^\top B^\top P B \tilde{\delta} \\ &= \mathbf{J}_{\alpha,\alpha}(x_1, x_2) + \tilde{\delta}^\top (\mathbf{R} + \eta B^\top P B)\tilde{\delta}. \end{aligned} \tag{3.54}$$

Consider now two policies $\boldsymbol{\pi}_1^\star, \boldsymbol{\pi}_2^\star$ differing from α only in the first action, and assume this first action to be optimal. Then, the two control policies provide the value function

$$\mathbf{J}_{\boldsymbol{\pi}_1^\star,\boldsymbol{\pi}_2^\star}(x_1, x_2) = \min_{\tilde{\boldsymbol{\pi}}} \{\tilde{x}^\top S(\bar{x})\tilde{x} + \tilde{\boldsymbol{\pi}}^\top \mathbf{R} \tilde{\boldsymbol{\pi}} + \eta \mathbf{J}_{\boldsymbol{\pi}_1^\star,\boldsymbol{\pi}_2^\star}(x^+_{1,\boldsymbol{\pi}_1^\star}, x^+_{2,\boldsymbol{\pi}_2^\star})\}. \tag{3.55}$$

However, in view of the above discussion and (3.54), the state-value function (3.55) is equivalent to

$$\mathbf{J}_{\boldsymbol{\pi}_1^\star,\boldsymbol{\pi}_2^\star}(x_1, x_2) = \min_{\tilde{\delta}} \{\mathbf{J}_{\alpha,\alpha}(x_1, x_2) + \tilde{\delta}^\top (\mathbf{R} + \eta B^\top P B)\tilde{\delta}\} = \mathbf{J}_{\alpha,\alpha}(x_1, x_2).$$

Then optimality of $\mathbf{J}_{\alpha,\alpha}$ follows by the recursive properties of Bellman's equations, thus showing optimality of α and concluding the proof. □

Remark 3.5. *The results of Theorem 3.3 can be generalized to time-varying dynamics. However, it is interesting to notice that, in the time-invariant scenario, contraction implies existence and global stability of a fixed point by Banach fixed point theorem [19]. As such, we can consider $x_2 = 0$ to be such a fixed point, without loss of generality. In turn, $\bar{x} = x_1$ and the objective* (3.45) *becomes*

$$J(x_1(\mathfrak{t}), u_1(\mathfrak{t})) = \sum_{k=0}^{\infty} \eta^k \left(x_1(k+\mathfrak{t})^\top S(x_1(k+\mathfrak{t}))x_1(k+\mathfrak{t}) + u_1(k+\mathfrak{t})\mathbf{R}u_1(k+\mathfrak{t})\right), \tag{3.56}$$

which is a single-agent minimization problem. Moreover, in the context of linear time-invariant dynamics, (3.56) *recovers the classical discounted LQR formulation, since S in* (3.46) *becomes the*

constant matrix

$$S := P - A^\top \left(\eta P - \sigma \eta^2 P B \left(R + \eta B^\top P B \right)^{-1} B^\top P \right) A.$$

DRL for contraction. Given the results of Theorem 3.3, one can envision learning a contractive feedback controller via data-driven methods. More specifically, we can define a two-agent problem where the aim is to minimize a cost of the form (3.45), yielding an instantaneous reward of the form

$$\mathbf{r}(x_1, x_2, u_1, u_2) := -(\tilde{x}^\top S(\bar{x})\tilde{x} + \tilde{u}\mathbf{R}\tilde{u}),$$

where $\mathbf{R} \in \mathbb{S}^{n_u}_{\succ 0}$ and $S : \mathbb{R}^{n_x} \to \mathbb{S}^{n_x}_{\succ 0}$ are user-defined. First, we notice that in order to obtain an approximation of the contractive controller (3.44) which is applicable in the single-agent scenario, the two agents cannot use information about the other agent states. Second, we remark that when $\mathbf{R}$ and S have been selected, we need a way of computing $\bar{s}$ in $\bar{x} = \bar{s}x_1 + (1 - \bar{s})x_2$. A thrilling possibility to avoid time-consuming online computations is the estimation of $\bar{s}$ via a DNN $\mathfrak{g} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_\theta} \to [0, 1]$ with parameters $\theta \in \mathbb{R}^{n_\theta}$. Given a dataset $\mathbb{D}$ of pairs (x_1, x_2), the DNN can be trained with the objective

$$\min_\theta \sum_{\mathbb{D}} \left| f(x_1) - f(x_2) - \frac{\partial f}{\partial x} \left(\mathfrak{g}(x_1, x_2, \theta)x_1 + (1 - \mathfrak{g}(x_1, x_2, \theta))x_2 \right) (x_1 - x_2) \right|^2 .$$

Once the offline training is complete, the DNN offers a fast estimation of the value $\bar{s}$ for two given states x_1, x_2, and it can be used in the computation of the cost for training the neural controller.

3.2 Robust synchronization via contraction theory

Multi-agent systems control has attracted a lot of attention from the control community. Many modern control problems can be formulated as networks of interacting agents that aim to achieve agreement [116]. In this section, we focus on the problem of state synchronization in an homogeneous network of discrete-time agents, namely, all agents are identical. The systems are described by time-varying nonlinear models that are linear in the control input.

The problem of synchronization presents mature results in the linear framework, especially for continuous-time agent dynamics [184, 205]. For discrete-time systems, notable results can be found in [53, 133, 179, 206]. However, in the discrete-time setting, further investigations are required to understand the relationship between the *structure* of a network (i.e., the eigenvalues and eigenvectors associated with its representation) and the possibility of finding a suitable synchronizing control. As a matter of fact, the structure of the communication graph has a significant impact on discrete-time networks compared to continuous-time ones, and, while graph normalization approaches like [97, 206] have proven effective, they may not always be feasible. Additionally, existing solutions in the nonlinear scenario often focus on specific agent structures, such as Lur'e system forms [194] or linear systems with saturated inputs [43], and commonly employ observer design [98, 128] or data-based optimization techniques [79].

In this section, propose a solution based on discrete-time contraction analysis and incremental stability. There are two main motivations for this choice. Firstly, contraction analysis allows the study of nonlinear systems via linear systems-like arguments. Hence, we can take inspiration from the well-established literature on linear systems [206] and provide a direct link

between the two scenarios. Secondly, incremental stability easily translates to synchronization of homogeneous networks. In such networks, each agent can be seen as a singular trajectory of the same system starting from different initial conditions. As trajectories of exponentially δISS (i.e. contractive) systems "forget" their initial conditions and distances between them exponentially decrease to zero, designing a distributed controller that makes the network dynamics a contraction indirectly solves the global exponential synchronization problem. Then, by making the network dynamics incrementally stable, we indirectly obtain robust state synchronization. We remark that, differently from convergent systems as in [165], incremental stability does not require the final trajectory to remain bounded [216, Section 4.3].

In what follows, we present two main contributions. Firstly, we establish new results on simultaneous stabilization and robust synchronization of discrete-time linear systems. Unlike [97, 179, 206], our approach does not rely on the normalized Laplacian and assumes that all local controllers are designed equally, without additional local degrees of freedom. This assumption aligns with many practical applications where agents only have access to aggregate information and lack knowledge about their neighbors in the graph or their *degree*, e.g. [60,158]. Thus, we provide conditions that encompass the normalized Laplacian as a subcase and we establish bounds that link the connectivity properties of the graph with the simultaneous stabilizability properties of the agents. Secondly, we address robust synchronization of homogeneous discrete-time nonlinear agents by leveraging the results on contraction analysis and incremental stability discussed in Section 3.1.2. Our analysis considers almost differentiable dynamics, linear input vector fields, and generic connected communication graphs (i.e., possibly directed and weighted). As for the linear case, we relate the connectivity properties of the graph with the simultaneous stabilizability properties of the agents.

For an introduction to fundamental concepts and results on graph theory relevant to the subsequent sections we refer to Appendix A.2.

3.2.1 The problem of multi-agent synchronization

Consider a homogeneous network of discrete-time agents, where the dynamics of each node is described by an input-affine, possibly time-varying, nonlinear difference equation of the form

$$x_i^+ = f(x_i, \mathfrak{t}) + Bu_i + w_i, \qquad i = 1, \dots, N, \tag{3.57}$$

where $f : \mathbb{R}^{n_x} \times \mathbb{N} \to \mathbb{R}^{n_x}$ is a continuous function of the state, $B \in \mathbb{R}^{n_x \times n_u}$ is full column rank, $x_i \in \mathbb{R}^{n_x}$ represents the state of node i and $u_i \in \mathbb{R}^{n_u}$ is the control input on node i, $w_i \in \mathbb{R}^{n_x}$ is the disturbance affecting node i entering in the input directions, at timestep $\mathfrak{t} \in \mathbb{N}$, and $x_i^+ \in \mathbb{R}^{n_x}$ represents the state of node i at timestep $\mathfrak{t} + 1$. We denote by $\mathcal{N}_i$ the set of neighbors of node i, and we define the state of the entire network and the entire disturbance as

$$\begin{aligned} \mathbf{x} &:= \operatorname{col}(x_1, \dots, x_N) \in \mathbb{R}^{Nn_x} \\ \mathbf{w} &:= \operatorname{col}(w_1, \dots, w_N) \in \mathbb{R}^{Nn_x}. \end{aligned} \tag{3.58}$$

Our synchronization objective is to design a distributed feedback control law of the form

$$u_i = \sum_{j \in \mathcal{N}_i} a_{ij} \Big[\alpha(x_i, \mathfrak{t}) - \alpha(x_j, \mathfrak{t}) \Big] = \sum_{j=1}^{N} \ell_{ij} \alpha(x_j, \mathfrak{t}) \tag{3.59}$$

for all $i = 1, \dots, N$, for some function $\alpha : \mathbb{R}^{n_x} \times \mathbb{N} \to \mathbb{R}^{n_u}$, that stabilizes the dynamics (3.57) on the so-called *synchronization manifold* $\mathcal{M}$ defined as

$$\mathcal{M} := \{\mathbf{x} \in \mathbb{R}^{Nn_x} \mid x_i = x_j, \text{ for all } i, j \in \{1, \dots, N\}\}, \tag{3.60}$$

where the states of all the agents of the network agree with each other. Furthermore, we require the control action u_i to be equal to zero on the synchronization manifold. In other words, when synchronization is achieved, no correction term is needed for each individual agent. As a consequence, independently stabilizing all the agents on a desired equilibrium point is not a valid solution in general. Note that the control law (3.59) satisfies this constraint by construction, due to the properties of the Laplacian matrix, see Appendix A.2. We consider the full-state information problem. Differently put, the i-th agent can use the complete state information x_j of its neighbors $j \in \mathcal{N}_i$ alongside its own local information x_i. We formalize our synchronization problem as follows.

Problem 3.1 (Robust network synchronization)**.** *The distributed feedback control law* (3.59) *solves the robust synchronization problem for the network* (3.58) *if there exists a function* α *and real numbers* $c \geqslant 1$*,* $\rho \in (0, 1)$ *and* $\gamma \geqslant 0$ *such that, for any initial condition* $(\mathbf{x}(\mathfrak{t}_0), \mathfrak{t}_0) \in \mathbb{R}^{Nn_x} \times \mathbb{N}$*, the solutions to the closed-loop system*

$$x_i^+ = f(x_i, \mathfrak{t}) + B \sum_{j=1}^{N} \ell_{ij} \alpha(x_j, \mathfrak{t}) + w_i, \quad i = 1, \dots, N.$$

satisfy for all $\mathfrak{t} \geqslant \mathfrak{t}_0$

$$|\mathbf{x}(\mathfrak{t})|_{\mathcal{M}} \leqslant c\, \rho^{\mathfrak{t}-\mathfrak{t}_0} |\mathbf{x}(\mathfrak{t}_0)|_{\mathcal{M}} + \sup_{\substack{t \in [\mathfrak{t}_0, \mathfrak{t}] \\ i,j \in [1,N]}} \gamma |w_i(t) - w_j(t)|, \tag{3.61}$$

where $\mathcal{M}$ *is defined in* (3.60)*.*

3.2.2 Continuous-time vs discrete-time synchronization

In order to understand the challenges of Problem 3.1, we start from the linear scenario and recall a few fundamental synchronization results. Consider a network of $N \in \mathbb{N}$ continuous-time linear systems of the form

$$\dot{x}_i = Ax_i + Bu_i, \tag{3.62}$$

with A in $\mathbb{R}^{n_x \times n_x}$, B in $\mathbb{R}^{n_x \times n_u}$ and u_i, selected, according to (3.59), as a diffusive coupling of the form

$$u_i = K \sum_{j=1}^{N} \ell_{ij} x_j. \tag{3.63}$$

It is well known that the solution to the synchronization problem is based on only two assumptions: the stabilizabilty of the pair (A, B) and the connectivity of the graph, see, e.g. [102]. This can be shown with an opportune change of coordinate recasting the synchronization problem as a stabilization one. As a matter of fact, the synchronization problem can be viewed as a simultaneous stabilization problem for the complex valued matrices

$$A + \lambda_i BK, \qquad i = 2, \dots, N \tag{3.64}$$

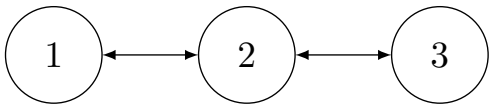


Figure 3.1: Simple undirected graph

with λ_i being the (non-zero) eigenvalues of the Laplacian matrix L. In continuous-time, the simultaneous stabilization of (3.64) via a unique gain K can be achieved by exploiting the so-called *infinite-gain margin property*[1] [191]. More specifically, the correction gain K of the controller (3.62) is obtained as a solution to the continuous-time algebraic Riccati equation (ARE) multiplied by a positive scalar gain, namely

$$\begin{aligned} &PA + A^\top P - PBR^{-1}B^\top P + Q = 0 \\ &K = -\kappa\, R^{-1}B^\top P \end{aligned} \tag{3.65}$$

where $\kappa > 0$ must be chosen large enough. The lower bound on gain κ is inversely proportional to the real part of the non-zero eigenvalue of the Laplacian matrix with the smallest real part. In few words, stabilization of the complex matrix $(A + \underline{\lambda}BK)$ with $\underline{\lambda}$ the eigenvalue of L with the smallest (non-zero) real part leads to the stabilization of all other complex matrices, thus leading to synchronization.

It is natural to hope that such a result can be extended to the discrete-time framework. Therefore, we consider a network of $N \in \mathbb{N}$ homogeneous agents whose dynamics can be described by a difference equation of the form

$$x_i^+ = Ax_i + Bu_i \tag{3.66}$$

with control input u_i defined as in (3.63). Similarly to the continuous-time framework, the discrete-time synchronization problem can be seen as the $N-1$ stabilization problems of (3.64) (see [97]). However, in the discrete-time framework, the infinite gain margin property does not hold. Hence, the simultaneous stabilization may not be achievable for arbitrary λ_i, thus imposing conditions on the solvability of the synchronization problem, e.g. [97, 206]. This phenomenon is illustrated in the following simple example.

Example 3.3. *Consider a network of $N = 3$ agents, whose dynamics are described by*

$$x_i^+ = 2x_i + u_i, \qquad x_i \in \mathbb{R}, \forall i \in \{1, 2, 3\} \tag{3.67}$$

with the simple (undirected) graph in Fig. 3.1 and Laplacian matrix

$$L = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}.$$

The eigenvalues of L are $\{0, 1, 3\}$. Indeed, by using the transformation

$$T = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & -\frac{1}{2} \\ -1 & 0 & 1 \end{pmatrix}$$

[1] Namely, if K stabilizes the pair (A, B), then aK is stabilizing for any $a \geqslant 1$.

one obtains

$$TLT^{-1} = \begin{pmatrix} 0 & -1 & -\frac{1}{2} \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Imposing a distributed control law of the form $u_i = \sum_{j=1}^{N} \ell_{ij} K x_j$ *with* $K \in \mathbb{R}$ *to be selected, using the transformation* T*, namely,*

$$\mathbf{x} \mapsto \begin{pmatrix} z \\ \mathbf{e} \end{pmatrix} = \begin{pmatrix} z \\ e_1 \\ e_2 \end{pmatrix} = \begin{pmatrix} x_1 \\ -\frac{1}{2}x_1 + x_2 - \frac{1}{2}x_3 \\ -x_1 + x_3 \end{pmatrix},$$

one obtains the following $(z, \mathbf{e})$ *system*

$$\begin{aligned} z^+ &= 2z - Ke_1 - \tfrac{1}{2}Ke_2\,, \\ \mathbf{e}^+ &= \begin{pmatrix} 2-3K & 0 \\ 0 & 2-K \end{pmatrix} \mathbf{e}\,. \end{aligned}$$

In order to guarantee state synchronization, one has to show that the $\mathbf{e}$*-component converges to zero, namely* K *should satisfy* $|2-3K| < 1$ *and* $|2-K| < 1$*. Clearly, such a* K *does not exist as the first inequality is satisfied for* $K \in (\frac{1}{3}, 1)$ *while the second one holds for* $K \in (1, 3)$*. In other words, the synchronization problem for this simple example with a scalar controllable plant and an undirected, not-weighted, connected Laplacian, which is diagonalizable with real eigenvalues, cannot be solved.*

In practical terms, stabilization of the pencil matrix associated to (3.64) in the discrete-time framework is achievable only for a set of eigenvalues whose norm lies inside a compact set. A possible workaround to this problem is presented in [179, 206] where the authors consider a *normalized* Laplacian matrix, namely each row of the Laplacian matrix is differently normalized based on the *in-degree* of each node [89]. Such a normalization allows containing the eigenvalues of L inside an opportune range where a unique solution for the $N-1$ stabilization of (3.64) indeed exists.

However, we aim at exploiting properties similar to the ones in continuous-time to provide general conditions on the spectrum of the Laplacian under which Problem 3.1 can be solved by controllers of the form (3.63). In the linear case, the resulting design is independent of any normalization, thus being applicable to a broad class of networked problems, such as *open networks* [76] or *switching networks* [40]. Normalization-based conditions can be recovered by restricting the considered Laplacians to the set of normalized ones. Furthermore, the proposed approach can be extended to the nonlinear case.

3.2.3 Synchronization of linear systems

Motivated by the previous discussion, we exploit simultaneous stabilization tools for solving the multi-agent synchronization problem. We revisit the results in [206] by providing a solution with non-normalized information exchange. Our solution applies to non-diagonalizable Laplacian matrices, thus covering the results of [143] as a corollary. We start by presenting necessary and sufficient conditions for state synchronization. Then, we propose a Riccati-based design allowing for stabilization with any Laplacian whose eigenvalues belong to a given compact set. We also show that such a design possesses gain margin properties. Finally, we specialize these results to the multi-agent framework.

Necessary and sufficient conditions. We start by presenting a general result for network synchronization for linear systems. It is shown that the existence of a common control law for systems associated with each non-zero eigenvalues of the Laplacian is equivalent to solving the synchronization problem. Hence, we extend [206, Lemma 1] to the case of general Laplacian matrices. While this section focuses on the full-state information case, we remark that it can be straightforwardly extended to the static output-feedback scenario.

Theorem 3.4. *For a network of $N \in \mathbb{N}$ agents described by dynamics*

$$x_i^+ = Ax_i + Bu_i + w_i \tag{3.68}$$

The diffusive control law u_i defined in (3.63) *solves Problem 3.1 if and only if*

(i) the interconnection graph (possibly directed, weighted) $\mathcal{G}$ is connected or matrix A is Schur stable, and

(ii) the gain K is such that matrix $(A + \lambda BK)$ is Schur stable for any $\lambda \in \operatorname{spec}(L) \setminus \{0\}$.

Proof. *Sufficiency.* Using the Kronecker notation, the closed-loop network dynamics can be written as

$$\mathbf{x}^+ = ((\mathrm{I}_N \otimes A) + (L \otimes BK))\mathbf{x} + \mathbf{w}, \tag{3.69}$$

with $\mathbf{x}$ and $\mathbf{w}$ defined in (3.58). To show convergence to the synchronization manifold $\mathcal{M}$, define a virtual leader as the node providing connectivity characterized in Definition A.1. Without loss of generality, assume $z := x_1$ to be such a node. Recalling the Laplacian structure (A.5), define $N-1$ error coordinates $\mathbf{e} := \operatorname{col}(e_2, \dots, e_N) \in \mathbb{R}^{Nn_x}$ with $e_i := x_i - z$. Compactly, this reads as

$$\begin{pmatrix} z \\ \mathbf{e} \end{pmatrix} := (T \otimes \mathrm{I}_{n_x})\mathbf{x} = \left(\begin{pmatrix} 1 & \mathbf{0}^\top \\ -\mathbf{1} & \mathrm{I}_{N-1} \end{pmatrix} \otimes \mathrm{I}_{n_x} \right) \mathbf{x}, . \tag{3.70}$$

where we observe that $T^{-1} = \left(\begin{smallmatrix} 1 & \mathbf{0}^\top \\ \mathbf{1} & \mathrm{I}_{N-1} \end{smallmatrix}\right)$ so that, according to the partitioning in (A.5), we have

$$\begin{pmatrix} 1 & \mathbf{0}^\top \\ -\mathbf{1} & \mathrm{I}_{N-1} \end{pmatrix} L \begin{pmatrix} 1 & \mathbf{0}^\top \\ \mathbf{1} & \mathrm{I}_{N-1} \end{pmatrix} = \begin{pmatrix} 0 & L_{12} \\ \mathbf{0} & L_{22} - \mathbf{1}\, L_{12} \end{pmatrix}.$$

Exploiting the structure of T, we obtain the error dynamics

$$\begin{aligned} \mathbf{e}^+ &= \begin{pmatrix} \mathbf{0} & \mathrm{I}_{N-1} \end{pmatrix} \left(((TT^{-1} \otimes A) + (TLT^{-1} \otimes BK)) \begin{pmatrix} z \\ \mathbf{e} \end{pmatrix} + (T \otimes \mathrm{I}_{n_x})\mathbf{w} \right) \\ &= A_{\mathrm{cl}}\mathbf{e} + \widetilde{\mathbf{w}}\,, \end{aligned} \tag{3.71a}$$

where we defined the closed-loop matrix A_{cl} as

$$A_{\mathrm{cl}} := (\mathrm{I}_{N-1} \otimes A) + ((L_{22} - \mathbf{1}\, L_{12}) \otimes BK) \tag{3.71b}$$

and $\widetilde{\mathbf{w}} := \operatorname{col}(\widetilde{w}_2, \dots, \widetilde{w}_N) \in \mathbb{R}^{Nn_x}$ with $\widetilde{w}_i := w_i - w_1$. If A_{cl} is Schur stable, the use of standard arguments for linear systems yields

$$|\mathbf{e}(\mathrm{t})| \leqslant c\, \rho^{\mathrm{t}-\mathrm{t}_0}\, |\mathbf{e}(\mathrm{t}_0)| + \sup_{t \in [\mathrm{t}_0, \mathrm{t}]} \gamma |\widetilde{\mathbf{w}}(t)|, \tag{3.72}$$

for some $c, \gamma > 0$ and $\rho \in (0,1)$. Since T in (3.70) has a bounded norm, we have, for some $c_1 > 0$,

$$\begin{aligned}|\mathbf{e}|^2 &= \left|\begin{pmatrix} z \\ \mathbf{e}\end{pmatrix} - \begin{pmatrix} z \\ \mathbf{0}\end{pmatrix}\right|^2 = \inf_{z^\star \in \mathbb{R}^{n_x}} \left|\begin{pmatrix} z \\ \mathbf{e}\end{pmatrix} - \begin{pmatrix} z^\star \\ \mathbf{0}\end{pmatrix}\right|^2 = \inf_{\mathbf{x}^\star \in \mathcal{M}} |(T \otimes \mathrm{I}_{n_x})\mathbf{x} - (T \otimes \mathrm{I}_{n_x})\mathbf{x}^\star|^2 \\ &\leqslant \inf_{\mathbf{x}^\star \in \mathcal{M}} |(T \otimes \mathrm{I}_{n_x})|^2 |\mathbf{x} - \mathbf{x}^\star|^2 \\ &\leqslant \inf_{\mathbf{x}^\star \in \mathcal{M}} c_1^{-1} |\mathbf{x} - \mathbf{x}^\star|^2 = c_1^{-1} |\mathbf{x}|_{\mathcal{M}}^2\end{aligned}$$

Also, since T is invertible, for some $c_2 > 0$,

$$\begin{aligned}|\mathbf{x}|_{\mathcal{M}}^2 &= \inf_{\mathbf{x}^\star \in \mathcal{M}} |\mathbf{x} - \mathbf{x}^\star|^2 = \inf_{\mathbf{x}^\star \in \mathcal{M}} |(T^{-1} \otimes \mathrm{I}_n)(T \otimes \mathrm{I}_{n_x})(\mathbf{x} - \mathbf{x}^\star)|^2 \\ &\leqslant \inf_{\mathbf{x}^\star \in \mathcal{M}} c_2 |(T \otimes \mathrm{I}_{n_x})\mathbf{x} - (T \otimes \mathrm{I}_{n_x})\mathbf{x}^\star|^2 \\ &\leqslant \inf_{z^\star \in \mathbb{R}^{n_x}} c_2 \left|\begin{pmatrix} z \\ \mathbf{e}\end{pmatrix} - \begin{pmatrix} z^\star \\ \mathbf{0}\end{pmatrix}\right|^2 = c_2 |\mathbf{e}|^2.\end{aligned}$$

Then, we obtain the following relations

$$\sqrt{c_1}|\mathbf{e}| \leqslant |\mathbf{x}|_{\mathcal{M}} \leqslant \sqrt{c_2}|\mathbf{e}|, \quad |\widetilde{\mathbf{w}}| \leqslant \sup_{i,j \in [1,N]} c_3 |w_i - w_j|, \tag{3.73}$$

for some $c_1, c_2, c_3 > 0$. As a consequence, if A_{cl} is Schur stable, one obtains robust synchronization as in Problem 3.1. Therefore, in the rest of the proof we set $\widetilde{\mathbf{w}} = 0$ and we show that A_{cl} in (3.71b) is Schur stable. Let $T_J \in \mathbb{C}^{(N-1)\times(N-1)}$ be a transformation such that $\mathcal{L} = T_J(L_{22} - \mathbf{1}\, L_{12})T_J^{-1}$ is in Jordan canonical form. By defining the change of coordinates $\widehat{A}_{\text{cl}} = (T_J \otimes \mathrm{I}_{N-1})A_{\text{cl}}(T_J^{-1} \otimes \mathrm{I}_{N-1})$, Schur stability of $\widehat{A}_{\text{cl}}$ implies Schur stability of A_{cl}. By the properties of the Kronecker product and (3.71b), we have

$$\widehat{A}_{\text{cl}} = (\mathrm{I}_{N-1} \otimes A) + (\mathcal{L} \otimes BK). \tag{3.74}$$

Since $\mathcal{L}$ is in its Jordan form, the former matrix is block triangular with diagonal block equal to $(A + \lambda BK)$ with λ in $\operatorname{spec}(\mathcal{L})$. Hence, Schur stability of $\widehat{A}_{\text{cl}}$ holds if and only if the complex matrix $(A + \lambda BK)$ is Schur stable for all $\lambda \in \operatorname{spec}(\mathcal{L})$. Due to the similarity transformations,

$$\operatorname{spec}(\mathcal{L}) = \operatorname{spec}(L_{22} - \mathbf{1}\, L_{12}) = \operatorname{spec}(L) \setminus \{\lambda_1\},$$

where $\lambda_1 = 0$ is the eigenvalue associated to the eigenvector $\mathbf{1}$. Consider now the case in item *(i)* where the graph is connected. By Lemma A.1, L has only one zero eigenvalue and the gain K is such that $(A + \lambda BK)$ is Schur stable for all $\lambda \in \operatorname{spec}(\mathcal{L})$, which implies Schur stability of A_{cl} in (3.71b). If instead A is Schur in item *(i)*, then $(A - \lambda BK)$ is Schur stable for all eigenvalues λ of L (including the zero ones) and A_{cl} in (3.71b) is exponentially stable.

Necessity. Consider a Laplacian matrix of the form (A.5). Following the lines of the sufficiency proof, the error dynamics between agents and a virtual leader are described by (3.71a). We first study the connectivity requirement. Suppose that synchronization is

achieved, A is unstable and at least one agent is not connected. Without loss of generality, assume x_1 to be such a node. Since it is not connected, the Laplacian takes the form

$$L = \begin{pmatrix} 0 & \mathbf{0}^\top \\ \mathbf{0} & L' \end{pmatrix},$$

where L' is the Laplacian matrix of the connected portion of the graph. Then, by (3.71a) with $\widetilde{\mathbf{w}} = 0$, we have

$$\mathbf{e}^+ = ((\mathrm{I}_N \otimes A) + (L' \otimes BK))\mathbf{e}\,.$$

Notice that L' describes a connected graph. Then, by Lemma A.1, it has one zero eigenvalue. By performing similar steps to the ones in the sufficiency proof, we define the transformed closed-loop matrix

$$\widehat{A}_{\text{cl}} = (\mathrm{I}_{N-1} \otimes A) + (\mathcal{L}' \otimes BK),$$

where $\mathcal{L}'$ is in Jordan form. Note that $\widehat{A}_{\text{cl}}$ is Schur stable if and only if the complex matrix $(A+\lambda BK)$ is Schur stable for all $\lambda \in \operatorname{spec}(\mathcal{L}')$. However, $\operatorname{spec}(\mathcal{L}')$ includes a zero eigenvalue. Hence, $\widehat{A}_{\text{cl}}$ is stable if and only if A is Schur stable, showing the first item by establishing contradiction. We now prove the necessity of item *(ii)*. If the agents are synchronized, the $\mathbf{e}$ subsystem in (3.71a) is asymptotically stable and the matrix $\widehat{A}_{\text{cl}}$ in (3.74) is Schur. Since $\mathcal{L}$ contains all the nonzero eigenvalues of L and $\widehat{A}_{\text{cl}}$ is block-upper triangular, item *(ii)* must hold, and this concludes the proof. □

Gain margin computation via Riccati design. The results presented in the above paragraph are not constructive. Therefore, following a similar approach to the one in [206], we provide a design procedure for pencil matrices stabilization, namely we consider a single discrete-time agent. Then, in the next paragraph, we will apply this result to the case of networks.

Consider a discrete-time linear system described by

$$x^+ = Ax + Bu\,, \tag{3.75}$$

with $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^n_u$ and, without loss of generality, B is assumed to be full-column rank. The goal is to find a gain matrix $K \in \mathbb{R}^{n_u \times n_x}$ such that a state feedback control law $u = Kx$ makes the complex closed-loop matrix $(A + \lambda BK)$ Schur for some complex numbers λ. Inspired by Definition 3.4, we formally define this notion as follows.

Definition 3.5 (Complex gain margin for linear systems)**.** *The matrix K is said to have a complex gain margin with radius $r > 0$ if $A + \lambda BK$ is Schur for any λ in $\{\lambda \in \mathbb{C} : |\lambda - 1| \leqslant r\}$.*

To find the complex gain margin of a matrix K, we propose a solution based on the discrete-time Modified Algebraic Riccati Inequality (MARI) [43, 199, 221] defined as

$$A^\top PA - \sigma A^\top PB(R + B^\top PB)^{-1}B^\top PA \preceq \rho P\,, \tag{3.76}$$

where $R \in \mathbb{S}^{n_u}_{\succeq 0}$, $P \in \mathbb{S}^{n_x}_{\succ 0}$ and generally $\sigma \in (0, 1]$, $\rho \in (0, 1)$. Note that, since B is assumed to be full column rank, the matrix $R + B^\top PB$ is positive definite and, consequently, invertible. The main difference between the MARI (3.76) and the more common discrete-time Algebraic Riccati Inequality (DARI)

$$A^\top PA - A^\top PB(R + B^\top PB)^{-1}B^\top PA + Q \prec P \tag{3.77}$$

lies in the presence of the scalar σ. First, note that if $R \succ 0$ the positive semi-definite matrix Q can be embedded in the right-hand side of (3.76) by exploiting $\rho P \prec P - Q$, which holds for a

suitable $\rho \in (0,1)$ as long as $P - Q \succ 0$. Inequality $P - Q \succ 0$ holds when $R \in \mathbb{S}^{n_u}_{\succ 0}$ since one can rearrange (3.77) as

$$A^\top [P - PB(R + B^\top PB)^{-1} B^\top P] A \prec P - Q$$

and, by applying the block matrix inversion identity

$$(\mathcal{A} + \mathcal{B}\mathcal{D}^{-1}\mathcal{C})^{-1} = \mathcal{A}^{-1} - \mathcal{A}^{-1}\mathcal{B}(\mathcal{D} + \mathcal{C}\mathcal{A}^{-1}\mathcal{B})^{-1}\mathcal{C}\mathcal{A}^{-1}\,, \tag{3.78}$$

with $\mathcal{A}^{-1} = P$, $\mathcal{D} = R$, $\mathcal{C} = B^\top$, $\mathcal{B} = B$, we obtain

$$0 \prec A^\top (P^{-1} + BR^{-1}B^\top)^{-1} A \prec P - Q\,. \tag{3.79}$$

Due to the discussion above, the DARI (3.77) is a special case of the MARI (3.76) when $\sigma = 1$ and R is positive definite. As such, the MARI allows for an extra degree of freedom. Its role is to weigh the impact of the input on the solution to the inequality. In other words, the smaller the σ, the less we can rely on the input to stabilize the system. This is evident for the special case $\sigma = 0$, where the MARI boils down to the Lyapunov inequality for autonomous discrete-time systems.

We show next that the degree of freedom offered by the MARI (3.76) allows stating sufficient conditions for the existence of a state feedback gain K solving the simultaneous stabilization problem. This result reinterprets the findings of [43, Theorem 1].

Proposition 3.7. *Let the pair (A, B) be stabilizable and $R \in \mathbb{S}^{n_u}_{\succ 0}$. Let $P \in \mathbb{S}^{n_x}_{\succ 0}$ be a solution to the MARI* (3.76) *for some $\sigma \in (0, 1]$ and for some $\rho \in (0, 1)$. Then the matrix*

$$K = -(R + B^\top PB)^{-1} B^\top PA\,, \tag{3.80}$$

has a complex gain margin with radius $r = \sqrt{1 - \sigma}$.

Proof. Consider the closed-loop matrix $(A + \lambda BK)$ for some arbitrary $\lambda \in \mathbb{C}$ and define the following matrix

$$\Gamma := (A^\top + \lambda^* K^\top B^\top) P (A + \lambda BK), \tag{3.81}$$

with P solution to (3.76) and superscript $*$ denoting the complex conjugate. Substituting K in (3.80) into (3.81) and by defining $Y := Y^\top = (R + B^\top PB)^{-1}$ for the sake of compactness, we obtain

$$\Gamma = (A^\top - \lambda^* A^\top PBYB^\top) P (A - \lambda BYB^\top PA).$$

Expanding the product and adding and subtracting $\sigma A^\top PBYB^\top PA$, by virtue of (3.76) and using $\lambda\lambda^* = |\lambda|^2$, $\lambda + \lambda^* = 2\mathfrak{Re}(\lambda)$, we obtain

$$\Gamma \preceq \rho P + A^\top PBY((\sigma - 2\mathfrak{Re}(\lambda))\,\mathrm{I}_{n_x} + |\lambda|^2 B^\top PBY) B^\top PA\,.$$

Since $R \succeq 0$, by the definition of Y we obtain $YB^\top PBY \preceq YY^{-1}Y \preceq Y$ which implies

$$\Gamma \preceq \rho P + (\sigma - 2\mathfrak{Re}(\lambda) + |\lambda|^2) A^\top PBYB^\top PA.$$

Since $Y \succ 0$, the second term at the right-hand side is negative semidefinite if $\sigma - 2\mathfrak{Re}(\lambda) + |\lambda|^2 \leqslant 0$. Recall that $|\lambda|^2 = \mathfrak{Re}(\lambda)^2 + \mathfrak{Im}(\lambda)^2$ and define the real scalars $\lambda_R := \mathfrak{Re}(\lambda) - 1$,

$\lambda_I := \mathfrak{Im}(\lambda)$. From the previous inequality, $\Gamma \preceq \rho P$ if

$$\sigma - 2(\lambda_R + 1) + (\lambda_R + 1)^2 + \lambda_I^2 \leqslant 0 \iff \lambda_R^2 + \lambda_I^2 \leqslant 1 - \sigma\,.$$

By the definition of λ_R and λ_I, these inequalities characterize the circle of radius $r = \sqrt{1-\sigma}$ centered at the point $c = (1, 0)$ of the complex plane. Therefore, if $|\lambda - 1| \leqslant r$ we have $\Gamma \preceq \rho P$ and consequently

$$(A^\top + \lambda^* K^\top B^\top) P (A + \lambda BK) - P \preceq -(1-\rho)P. \tag{3.82}$$

By [208, Theorem 3.2], since $P \in \mathbb{S}^{n_x}_{\succ 0}$, the complex closed-loop matrix $(A + \lambda BK)$ is Schur stable, which concludes the proof. □

Remark 3.6. *Since the DARI can be seen as a special case of the MARI with $\sigma = 1$, the gain margin radius r degenerates to zero. This drastically reduces the set of simultaneously stabilizable matrices, as only infinitesimal variations from the value $\lambda = 1$ are allowed thanks to continuity.*

Remark 3.7. *We remark that in [206] the use of the MARI is discouraged, as it is stated that no standard algorithm exists to provide a solution. Also, the authors state that it is not clear when such a solution exists. However, we highlight that recent results showed that LMI approaches provide useful tools for finding such a solution, see e.g., [199, 221]. Moreover, concerning the existence of positive definite stabilizing solutions of the MARI* (3.76)*, we refer to [43, Proposition 3]. In simple words, the authors of [43] prove the existence of at least one stabilizing positive definite solution to the MARI for $\sigma \in (\underline{\sigma}, 1]$, where $\underline{\sigma} > 0$ depends on R, ρ and the most unstable eigenvalue of A. In particular, stabilizing solutions were shown to exist when the parameter is sufficiently close to* 1 *and $R \succ 0$ (i.e., when we are sufficiently close to the standard DARI), see [43, 199].*

Robust linear synchronization. We now exploit Proposition 3.7 for discrete-time linear network synchronization. We consider a network of systems (3.68) and combine the results of Theorem 3.4 and Proposition 3.7 to design the state-feedback gain K inducing synchronization over general time-invariant graphs. To this end, define the following quantities:

$$\eta_i := \left(\frac{|\lambda_i|}{\mathfrak{Re}(\lambda_i)}\right)^2 = 1 + \left(\frac{\mathfrak{Im}(\lambda_i)}{\mathfrak{Re}(\lambda_i)}\right)^2,\ i = 2, \dots, N, \tag{3.83a}$$

$$\bar{\eta} := \max\{\eta_2, \dots, \eta_N\}, \quad \underline{\eta} := \min\{\eta_2, \dots, \eta_N\}, \tag{3.83b}$$

$$\bar{\lambda} := \max_{i \in \{2,\dots,N\}} \mathfrak{Re}(\lambda_i), \quad \underline{\lambda} := \min_{i \in \{2,\dots,N\}} \mathfrak{Re}(\lambda_i), \tag{3.83c}$$

for the non-zero eigenvalues λ_i, $i = 2, \dots, N$, of a connected Laplacian matrix L. Our MARI-based design is effective whenever the following inclusion holds for the graph-induced quantities (3.83) and the MARI parameter $\sigma \in (0, 1]$:

$$\bar{\eta}\sigma \in \left(0,\, 1 - \frac{(\bar{\eta}\,\bar{\lambda} - \underline{\eta}\,\underline{\lambda})^2}{(\bar{\eta}\,\bar{\lambda} + \underline{\eta}\,\underline{\lambda})^2}\right]. \tag{3.84}$$

The following lemma establishes a useful implication of (3.84).

Lemma 3.1. *Consider the quantities $\underline{\eta}$, $\bar{\eta}$, $\underline{\lambda}, \bar{\lambda}$ in* (3.83) *and let $\sigma \in (0, 1]$. The following interval of the real axis*

$$\mathcal{K} := \left[\frac{1 - \sqrt{1 - \bar{\eta}\,\sigma}}{\underline{\eta}\,\underline{\lambda}},\ \frac{1 + \sqrt{1 - \bar{\eta}\,\sigma}}{\bar{\eta}\,\bar{\lambda}}\right] \tag{3.85}$$

is nonempty if and only if (3.84) *holds.*

Proof. It is trivial that (3.84) implies $\bar{\eta}\sigma \leqslant 1$ and then the square root in (3.1) is well defined. To complete the proof, we show that the bound

$$\frac{1-\sqrt{1-\bar{\eta}\sigma}}{\underline{\eta}\underline{\lambda}} \leqslant \frac{1+\sqrt{1-\bar{\eta}\sigma}}{\bar{\eta}\bar{\lambda}} \tag{3.86}$$

holds if and only if (3.84) is satisfied, namely if and only if

$$\bar{\eta}\,\sigma \leqslant 1 - \frac{(\bar{\eta}\,\bar{\lambda} - \underline{\eta}\,\underline{\lambda})^2}{(\bar{\eta}\,\bar{\lambda} + \underline{\eta}\,\underline{\lambda})^2} = \frac{4\bar{\eta}\,\underline{\eta}\,\bar{\lambda}\,\underline{\lambda}}{(\bar{\eta}\,\bar{\lambda} + \underline{\eta}\,\underline{\lambda})^2} = \left(\frac{2\bar{\eta}\,\bar{\lambda}}{\bar{\eta}\,\bar{\lambda} + \underline{\eta}\,\underline{\lambda}}\right)^2 \frac{\underline{\eta}\,\underline{\lambda}}{\bar{\eta}\,\bar{\lambda}}\,,$$

which, due to the positivity of the squared term, is equivalent to

$$\left(\frac{\bar{\eta}\,\bar{\lambda} + \underline{\eta}\,\underline{\lambda}}{2\bar{\eta}\,\bar{\lambda}}\right)^2 \bar{\eta}\,\sigma - \frac{\underline{\eta}\,\underline{\lambda}}{\bar{\eta}\,\bar{\lambda}} \leqslant 0\,. \tag{3.87}$$

Thus, we must show that (3.86) $\iff$ (3.87). By the lower bound of (3.84), $\bar{\eta}\,\sigma > 0$. Then, multiplying (3.87) by $\bar{\eta}\,\sigma$ paired with addition and subtraction of $1 - \bar{\eta}\,\sigma$ at the right-hand side yields the equivalent inequality

$$1 - \bar{\eta}\,\sigma \geqslant \left(\frac{\bar{\eta}\,\bar{\lambda} + \underline{\eta}\,\underline{\lambda}}{2\bar{\eta}\,\bar{\lambda}}\right)^2 (\bar{\eta}\,\sigma)^2 + 1 - \left(1 + \frac{\underline{\eta}\,\underline{\lambda}}{\bar{\eta}\,\bar{\lambda}}\right)\bar{\eta}\,\sigma \geqslant \left(1 - \frac{\bar{\eta}\,\bar{\lambda} + \underline{\eta}\,\underline{\lambda}}{2\bar{\eta}\,\bar{\lambda}}\,\bar{\eta}\,\sigma\right)^2. \tag{3.88}$$

By taking the square root, (3.88) is equivalent to

$$\sqrt{1-\bar{\eta}\,\sigma} \geqslant 1 - \frac{\bar{\eta}\,\bar{\lambda} + \underline{\eta}\,\underline{\lambda}}{2\bar{\eta}\,\bar{\lambda}}\bar{\eta}\,\sigma, \tag{3.89}$$

where the right-hand side is non-negative because $\underline{\eta}\,\underline{\lambda} \leqslant \bar{\eta}\,\bar{\lambda}$ and $\bar{\eta}\,\sigma \leqslant 1$. Exploiting the expansion $\bar{\eta}\,\sigma = (1-\sqrt{1-\bar{\eta}\,\sigma})(1+\sqrt{1-\bar{\eta}\,\sigma}) > 0$, inequality (3.89) is equivalent to

$$\begin{aligned}\frac{\underline{\eta}\,\underline{\lambda}}{\bar{\eta}\,\bar{\lambda}} &\geqslant \left(1-\sqrt{1-\bar{\eta}\,\sigma}\right)\frac{2}{\bar{\eta}\,\sigma} - 1 \geqslant \frac{1+(1-\bar{\eta}\,\sigma)-2\sqrt{1-\bar{\eta}\,\sigma}}{\bar{\eta}\,\sigma} \\ &\geqslant \frac{\left(1-\sqrt{1-\bar{\eta}\,\sigma}\right)^2}{(1-\sqrt{1-\bar{\eta}\,\sigma})(1+\sqrt{1-\bar{\eta}\,\sigma})} \geqslant \frac{1-\sqrt{1-\bar{\eta}\,\sigma}}{1+\sqrt{1-\bar{\eta}\,\sigma}}\,,\end{aligned}$$

which coincides with (3.86), thus completing the proof. □

We are ready to present the main result on robust synchronization of linear systems.

Theorem 3.5. *Consider the network* (3.68) *and suppose that L is a Laplacian matrix describing a connected, directed and weighted communication graph. Let $R \in \mathbb{S}^{n_u}_{\succeq 0}$ and suppose that there exists $P \in \mathbb{S}^{n_x}_{\succ 0}$ such that* (3.76) *holds for a selection of σ satisfying* (3.84) *with $\bar{\eta}, \underline{\eta}, \bar{\lambda}, \underline{\lambda}$ defined in* (3.83)*. Then, the distributed control law u_i in* (3.59)*, with $\alpha(x) = \kappa K x$ and K selected as in* (3.80)*, solves Problem 3.1 for any scalar gain $\kappa \in \mathcal{K}$ as defined in* (3.85)*.*

Before proving Theorem 3.5, we highlight the importance of bounds (3.84) on σ in Proposition 3.5. First, differently from the continuous-time scenario [102, Chapter 5], the bounds on the scalar gain κ depend on the imaginary part of the Laplacian eigenvalues via $\underline{\eta}$ and $\bar{\eta}$. This is

expected, as discrete-time stability requires the eigenvalues to lay inside the unit disc, and not in the negative half-plane. Hence, in the case where there is at least one complex eigenvalue, definitions (3.83) imply $\bar{\eta} > 1$. As a consequence, it is necessary that $\sigma < 1$ for a real solution to the square roots in (3.85) to exist. In the case of real eigenvalues, $\eta_i = 1$ for all $i = 1, \dots, N$. Then, (3.85) simplifies to

$$\kappa \in \left[\frac{1 - \sqrt{1-\sigma}}{\underline{\lambda}} \, , \, \frac{1 + \sqrt{1-\sigma}}{\bar{\lambda}} \right],$$

with condition (3.84) on σ simplified to

$$\sigma \in \left(0, \, 1 - \frac{(\bar{\lambda} - \underline{\lambda})^2}{(\bar{\lambda} + \underline{\lambda})^2} \right].$$

This last bound recovers the results in [53], where all eigenvalues of L are supposed to be real. We emphasize that $\sigma = 1$ is a worst-case value not leading to synchronization unless $\underline{\lambda} = \bar{\lambda}$, as in Remark 3.6. Moreover, smaller values of σ lead to robust synchronization over broader range of graphs. We now show the proof for Theorem 3.5.

Proof. Theorem 3.5 By Theorem 3.4, Problem 3.1 is solved (equivalently, (3.61) holds) if the matrices $(A + \lambda_i \kappa B K)$ are Schur for all $\lambda_i \in \operatorname{spec}(L) \setminus 0$. By Proposition 3.7, each one of these matrices is Schur if $|\kappa\lambda_i - 1|^2 \leqslant 1 - \sigma$. By expanding the norm, we conclude that the closed-loop matrix associated to λ_i is Schur if $\sigma - 2\kappa\, \mathfrak{Re}(\lambda_i) + \kappa^2 |\lambda_i|^2 \leqslant 0$. Solving for κ and recalling the definition of η_i in (3.83), we obtain robust synchronization if

$$\kappa \in \left[\frac{1 - \sqrt{1 - \eta_i\, \sigma}}{\eta_i\, \mathfrak{Re}(\lambda_i)}, \frac{1 + \sqrt{1 - \eta_i\, \sigma}}{\eta_i\, \mathfrak{Re}(\lambda_i)} \right] \; \forall i = 2, \dots, N, \tag{3.90}$$

because we simultaneously stabilize all the closed-loop matrices. First, note that from (3.83c) we have $\eta_i \geqslant 1$. Moreover, $\eta_i < \infty$, because all eigenvalues λ_i have positive real part. Then, since $\sigma > 0$, for any $i = 2, \dots, N$ it holds that

$$\frac{1 + \sqrt{1 - \bar{\eta}\sigma}}{\bar{\eta}\bar{\lambda}} \leqslant \frac{1 - \sqrt{1 - \eta_i \sigma}}{\eta_i \mathfrak{Re}(\lambda_i)} \leqslant \frac{1 - \sqrt{1 - \bar{\eta}\sigma}}{\underline{\eta}\underline{\lambda}}.$$

Consequently, for any $\kappa \in \mathcal{K}$ as per (3.85), condition (3.90) holds and (3.61) holds, as to be proven. □

Remark 3.8. *We highlight that, in combination with the results in [43], condition* (3.84) *implies that there exist* $(\underline{\sigma}, \bar{\sigma}) \in (0, 1]^2$ *such that synchronizing solutions to the multi-agent problem based on MARI-design exist for some* σ *satisfying* $\sigma \geqslant \underline{\sigma}$ *and* $\sigma \leqslant \bar{\sigma}$*. However, this set is not guaranteed to be nonempty. In other words, it may happen that* $\underline{\sigma} > \bar{\sigma}$*. In particular, for unstable linear systems, the lower bound* $\underline{\sigma}$ *depends on the choice of* R, ρ *in* (3.76) *and the most unstable eigenvalue of* A *[43]. The upper bound* $\bar{\sigma}$ *depends on the Laplacian eigenvalues. The first implication is that it may not be possible to synchronize a network of arbitrarily unstable systems under an arbitrary communication graph. Similarly, some choices of convergence rate-control penalty pair* (ρ, R) *may not be suitable for a given system-graph pair.*

3.2.4 Synchronization of nonlinear systems

As a follow-up to the linear time-invariant scenario, in this subsection we present the design procedure of robust controllers for nonlinear multi-agent synchronization. It is worth noting

that the result presented in the following hold implicitly for linear time varying and parameter varying systems. For the sake of generality, they are all presented in the nonlinear framework. Motivated by the linear framework and the use of robust stabilization tools, we explore the design of robust stabilizers in the nonlinear framework by means of incremental input-to-state stability (δISS) arguments presented in Section 3.1.2.

The link between Proposition 3.7 and Theorem 3.2 is evident. However, since the Jordan transformation used in Section 3.2.3 cannot be easily applied in the nonlinear scenario, we will exploit the novel results of Lemma A.2 to design an appropriate transformation. To the best of the authors' knowledge, while a general contraction-based approach appeared in [4], there is no result paralleling Theorem 3.6 in the continuous-time framework, nor there is a Lyapunov-based result addressing robust exponential synchronization of nonlinear agents under general weighted, directed graphs. The main issue in continuous-time arises when considering a Lyapunov function of the form (3.98). Indeed, it is not trivial to perform continuous-time parallel steps similar to those at the end of the proof of Theorem 3.6, which exploit Demidovich-like conditions [163, Theorem 1] to derive upper bounds on the Lyapunov decrease. We now present the following main result on network synchronization of nonlinear systems.

Theorem 3.6. *Consider the network* (3.57) *and suppose that f satisfies Property 3.1 for some $\mathcal{D}f \subset \mathbb{R}^{n_x \times n_x}$ and L is a Laplacian matrix describing a connected, directed and weighted communication graph. Let $\rho \in (0,1)$ and $\sigma \in (0,1]$ satisfy*

$$\rho \leqslant \rho_M, \qquad \sigma \leqslant \frac{1}{\varsigma}, \qquad \varsigma := \left(\frac{\bar{\mu}}{\underline{\mu}}\right)^2, \tag{3.91}$$

with $\rho_M, \underline{\mu}, \bar{\mu}$ as in Lemma A.2. If, for some $R \in \mathbb{S}^{n_u}_{\succ 0}$, there exists $P \in \mathbb{S}^{n_x}_{\succ 0}$ satisfying (3.16a), (3.16b)*, then, the distributed control law $u_i = \sum_{j=1}^{N} \ell_{ij}\alpha(x_j, \mathfrak{t})$ in* (3.59) *with α defined as in* (3.17) *and κ satisfying*

$$\kappa \in \left[\frac{\left(1-\sqrt{1-\varsigma\sigma}\right)}{\varsigma\underline{\mu}}\,,\ \frac{\left(1+\sqrt{1-\varsigma\sigma}\right)}{\varsigma\underline{\mu}}\right], \tag{3.92}$$

solves Problem 3.1 for the network (3.57)*, namely,* (3.61) *holds.*

Proof. Mimicking the linear framework, we show convergence to the synchronization manifold $\mathcal{M}$ by focusing our analysis on the error between agents. If these error dynamics are robustly stable (ISS) with respect to an incremental version of $\mathbf{w}$, then Problem 3.1 is solved. Bearing in mind the steps of the proof of Theorem 3.4, we define a virtual leader and consider, without loss of generality, x_1 to be such a node. Next, define $N-1$ error coordinates with respect to such a leader node, $\mathbf{e} := \mathrm{col}(e_2, \dots, e_N) \in \mathbb{R}^{Nn_x}$ with $e_i := x_i - z$ for all $i = 1, \dots, N$ and $z = x_1$. Similarly, we define the incremental disturbance $\widetilde{\mathbf{w}} := \mathrm{col}(\widetilde{w}_2, \dots, \widetilde{w}_N)$ with $\widetilde{w}_i := w_i - w_1$. The error dynamics are described, for all $i = 2, \dots, N$, by

$$e_i^+ = f(z+e_i, \mathfrak{t}) - f(z, \mathfrak{t}) + B\sum_{j=1}^{N} (\ell_{ij} - \ell_{1j})\,\alpha(z+e_j, \mathfrak{t}) + \widetilde{w}_i. \tag{3.93}$$

Since by definition of the Laplacian entries $\sum_{j=1}^{N} \ell_{ij} = 0$ for any agent i, we can subtract

$B\sum_{j=1}^{N}(\ell_{ij}-\ell_{1j})\,\alpha(z,\mathfrak{t})=0$ from the right-hand side so that (3.93) becomes

$$e_i^+=\tilde{f}(e_i,\mathfrak{t})+B\sum_{j=2}^{N}\tilde{\ell}_{ij}\tilde{\alpha}(e_j,\mathfrak{t})+\widetilde{w}_i, \tag{3.94}$$

with

$$\tilde{\ell}_{ij}=\ell_{ij}-\ell_{1j},\quad \tilde{f}(e_i,\mathfrak{t})=f(z+e_i,\mathfrak{t})-f(z,\mathfrak{t}),\quad \tilde{\alpha}(e_j,\mathfrak{t})=\alpha(z+e_j,\mathfrak{t})-\alpha(z,\mathfrak{t}). \tag{3.95}$$

Overall, the closed-loop system can be written in compact form as

$$\mathbf{e}^+=\boldsymbol{\varphi}(\mathbf{e},\mathfrak{t})+\widetilde{\mathbf{w}}, \tag{3.96}$$

where we defined

$$\boldsymbol{\varphi}(\mathbf{e},\mathfrak{t}):=\begin{pmatrix}\tilde{f}(e_2,\mathfrak{t})+B\sum_{j=2}^{N}\tilde{\ell}_{2j}\tilde{\alpha}(e_j,\mathfrak{t})\\ \vdots\\ \tilde{f}(e_N,\mathfrak{t})+B\sum_{j=2}^{N}\tilde{\ell}_{Nj}\tilde{\alpha}(e_j,\mathfrak{t})\end{pmatrix}. \tag{3.97}$$

Now, select the following candidate Lyapunov function

$$V(\mathbf{e})=\mathbf{e}^\top(M\otimes P)\mathbf{e}, \tag{3.98}$$

where M comes from Lemma A.2. Note that, due to the properties of the Kronecker product, $M\otimes P$ is symmetric and positive-definite, being both M and P positive-definite matrices. Now, for each value of z and $\mathbf{e}=\mathrm{col}(e_2,\dots,e_N)$, define the scalar function $F_{\mathfrak{t}}:\mathbb{R}\to\mathbb{R}^{Nn_x}$ as

$$F_{\mathfrak{t}}(s):=\begin{pmatrix}\tilde{f}_s(s,e_2,\mathfrak{t})+B\sum_{j=2}^{N}\tilde{\ell}_{2j}\tilde{\alpha}_s(s,e_j,\mathfrak{t})\\ \vdots\\ \tilde{f}_s(s,e_N,\mathfrak{t})+B\sum_{j=2}^{N}\tilde{\ell}_{Nj}\tilde{\alpha}_s(s,e_j,\mathfrak{t})\end{pmatrix}, \tag{3.99}$$

with the definitions

$$\tilde{f}_s(s,e_i,\mathfrak{t}):=f(z+se_i,\mathfrak{t})-f(z,\mathfrak{t}), \tag{3.100a}$$
$$\tilde{\alpha}_s(s,e_j,\mathfrak{t}):=\alpha(z+se_j,\mathfrak{t})-\alpha(z,\mathfrak{t})=-\kappa YB^\top P\tilde{f}_s(s,e_i,\mathfrak{t}), \tag{3.100b}$$

where we used (3.17) and $Y:=\left(R+B^\top PB\right)^{-1}$. From (3.99)-(3.100) we have $F_{\mathfrak{t}}(0)=0$ and from (3.5) we get

$$\boldsymbol{\varphi}(\mathbf{e},\mathfrak{t})=F_{\mathfrak{t}}(1)=F_{\mathfrak{t}}(1)-F_{\mathfrak{t}}(0)=\int_0^1\partial F(s)ds\,\mathbf{e}, \tag{3.101}$$

with the following selections, obtained from (3.99) by proceeding as in (3.71),

$$\partial F(s):=[\mathrm{I}_{(N-1)n_x}-\kappa((L_{22}-\mathbf{1}L_{12})\otimes BYB^\top P)]\Psi(s), \tag{3.102}$$
$$\Psi(s):=\mathrm{diag}\left(\tilde{\psi}_2(s)\ \dots\ ,\ \tilde{\psi}_N(s)\right), \tag{3.103}$$
$$\tilde{\psi}_i(s)\in\mathcal{D}f,\qquad \forall i=2,\ \dots\ ,\ N. \tag{3.104}$$

Since $V(\mathbf{e}^+) = 2(\mathbf{e}^+)^\top (M \otimes P)\mathbf{e}^+ - V(\mathbf{e}^+)$, subtracting $\rho V(\mathbf{e})$ on both sides, we obtain by combining (3.96) and (3.101),

$$V(\mathbf{e}^+) - \rho V(\mathbf{e}) = 2\,(\mathbf{e}^+)^\top (M \otimes P) \int_0^1 \partial F(s)\, ds\, \mathbf{e} + 2(\mathbf{e}^+)^\top (M \otimes P)\widetilde{\mathbf{w}} - \Big[(\mathbf{e}^+)^\top (M \otimes P)\mathbf{e}^+ + \rho\, \mathbf{e}^\top (M \otimes P)\mathbf{e}\Big] \int_0^1 ds.$$

Then, by collecting everything under the integral and defining the extended error vector $\boldsymbol{\xi} = \mathrm{col}(\mathbf{e},\, \mathbf{e}^+)$ we obtain

$$V(\mathbf{e}^+) - \rho V(\mathbf{e}) = -\int_0^1 \boldsymbol{\xi}^\top \Upsilon(s) \boldsymbol{\xi}\, ds + 2(\mathbf{e}^+)^\top (M \otimes P)\widetilde{\mathbf{w}}, \tag{3.105a}$$

$$\Upsilon(s) := \begin{pmatrix} \rho(M \otimes P) & -\partial F^\top(s)(M \otimes P) \\ -(M \otimes P)\partial F(s) & (M \otimes P) \end{pmatrix}. \tag{3.105b}$$

Since $P \succ 0$, $M \succ 0$, and $M \otimes P$ is invertible, we can study the positive definiteness of $\Upsilon(s)$ via its Schur complement

$$\hat{\Upsilon}(s) = \rho(M \otimes P) - \partial F^\top(s)(M \otimes P)\partial F(s)\,.$$

By using the definition of $\partial F(s)$ in (3.102) and the properties of Kronecker products, we obtain

$$\hat{\Upsilon}(s) = \rho(M \otimes P) - \Psi(s)^\top (T_a + \mathrm{He}\,\{T_b\} + T_c)\Psi(s) \tag{3.106}$$

where we defined

$$\begin{aligned} T_a &:= M \otimes P, \\ T_b &:= -\kappa(M(L_{22} - \mathbf{1}L_{12}) \otimes PBYB^\top P), \\ T_c &:= \kappa^2[(L_{22} - \mathbf{1}L_{12})^\top M(L_{22} - \mathbf{1}L_{12}) \otimes PBYB^\top PBYB^\top P]. \end{aligned}$$

For T_b, by the properties of the Kronecker product, since $PBYB^\top P$ is symmetric we obtain

$$\mathrm{He}\,\{T_b\} = -\kappa(\mathrm{He}\,\{M(L_{22} - \mathbf{1}L_{12})\} \otimes PBYB^\top P).$$

Consequently, by Lemma A.2 and using again the properties of the Kronecker product, the following holds

$$\mathrm{He}\,\{T_b\} \preceq -2\kappa \underline{\mu}(M \otimes PBYB^\top P). \tag{3.107}$$

Similarly, since $R \succeq 0$, by exploiting the Kronecker product and by using again Lemma A.2, we get

$$T_c \;\preceq \kappa^2 \bar{\mu}^2 [M \otimes PBY(R + B^\top PB)YB^\top P] \preceq \kappa^2 \bar{\mu}^2 (M \otimes PBYB^\top P). \tag{3.108}$$

Using (3.107) and (3.108), matrix $\hat{\Upsilon}$ in (3.106) can be bounded as

$$\begin{aligned}\hat{\Upsilon}(s) &\succeq \rho(M \otimes P) - \Psi(s)^\top (M \otimes \overline{P})\Psi(s) \\ \overline{P} &= P + (\kappa^2 \bar{\mu}^2 - 2\kappa \underline{\mu}) PBYB^\top P\end{aligned} \tag{3.109}$$

Now, consider $\overline{P}$. By addition and subtraction, it can be rewritten as

$$\overline{P} = P - \sigma PBYB^\top P + (\kappa^2 \bar{\mu}^2 - 2\kappa \underline{\mu} + \sigma) PBYB^\top P.$$

Then, if $\kappa^2 \bar{\mu}^2 - 2\kappa \underline{\mu} + \sigma \leqslant 0$, namely if

$$\frac{\underline{\mu}}{\bar{\mu}^2}\left(1 - \sqrt{1 - \left(\frac{\bar{\mu}}{\underline{\mu}}\right)^2 \sigma}\right) \leqslant \kappa \leqslant \frac{\underline{\mu}}{\bar{\mu}^2}\left(1 + \sqrt{1 - \left(\frac{\bar{\mu}}{\underline{\mu}}\right)^2 \sigma}\right),$$

which holds due to the selection in (3.92), we obtain

$$\overline{P} \preceq P - \sigma PBYB^\top P = Q, \tag{3.110}$$

with Q defined in (3.16b). Using (A.6a) from Lemma A.2 and (3.110), $\hat{\Upsilon}(s)$ in (3.109) satisfies

$$\hat{\Upsilon}(s) \succeq \underline{m}\rho(\mathrm{I}_{N-1} \otimes P) - \overline{m}\Psi(s)^\top (\mathrm{I}_{N-1} \otimes Q)\Psi(s).$$

Recalling, from (3.103), the block-diagonal structure of $\Psi(s)$, and exploiting (3.16) we obtain

$$\begin{aligned}\hat{\Upsilon}(s) &\succeq \underline{m}\rho(\mathrm{I}_{N-1} \otimes P) - \overline{m}\,\mathrm{diag}(\{\tilde{\psi}_i(s)^\top Q \tilde{\psi}_i(s)\}_{i=2}^N) \\ &\succeq \underline{m}\rho(\mathrm{I}_{N-1} \otimes P) - \overline{m}\,\mathrm{diag}(\{\rho^2 P\}_{i=2}^N) = \underline{m}\rho(\mathrm{I}_{N-1} \otimes P) - \overline{m}\rho^2(\mathrm{I}_{N-1} \otimes P) \\ &\succeq \rho(\underline{m} - \overline{m}\,\rho)(\mathrm{I}_{N-1} \otimes P) \succeq \rho(\underline{m} - \overline{m}\,\rho_M)(\mathrm{I}_{N-1} \otimes P) \succeq 0,\end{aligned}$$

where we used $0 < \rho \leqslant \rho_M = \underline{m}\overline{m}^{-1}$. Since $\hat{\Upsilon}(s) \succeq 0$ for each $s \in [0,1]$, we conclude that also $\Upsilon(s)$ defined in (3.105b) satisfies $\Upsilon(s) \succeq 0$ for all $s \in [0,1]$, and (3.105a) implies

$$V(\mathbf{e}^+) - \rho V(\mathbf{e}) \leqslant 2(\mathbf{e}^+)^\top (M \otimes P)\widetilde{\mathbf{w}} \tag{3.111}$$

By the generalized Young's inequality and by considering the factorization of the positive matrix $M \otimes P = \sqrt{M \otimes P}^\top \sqrt{M \otimes P} = (\sqrt{M \otimes P})^2$ (with $\sqrt{M \otimes P}$ denoting the unique positive square root of $M \otimes P \succ 0$), we have

$$\begin{aligned}2(\mathbf{e}^+)^\top (M \otimes P)\widetilde{\mathbf{w}} &= 2(\mathbf{e}^+)^\top (\sqrt{M \otimes P})^2 \widetilde{\mathbf{w}} \\ &\leqslant (1 - \sqrt{\rho})V(\mathbf{e}^+) + \frac{1}{1 - \sqrt{\rho}}\widetilde{\mathbf{w}}^\top (M \otimes P)\widetilde{\mathbf{w}}.\end{aligned}$$

Then, since $\rho \in (0,1)$, inequality (3.111) implies

$$V(\mathbf{e}^+) - \sqrt{\rho}V(\mathbf{e}) \leqslant \frac{1}{\sqrt{\rho}(1-\sqrt{\rho})}\widetilde{\mathbf{w}}^\top (M \otimes P)\widetilde{\mathbf{w}},$$

thus proving exponential ISS properties of the $\mathbf{e}$ dynamics due to the quadratic form of (3.98). Finally, similarly to the linear scenario of Theorem 3.4, relations (3.73) hold and robust synchronization as in Problem 3.1 is obtained, thus concluding the proof. □

Remark 3.9. *Note that the contraction inequality* (3.16a) *in the context of Theorem 3.6 is tightly related to the structure of the Laplacian matrix and its eigenvalues. To appreciate this link, consider the network graph to be undirected and leader-connected. Under these conditions, the Laplacian* L *in* (A.5) *satisfies* $L_{11} = 0$, $L_{12} = \mathbf{0}$. *Then,* $L_{22} = L_{22}^\top \succ 0$ *and we can select* $M = \mathrm{I}_{N-1}$ *and* $\underline{\mu}$ *(resp.* $\bar{\mu}$*) in Lemma A.2 as the smallest (resp. largest) eigenvalue of* L_{22}. *Consequently, the admissible values of* σ *are related to the condition number of* L_{22}. *Moreover, by picking* $M = \mathrm{I}_{N-1}$, *the contraction rate* ρ *disentangles from the network structure, as* $\underline{m} = \overline{m} = 1$. *Indeed, from* (A.6a) *we can select* $\rho_M = 1$ *and, consequently, condition* (3.91) *in Theorem 3.6 imposes no constraints on* $\rho \in (0,1)$*, as in Proposition 3.2.*

Numerical example. In what follows, we propose a simple numerical example to validate the presented results. Consider a network of $N = 6$ agents connected according to the weighted, directed graph in Figure 3.4 and evolving according to the Lur'e dynamics

$$x_i^+ = Ax_i + F\phi(Cx_i) + Bu_i, \quad i = 1, \dots, 6$$

where

$$A = \begin{pmatrix} 1.1 & 0.1 \\ -0.3 & 0.5 \end{pmatrix}, \; B = \begin{pmatrix} 2 \\ 0.3 \end{pmatrix}, \; C = \begin{pmatrix} 1 & -1 \end{pmatrix}, \; F = \begin{pmatrix} -0.1 \\ 0.7 \end{pmatrix},$$

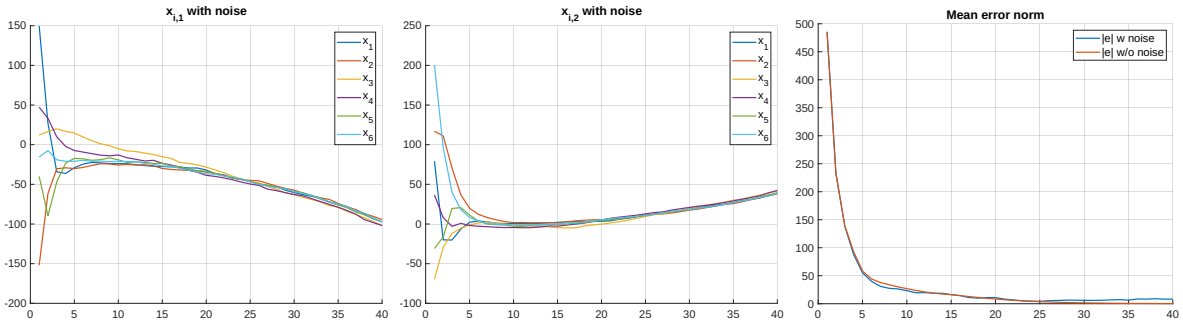
and $\phi(\cdot) = \mathrm{sat}_{10}(\cdot) = \max(\min(\cdot, 10), -10)$. It is simple to verify that $J_\phi \in \{0, 1\}$ and the Laplacian matrix is

$$L = \begin{pmatrix} 3 & -1 & 0 & 0 & -2 & 0 \\ -1 & 2 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & -3 & -1 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}.$$

Conditions (3.34), (3.35) and (3.25b) are easily solved with $\rho = 0.9$, $\sigma = 0.285$, $S = \Omega = 1$ and provide

$$W = \begin{pmatrix} 0.0408 & -0.1747 \\ -0.1747 & 1.1273 \end{pmatrix}, \qquad \Sigma = \begin{pmatrix} 0.0388 & -0.1294 \\ -0.1294 & 0.8495 \end{pmatrix}.$$

Then, we select a control law of the form (3.59) with α as in (3.17), $R = 1$ and $\kappa = 0.2$. It is interesting to notice that, by solving (A.6) via semi-definite programming, ρ, σ and κ fall outside the required bounds. This shows the conservativeness of our Lyapunov analysis, that is aimed at obtaining a very general result. Finally, we simulate the proposed closed-loop under the action of a random Gaussian noise $w \in \mathcal{N}(0, 0.5)$. Robust exponential convergence to a non-trivial trajectory with initial conditions sampled from a normal distribution $\mathcal{N}(0, 100)$ are presented in Figure 3.2 and Figure 3.3. As expected, Figure 3.3 shows an exponential decrease of the average error between the agents, which converges to a bounded value in the presence of additive noise.



(a) (b) (c)

Figure 3.2: Trajectories during transient. a-b) state components with noise. c) mean error wrt agent 1 with and without noise.

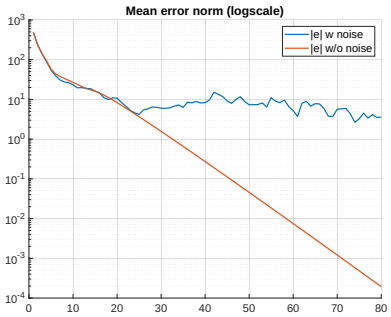


Figure 3.3: Long-term mean error wrt agent 1 in logarithmic scale.

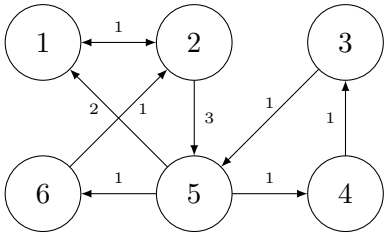


Figure 3.4: Communication graph.

4 Learning contractive controllers

In this chapter, we will concentrate on the classical notion of contraction and explore the integration of machine learning techniques in the design loop of controllers based on contraction analysis. In particular, we will demonstrate how recent numerical approaches rooted in deep learning can address the challenges of finding explicit solutions to feedback designs that rely on solving partial differential inequalities (PDIs). The framework of contraction analysis saw the development of different conditions for the verification of related properties. In what follows, we focus on the so-called metric-based conditions, thus paralleling the results proposed in Chapter 3. However, the continuous-time framework already witnessed the extension of Euclidean metric-based conditions (also known as Demidovich conditions [57, 163]) to the non-Euclidean and Riemannian settings [6,37]. This extension allowed the study of design procedures for time-varying, input-affine nonlinear systems with nonlinear input vector field, e.g. [86, 140]. Therefore, we will focus our attention on the more general approaches based on Riemannian metric conditions, that extend the Riccati equation and the so-called L_gV conditions [171].

The chapter begins by reviewing the results for contractive feedback design with infinite-gain margin properties. The proposed solution shares conceptual similarities with Control Contraction Metrics (CCMs), but differs in terms of the control action derivation [140]. Specifically, in our approach, the control action is obtained without the need for path integration along the geodesic, thus differing from CCMs-based methods (see [140, equation (6)]). Then, paralleling Chapter 3, our attention turns to the multi-agent synchronization problem, and we demonstrate how Riemannian metric-based approaches can be leveraged to address it. Finally, we leverage the aforementioned results to tackle the robust regulation problem.

Since obtaining an analytical solution for the proposed controller design is often challenging, we rely on DNNs for practical controller derivation. We propose an algorithm for learning a synchronizing control law based on a Physics-Informed Neural Network (P-INN) approximating the solution of a static PDI, which conceptually extends the stabilizability Riccati-like algebraic inequality (see, e.g., [126, Section II.C]). It has to be noted that the use of DNNs for approximating contractive controllers recently attracted attention in the control community [58]. There are two primary motivations for employing modern optimization methods such as deep learning in the design of controllers based on Riemannian metrics. Firstly, these designs often involve solving PDIs, for which explicit solutions are hard to obtain. Deep learning offers well-established methods for approximately solving partial differential equations, with P-INNs being

a notable example [65, 120, 173, 177]. Secondly, recent Riemannian metric-based controllers exhibit infinite-gain margin properties [84, 85]. In simple terms, this implies that scaling the control action by any scalar gain greater than 1 still yields a stabilizing effect. As mentioned in Section 3.2, this property is associated with some form of robustness, making this class of controllers particularly appealing for guaranteed DNN-based feedback design.

Several related works have explored similar approaches [58, 209, 219, 228, 245]. In [219], the authors propose a convex optimization problem to compute a suitable metric and subsequently approximate the solution using a DNN. The convex optimization is solved on a finite number of points and the DNN provides a continuous interpolation. However, the learning techniques in [209, 228] are more aligned with our approach. In [209] the authors propose learning CCMs to solve output tracking problems. Similar to our scenario, both [228] and [209] aim at minimizing a loss function defined by the matrix conditions required for contraction. Once such a function reaches zero, the DNN provides the entries of a suitable metric for each point in the training/test datasets. In [228], the authors propose learning the metric via a Siamese DNN structure [193]. Although related, our solution differs on several fundamental aspects. Firstly, when considering the approximation error induced by the learning procedure, the results in [209] only offer probabilistic convergence guarantees to trajectories close to the reference one. Furthermore, the learned controller in [209] needs to adhere to specific structural constraints to ensure assumptions are always satisfied. In contrast, our approach allows for greater expressivity of the selected approximator, without imposing constraints on the controller structure. Additionally, our sign definite cost relies on eigenvalues, ensuring sign-definiteness for all points in the dataset, unlike the random sampling approach in [209]. Both [209] and [228] rely on CCMs. Hence, while the proposed loss functions are similar in the sense they involve sign definiteness-related costs, their components are different. Moreover, we introduce a separate estimator that optimizes the cost function parameters and collaborates with the DNN during the optimization process, resulting in more adaptable and flexible constraints. Furthermore, our solution offers asymptotic convergence guarantees even when assumptions are only approximately satisfied (see Theorem 4.4 and Theorem 4.5). Instead of computing the control law through approximate integration along the geodesic, we rely on these relaxed assumptions. This simplifies the algorithm significantly, as finding geodesics is generally a computationally demanding task that often involves solving an online optimization problem in each point, with no guarantee of finding the exact solution due to optimization being performed only in compact sets. In contrast, our design derives the structure of the control action analytically, and its implementation is obtained through offline optimization. Lastly, unlike [228], we operate in the continuous-time framework, avoiding the need for a Siamese network by computing the DNN Jacobian. Common libraries like PyTorch [162] provide automatic differentiation tools that facilitate obtaining the Jacobian.

4.1 Preliminaries on Riemannian metric conditions for feedback design

Since we will base our analysis on Riemannian metrics, we start by recalling some fundamental results. Given a C^1 vector field $f : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_x}$ and a C^1 2-tensor $P : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_x \times n_x}$,

we indicate with $\mathcal{L}_f P(x,t)$ the Lie derivative of the tensor P along f defined as

$$\mathcal{L}_f P(x,t) := \mathfrak{d}_f P(x,t) + P(x,t)\tfrac{\partial f}{\partial x}(x,t) + \tfrac{\partial f}{\partial x}(x,t)^\top P(x,t)\,,$$
$$\mathfrak{d}_f P(x,t) := \lim_{h\to 0} \frac{P(X(x,t+h,t),t) - P(x,t)}{h} + \frac{\partial P}{\partial t}(x,t)\,,$$

where $X(x,t_0+t,t_0)$ is the solution of the initial value problem

$$\tfrac{\partial}{\partial t} X(x,t_0+t,t_0) = f(X(x,t_0+t,t_0)), \quad X(x,t_0,t_0) = x,$$

for all $t \geqslant 0$. Note that $\mathcal{L}_f P(x,t)$ can be equivalently expressed as

$$\mathcal{L}_f P(x,t) = \lim_{h\to 0} \frac{(I + h\frac{\partial f}{\partial x}(x,t))^\top P(x+hf(x,t),t+h)(I + h\frac{\partial f}{\partial x}(x,t)) - P(x,t)}{h}\,,$$

with coordinates

$$(\mathcal{L}_f P(x,t))_{i,j} = \sum_k \left[2P_{ik}(x,t)\frac{\partial f_k}{\partial x_j}(x,t) + \frac{\partial P_{ij}}{\partial x_k}(x,t) f_k(x,t) + \frac{\partial P_{ij}}{\partial t}(x,t) \right],$$

being P_{ij} (resp f_i) the component of $P(x,t)$ (resp. f) at coordinates (i,j) (resp. i). Given any two elements $x_1, x_2 \in \mathbb{R}^{n_x}$, let $\Phi_t : [0,1] \to \mathbb{R}^{n_x}$ be any C^1 path (parametrized by t) such that at time t it satisfies $\Phi_t(0) = x_1$ and $\Phi_t(1) = x_2$. We define the length of the curve Φ_t in the operator norm $P(x,t)$ as

$$\ell(\Phi_t) := \int_0^1 \sqrt{\frac{d\Phi_t}{ds}(s)^\top P(\Phi_t(s),t)\frac{d\Phi_t}{ds}(s)}\; ds\,. \tag{4.1}$$

The Riemannian distance between x_1 and x_2 is then defined as the infimum of the length among all the possible piecewise C^1 paths Φ, namely

$$d_P(x_1,x_2) := \inf_{\Phi_t} \left\{ \ell_P(\Phi_t) \right\}\,.$$

For more details on Riemannian analysis, we refer to [181, 212] and references therein.

4.1.1 Riemannian metric conditions for incremental properties

We now move to the presentation of stability results based on Riemannian metric conditions. As a foundation, we start with the introduction of incremental properties in the autonomous framework, thus creating a clear link with the results of Chapter 5. Then, we move to the non-autonomous setting, and consider input-affine nonlinear systems.

Riemannian metric conditions for δGES. We start by considering autonomous nonlinear systems of the form

$$\dot{x} = f(x,t) \tag{4.2}$$

where $f : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_x}$ is a C^2 vector field in the first argument and piece-wise continuous in the second. We denote by $X(x,t,t_0)$ the solution of system (4.2) with initial condition x evaluated at time $t \geqslant t_0$, and we assume existence and uniqueness of trajectories. We define the notion of incremental stability as follows.

Definition 4.1 (Incremental exponential stability)**.** *System* (4.2) *is incrementally globally exponentially stable (δGES) if there exist two strictly positive real numbers $\lambda, k > 0$ such that*

$$|X(x_1, t, t_0) - X(x_2, t, t_0)| \leqslant k|x_1 - x_2|e^{-\lambda(t-t_0)} \tag{4.3}$$

for any couple of initial conditions $(x_1, x_2) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_x}$ and for all $t \geqslant t_0$.

Riemannian metric conditions can be used to show that shrinking lengths of the form (4.1) implies convergence of trajectories as per Definition 4.1. Following the metric approach [6, 87, 134], a dynamical system of the form (4.2) is δGES if there exists a Riemannian metric for which the mapping $t \mapsto X(x, t, t_0)$ is a contracting mapping. This concept is formally expressed by the following theorem.

Theorem 4.1. *Consider system* (4.2) *and suppose there exist three real numbers $\overline{p}, \underline{p}, \lambda > 0$ and a C^1 matrix function $P : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_x \times n_x}$ taking symmetric positive definite values such that the following holds*

$$\underline{p}I \preceq P(x, t) \preceq \overline{p}I\,, \tag{4.4a}$$

$$\mathcal{L}_f P(x, t) \preceq -2\lambda P(x, t)\,, \tag{4.4b}$$

for all $x \in \mathbb{R}^{n_x}$ and for all $t \geqslant t_0$. Then the system (4.2) *is δGES.*

A proof can be found in [5, 83, 134]. A converse theorem can be found in [5, Proposition IV] under the assumption of f being a globally Lipschitz vector field. Note that the lower bound in (4.4a) is required to make sure that the whole $\mathbb{R}^{n_x}$ space endowed with the Riemannian metric P is complete. Such a condition guarantees that every geodesic (i.e. the shortest curve between (x_1, x_2)) can be maximally extended to $\mathbb{R}$, see e.g. [181]. By Hopf-Rinow's Theorem (see [212, Theorem 1.1]) this implies that the metric is complete and hence that the minimum of the length of any curve connecting two point (x_1, x_2) is actually given by the length of the geodesic at any time instant. Similarly, it guarantees that the Lyapunov function defined as the distance associated to the norm operator $|x|_P := x^\top P(x, t)x$ is radially unbounded, and therefore incremental properties are obtained globally in the state space. On the other hand, the upper bound in (4.4a) is introduced for solutions to be uniformly decreasing with respect to time and to correlate the Riemaniann distance in P to the Euclidean one in (4.3).

Riemannian metric conditions for δISS. We now move to input-affine systems, thus studying the incremental input-to-state (δISS) properties of a system of the form

$$\dot{x} = \varphi(x, t) + g(x, t)w \tag{4.5}$$

where w is an exogenous signal taking values in a compact set $\mathcal{W} \subset \mathbb{R}^{n_w}$, $\varphi : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_x}$ and $g : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_x \times n_w}$ are C^2 functions. Hence, we present a continuous-time counterpart of the δISS property proposed in Chapter 3. We denote by $X(x, w, t, t_0)$ the solution of system (4.5) starting at initial condition x at time t_0 with input $w = w(t)$ and satisfying the initial value problem

$$X(x, w, t_0, t_0) = x, \qquad \frac{\partial X}{\partial t}(x, w, t, t_0) = f(X(x, w, t, t_0), t) + g(X(x, w, t, t_0), t)w. \tag{4.6}$$

We state the continuous-time counterpart of Definition 3.3 in the following definition.

Definition 4.2 (Incremental exponential Input-to-State Stability)**.** *System* (4.5) *is incrementally exponentially input-to-state stable (δISS) if there exist positive real numbers $k, \lambda, \gamma > 0$ such that*

$$|X(x_1, w_1, t, t_0) - X(x_2, w_2, t, t_0)| \leqslant ke^{-\lambda(t-t_0)}|x_1 - x_2| + \gamma \sup_{s\in[t_0,t)} |w_1(s) - w_2(s)| \tag{4.7}$$

for all initial conditions $x_1, x_2 \in \mathbb{R}^{n_x}$ and for all inputs w_1, w_2 taking values in $\mathcal{W} \subset \mathbb{R}^{n_w}$, for all $t \geqslant t_0$.

Similar to the result of Theorem 4.1, we present metric-based sufficient conditions to establish incremental exponential ISS properties. To this aim, we first introduce the notion of Killing vector field[1].

Definition 4.3 (Killing vector field)**.** *Given a C^1 2-tensor $P : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_x \times n_x}$ and a C^1 matrix function $g : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_x \times n_w}$, we say that g is a Killing vector field with respect to P if*

$$\mathcal{L}_{g_\iota} P(x, t) = 0, \qquad \iota = 1, \dots, n_w, \qquad \forall (x, t) \in \mathbb{R}^{n_x} \times \mathbb{R}, \tag{4.8}$$

with g_ι being the ι-th column of g.

In what follows, we will use the notation $\mathcal{L}_g P(x, t) = 0$ to indicate (4.8), with a slight abuse of notation. The Killing Vector property implies that distances between different trajectories generated by the vector field $g(x, t)$ in the norm $|\cdot|_P$ are invariant. Roughly speaking, the signals that enter in the directions of the vector field g do not affect the distances, in the sense that the distance (associated with the norm provided by P) between different trajectories of the differential equation $\dot{x} = g(x, t)$ is constant for any $t \geqslant t_0$. It is easy to verify that the Killing vector property is always satisfied between two constant matrices $P(x) = P$ and $g(x) = B$. Similarly, there always exists a matrix valued function $P : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x \times n_x}$ satisfying the Killing vector property for the case of scalar, time-invariant input vector field $g : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$, see [83, Remark 1.3.2]. Based on the previous notion of Killing vector, we have the following result.

Theorem 4.2. *Consider system* (4.5) *and suppose that g is a bounded vector field, namely there exists a real number $\bar{g} > 0$ such that $|g(x, t)| \leqslant \bar{g}$ for all $(x, t) \in \mathbb{R}^{n_x} \times \mathbb{R}$. If there exists a C^1 matrix function $P : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_x \times n_x}$ taking symmetric positive definite values and three real numbers $\overline{p}, \underline{p}, \lambda > 0$ satisfying*

$$\underline{p} I \preceq P(x, t) \preceq \overline{p} I, \tag{4.9a}$$

$$\mathcal{L}_f P(x, t) \preceq -2\lambda P(x, t), \tag{4.9b}$$

$$\mathcal{L}_g P(x, t) = 0, \tag{4.9c}$$

for all $x \in \mathbb{R}^{n_x}$ and for all $t \geqslant t_0$, then system (4.5) *is δISS.*

A proof can be found in [83,85]. Differently to some definitions from the literature, e.g. [9], the signal w is required to live in bounded compact sets. This is needed to derive the δISS condition (4.7) from the metric-based conditions (4.9). However, we emphasize that the parameter γ doesn't depend on the compact set $\mathcal{W}$. Furthermore, we restrict the definition to the case of exponential convergence. This is due to the fact that we look for metric-based conditions to obtain the δISS property (4.7). In the more general definition considered in [9], any class-$\mathcal{KL}$

[1]The name "Killing vector field" takes the name after Wilhelm Killing, a German mathematician.

function can be selected. For instance, the previous result can be easily extended to the case in which the system (4.5) is defined as

$$\dot{x} = f(x,t) + g(x,t)\rho(w)$$

with ρ being any function for which there exists a class-$\mathcal{K}_\infty$ function δ_ρ such that

$$|\rho(w_1) - \rho(w_2)| \leqslant \delta_\rho(|w_1 - w_2|).$$

In such a case, γ in (4.7) becomes a class-$\mathcal{K}$ function, [83, Section 1.3].

4.1.2 Design of a contractive infinite-gain margin feedback

Building on the above results, we now focus on the design of controllers making the closed loop incrementally stable via contraction. Consider now a nonlinear system of the form

$$\begin{aligned} \dot{x} &= f(x,t) + g(x,t)u \\ y &= h(x,t) \end{aligned} \tag{4.10}$$

with state $x \in \mathbb{R}^{n_x}$, control input $u \in \mathbb{R}^{n_u}$ and measured output $y \in \mathbb{R}^{n_y}$. The design of infinite-gain margin laws in the context of input-affine nonlinear systems of the form (4.10) has been investigated in the context of control Lyapunov function [191, Chapter 3] and arises quite naturally in the context of feedback design for passive systems, e.g. [131]. Here, we study an extension of the linear case in the context of contractive feedback laws. In particular, we state the following definition.

Definition 4.4 (Contractive feedback with infinite-gain margin)**.** *Consider system* (4.10)*. We say that the C^1 function $\psi : \mathbb{R}^{n_y} \times \mathbb{R} \to \mathbb{R}^{n_u}$ is a contractive control law with infinite gain margin for system* (4.10) *if there exist a C^1 matrix function $P : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_x \times n_x}$, taking symmetric positive definite values and three real numbers $\overline{p}, \underline{p}, \lambda > 0$ such that the following holds*

$$\underline{p} I \preceq P(x,t) \preceq \bar{p} I \tag{4.11a}$$

$$\mathcal{L}_{\varphi_\kappa} P(x,t) \preceq -2\lambda P(x,t), \tag{4.11b}$$

for all $x \in \mathbb{R}^{n_x}$, all $t \geqslant 0$ and all $\kappa \geqslant 1$, where we defined the k-parametrized closed-loop

$$\varphi_\kappa(x,t) := f(x,t) + g(x,t)\kappa\,\psi(h(x,t))\,.$$

Results for multiple scenarios (e.g., output feedback and observer forms) can be found in [83]. In what follows, we recall the ones for full-state feedback design $h(x,t) = x$, which will be the main focus of the following sections. For systems in *state-feedback form*

$$\begin{aligned} \dot{x} &= f(x,t) + g(x,t)(u+w) \\ y &= x \end{aligned} \tag{4.12}$$

with state $x \in \mathbb{R}^{n_x}$ control input $u \in \mathbb{R}^{n_u}$ and disturbance $w \in \mathcal{W} \subset \mathbb{R}^{n_u}$ satisfying the matching condition [172, 175]. We have the following result.

Proposition 4.1. *Consider system* (4.12)*, and suppose there exist a* C^1 *matrix function* $P : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_x} \times \mathbb{R}^{n_x}$ *taking positive definite symmetric values, a function* $\alpha : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_u}$ *and real numbers* $\underline{p}, \bar{p}, \lambda > 0$ *such that the following hold*

$$\underline{p} I \preceq P(x,t) \preceq \bar{p} I \,, \tag{4.13a}$$

$$\mathcal{L}_f P(x,t) - P(x,t) g(x,t) R^{-1} g(x,t)^\top P(x,t) \preceq -2\lambda P(x,t) \,, \tag{4.13b}$$

$$\mathcal{L}_g P(x,t) = 0 \,, \tag{4.13c}$$

$$\frac{\partial \alpha}{\partial x}(x,t)^\top = P(x,t) g(x,t) \,, \tag{4.13d}$$

for all $x \in \mathbb{R}^{n_x}$*, all* $t \geqslant 0$ *and for some positive definite symmetric matrix* $R \in \mathbb{R}^{n_u \times n_u}$*. Then, the feedback law* $\psi(x,t) = -\frac{1}{2} R^{-1} \alpha(x,t)$ *is a contractive control law with infinite-gain margin for system* (4.12)*. Moreover, the closed-loop system is* δ*ISS with respect to* w*.*

A proof can be obtained by adapting the one presented in [85, Theorem 3], see [85, Remark 1]. Condition (4.13b) can be seen as a Riccati-like inequality where P is a matrix function. For linear systems of the form $\dot{x} = Ax + Bu$, condition (4.13b) boils down to the well-known algebraic Riccati inequality (ARI) $PA + A^\top P - PBR^{-1}B^\top P \preceq -2\lambda P$ which admits a solution under the mild assumption that (A, B) is stabilizable. In this case, a stabilizing control action is given by $u = -\kappa R^{-1} B^\top P$ for any $\kappa \geqslant \frac{1}{2}$. Furthermore, we remark that such a design possesses the infinite-gain margin property [84]. Such a property will play a fundamental role in the following section on multi-agent synchronization, as it will provide a synchronizing control law that is "robust" with respect to the graph topology, see [102, Section 5], thus paralleling the results in Section 3.2. Notice also that condition (4.13d) is asking for some integrability property of the function Pg. Moreover, it is interesting to notice that it is a condition acting on the Jacobian of the control law α, which can theoretically be shifted by any function of time without loss of the contractivity properties of the closed-loop. This reflects the fact that stability is imposed in the incremental framework, thus allowing convergence to non-trivial trajectories. Such a fact turns out to be extremely useful in the context of multi-agent synchronization and output tracking of the following sections. As a final remark, we highlight that for linear systems both (4.13d) and (4.13c) are always satisfied. Indeed P is Euclidean (and thus constant) and so it is $g(x,t) = B$. This implies that the Killing vector assumption (4.13c) holds and the function α in (4.13d) is $\alpha(x,t) = B^\top P x$.

4.2 Learning synchronizing controllers for nonlinear systems

In light of the results from Section 4.1, our objective is to leverage these findings to design synchronizing controllers for multi-agent systems. As discussed in Chapter 3, synchronization or consensus refers to the problem of a group of agents trying to achieve agreement. For the synchronization of continuous-time linear systems, fundamental results were obtained in [126, 183] and a comprehensive discussion can be found in [102, Section 5]. For nonlinear systems, most results exploit existing techniques developed for single-agent systems, specifically adapted to deal with a distributed framework, e.g., [13,39,47,104,205]. A popular approach to solve the synchronization problem consists in exploiting tools derived from contraction and incremental stability theory (see [5, 9, 57, 72]). Based on this framework, most of the results considered quadratic Lyapunov functions or, equivalently [57], Euclidean metrics, e.g., [8, 14, 51, 164, 226, 240, 243]. However, only few investigated the use of nonlinear metrics, e.g., [4, 105].

In this section, we address the synchronization problem for homogeneous networks (i.e., networks where the agents' dynamics are identical) of continuous-time input-affine nonlinear systems using a distributed control feedback, i.e., diffusive coupling. The fundamental means to achieve such a result is the use of control laws based on metric conditions, thus paralleling Chapter 3. Since the input vector field is assumed to be a nonlinear function of the state, we will exploit nonlinear Riemmanian metrics. Hence, we aim to design a static distributed state-feedback controller based on the conditions presented in Section 4.1. Our focus is on undirected and leader-connected graphs, where a designated leader agent can only transmit information to other nodes, while bidirectional communication exists among the remaining nodes, see Appendix A.2.

Our approach is based on the solution of a PDI, which is very complex to verify analytically. Therefore, to deal constructively with such a complexity, we show that synchronization can be achieved in a regional context under less stringent assumptions. Thanks to this relaxation, we show how machine learning techniques can aid in overcoming such a complexity without loosing important convergence guarantees. Hence, we provide a formulation of a practical algorithm based on DNNs to check the solvability of such a PDI. To show the potential of this approach, we consider the problem of synchronizing a network of Lorentz oscillators in which the input gain is highly nonlinear (and for which existing techniques cannot be easily applied).

4.2.1 A relaxed nonlinear metric-based solution

For an introduction about graph theoretic fundamental concepts we refer to Appendix A.2. We now present the leader-synchronization problem, which can be interpreted as a specialized version of Problem 3.1 in continuous-time. Consider a network of N agents. Without loss of generality, we assume the leader node to be node 1. The agents dynamics are described by

$$\begin{aligned} \dot{x}_1 &= f(x_1, t) \\ \dot{x}_i &= f(x_i, t) + g(x_i, t)u_i, \qquad i = 2, \dots, N, \end{aligned} \tag{4.14}$$

where $x_1 \in \mathbb{R}^{n_x}$ is the state of the leader, $x_i \in \mathbb{R}^{n_x}$ is the state of node i and $u_i \in \mathbb{R}^{n_u}$ is the control action on node i, for all $i = 2, \dots, N$. We suppose that f, g are C^2 functions in their first argument and piece-wise continuous in the second. We denote the state of the entire network as

$$\mathbf{x} := \mathrm{col}(x_1, \dots, x_N) \in \mathbb{R}^{Nn_x}. \tag{4.15}$$

Furthermore, we denote with $X_i(x_i^\circ, t, t_0)$ the trajectory of agent i evaluated at time $t \geqslant t_0$ such that $X_i(x_i^\circ, t_0, t_0) = x_i^\circ$, and with $\mathbf{X}(\mathbf{x}_0, t, t_0)$ the trajectory of the entire network (4.15) evaluated at time $t \geqslant t_0$, with initial condition $\mathbf{x}_0 \in \mathbb{R}^{Nn_x}$ at initial time $t_0 \in \mathbb{R}$. Similar to Section 3.2.1, our synchronization objective is to design a nonlinear diffusive coupling, namely a distributed feedback control law of the form

$$u_i = -\sum_{j=1}^{N} \ell_{ij} \beta(x_j, t) \tag{4.16}$$

for all $i = 2, \dots, N$, for some C^1 function $\beta : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_u}$, that stabilizes the dynamics (4.14) on the so-called *leader-synchronization manifold* $\mathcal{M}$ defined as

$$\mathcal{M} := \{\mathbf{x} \in \mathbb{R}^{Nn_x} \mid x_i = x_1, \ \text{ for all } i \in \{1, \dots, N\}\}, \tag{4.17}$$

where the states of all the agents of the network agree with the leader. By construction, the i-th agent uses only the state information x_j of its neighborhoods $j \in \mathcal{N}_i$ and its own local

information x_i. Furthermore, the control action u_i is equal to zero on the synchronization manifold. In other words, when consensus is achieved, no correction term is needed for each individual agent. As a consequence, stabilizing all the agents on a desired equilibrium point is generally not a valid solution in such a framework. We formalize our synchronization problem as follows.

Problem 4.1 (Leader synchronization)**.** *The distributed feedback control law* (4.16) *solves the leader-synchronization problem for the network* (4.14) *if the manifold $\mathcal{M}$ defined in* (4.17) *is globally uniformly exponentially stable for the closed-loop dynamics*

$$\begin{aligned} \dot{x}_1 &= f(x_1, t) \\ \dot{x}_i &= f(x_i, t) - g(x_i, t) \sum_{j=1}^{N} \ell_{ij} \beta(x_j, t), \quad i = 2, \dots, N, \end{aligned} \tag{4.18}$$

namely, there exist positive constants k and $\lambda > 0$ such that for all $(\mathbf{x}_0, t_0)$ in $\mathbb{R}^{Nn_x} \times \mathbb{R}$ solutions of (4.18) *are defined for all $t \geqslant t_0$ and*

$$|\mathbf{X}(\mathbf{x}_0, t, t_0)|_{\mathcal{M}} \leqslant k e^{-\lambda (t - t_0)} |\mathbf{x}_0|_{\mathcal{M}}, \quad \forall t \geqslant t_0. \tag{4.19}$$

Then, in order to solve our leader synchronization problem, we assume existence of solutions and leader-connected graphs. Moreover, we suppose that the pair f, g satisfies the following controllability, Killing vector and integrability assumptions.

Assumption 4.1. *The graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ is undirected and leader-connected. Moreover, for each (x_1°, t_0) in $\mathbb{R}^{n_x} \times \mathbb{R}$ the trajectory of* (4.14) *exists for all $t \geqslant t_0$.*

Assumption 4.2. *There exist a C^1 matrix function $P : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_x \times n_x}$ taking symmetric positive definite values and positive real numbers $\underline{p}, \bar{p}, \rho, \lambda > 0$ such that the following holds for all $(x, t) \in \mathbb{R}^{n_x} \times \mathbb{R}$*

$$\begin{aligned} &\mathcal{L}_f P(x,t) - \rho P(x,t) g(x,t) g(x,t)^\top P(x,t) \preceq -2\lambda P(x,t)\,, \\ &\underline{p} I \preceq P(x,t) \preceq \bar{p} I\,. \end{aligned} \tag{4.20}$$

Assumption 4.3. *The matrix function g has the Killing vector field property with respect to P, namely*

$$\mathcal{L}_g P(x,t) = 0\,, \quad \forall (x,t) \in \mathbb{R}^{n_x} \times \mathbb{R}\,. \tag{4.21}$$

Assumption 4.4. *The vector field Pg satisfies an integrability condition. Specifically, denoting $g = [g_1 \dots g_{n_u}]$, there exists a C^2 function $\alpha = (\alpha_1, \dots, \alpha_{n_u})$, $\alpha_\iota : \mathbb{R}^{n_x} \times \mathbb{R} \mapsto \mathbb{R}$ for $\iota = 1, \dots, n_u$, satisfying*

$$\frac{\partial \alpha_\iota}{\partial x}(x,t) = g_\iota(x,t)^\top P(x,t)\,, \quad \forall (x,t) \in \mathbb{R}^{n_x} \times \mathbb{R}\,. \tag{4.22}$$

We highlight that Assumption 4.2 and Assumption 4.3 recover the design proposed in [140, Section III.A]. We are now ready to show the first result of this section, exploiting contraction analysis for the design of synchronizing controllers.

Theorem 4.3. *Consider a network $G = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ of agents* (4.14) *and let Assumptions 4.1 to 4.4 hold. Then, for any $\kappa \geqslant \frac{\rho}{2\underline{\mu}}$, with $\underline{\mu}$ given by Lemma A.3, the distributed state-feedback control law* (4.16) *with*

$$\beta(x_i, t) = \kappa\, \alpha(x_j, t)\,, \tag{4.23}$$

and α satisfying (4.22)*, solves Problem 4.1.*

Proof. The main goal is to show that the norm of the difference between any agent x_i and the leader x_1 exponentially decreases to zero. Therefore, let us consider the following change of coordinates

$$x_i \mapsto \tilde{x}_i := x_i - x_1, \qquad i = 2, \dots, N$$

and let us collect all the vectors $\tilde{x}_i$ as $\tilde{x} := \mathrm{col}(\tilde{x}_2, \dots, \tilde{x}_N)$ and define $z = x_1$. Since $\ell_{ij} = 0$ for all $j \notin \mathcal{N}_i$, the dynamics of the error $\tilde{x}_i$ for all $i = 2, \dots, N$ under the control law (4.16), (4.23) can be rewritten as

$$\dot{\tilde{x}}_i = f(z + \tilde{x}_i, t) - f(z, t) - \kappa g(z + \tilde{x}_i, t) \left[\sum_{j=2}^{N} \ell_{ij} \alpha(z + \tilde{x}_j, t) + \ell_{i1} \alpha(z, t) \right].$$

Note that there is no term on $g(z, t)$, since no control action is acting on the leader. Since $\sum_{j=1}^{N} \ell_{ij} = 0$ for all $i = 1, \dots, N$, we can add the term $\kappa g(z + \tilde{x}_i, t) \left(\sum_{j=1}^{N} \ell_{ij} \right) \alpha(z, t) = 0$, thus obtaining

$$\dot{\tilde{x}}_i = f(z + \tilde{x}_i, t) - f(z, t) - \kappa g(z + \tilde{x}_i, t) \sum_{j=2}^{N} \ell_{ij} \left[\alpha(z + \tilde{x}_j, t) - \alpha(z, t) \right]. \tag{4.24}$$

Furthermore, in the new coordinates, the leader-synchronization manifold defined in (4.17) corresponds to the origin of the $\tilde{x}$-dynamics. Now, given $(z_0, \tilde{x}_0, t_0)$ in $\mathbb{R}^{N n_x} \times \mathbb{R}$, let $\mathbf{T} > t_0$ be the time of existence of the solution of (4.24) initialized in $(z_0, \tilde{x}_0)$ at time t_0. For t in $[t_0, \mathbf{T})$, let $(Z(t), \widetilde{X}(t))$ denote this solution. Consider the function $\Gamma : [0, 1] \times [t_0, \mathbf{T}] \to \mathbb{R}^{N n_x}$, with $\Gamma = (\Gamma_2, \dots, \Gamma_N)$ which satisfies $\Gamma(s, t_0) = s\, \tilde{x}_0$, and where Γ_i, $i = 2, \dots, N$, is the solution of the following ordinary differential equation for $t_0 \leqslant t < t_0 + \mathbf{T}$

$$\frac{\partial \Gamma_i}{\partial t}(s, t) = f(\zeta_i(s, t), t) - f(Z(t), t) - \kappa g(\zeta_i(s, t), t) \sum_{j=2}^{N} \ell_{ij} (\alpha(\zeta_j(s, t), t) - \alpha(Z(t), t))$$

where we denoted $\zeta_i(s, t) = Z(t) + \Gamma_i(s, t)$. Note that, by uniqueness of the solution, Γ satisfies

$$\Gamma(0, t) = 0, \quad \Gamma(1, t) = \widetilde{X}(t), \qquad \forall t \in [t_0, \mathbf{T})\,. \tag{4.25}$$

Consider now the function V defined by

$$V = \sum_{i=2}^{N} V_i, \quad V_i(\cdot) = \int_0^1 \frac{\partial \Gamma_i}{\partial s}(s, \cdot)^\top P(\zeta_i(s, \cdot), \cdot) \frac{\partial \Gamma_i}{\partial s}(s, \cdot) ds. \tag{4.26}$$

Note that, for all coordinates (k,l) in $\{1,\dots,n_x\}^2$, we have

$$\begin{aligned}\frac{d}{dt}[P(\zeta_i(s,t),t)_{kl}] &= \frac{\partial P_{kl}}{\partial x}(\zeta_i(s,t),t)\frac{\partial \zeta_i}{\partial t}(s,t) + \frac{\partial P_{kl}}{\partial t}(\zeta_i(s,t),t),\\ &= \frac{\partial P_{kl}}{\partial x}(\zeta_i(s,t),t)\left[f\left(Z(t)+\frac{\partial \Gamma_i}{\partial t}(s,t)\right)\right] + \frac{\partial P_{kl}}{\partial t}(\zeta_i(s,t),t).\end{aligned}$$

This implies that for all vectors ν in $\mathbb{R}^{n_x}$, and $i=2,\dots,N$,

$$\begin{aligned}\frac{d}{dt}\left[\nu^\top P(\zeta_i(s,t),t)\nu\right] = \nu^\top \mathfrak{d}_f P(\zeta_i(s,t),t))\nu - \kappa\sum_{j=2}^{N}\Bigg[\ell_{ij}\sum_{\iota=1}^{n_u}\nu^\top \mathfrak{d}_{g_\iota}P(\zeta_i(s,t),t)\nu\\ \times(\alpha_\iota(\zeta_j(s,t),t)-\alpha_\iota(Z(t),t))\Bigg].\end{aligned}$$

By using the Killing vector assumption (4.21) and the integrability assumption (4.22), the time derivative of V_i becomes

$$\dot{V}_i(t) = \int_0^1 \left[\frac{\partial \Gamma_i}{\partial s}(s,t)^\top \mathcal{L}_f P(\zeta_i(s,t),t)\frac{\partial \Gamma_i}{\partial s}(s,t) - 2\kappa\sum_{j=2}^{N}\ell_{ij}\Omega_i(s,t)\Omega_j(s,t)^\top\right] ds, \quad (4.27)$$

where, for a generic index k, we defined

$$\Omega_k(s,t) := \frac{\partial \Gamma_k}{\partial s}(s,t)^\top P(\zeta_k(s,t),t)g(\zeta_k(s,t),t).$$

With the following notations,

$$D(s,t) := \operatorname{diag}\left(\mathcal{L}_f P(\zeta_i(s,t),t)\right)_{i=2,\dots,N}, \quad \Psi(s,t) := \operatorname{col}\left(\Omega_i(s,t)\right)_{i=2,\dots,N},$$

we compute the derivative of V as follows

$$\begin{aligned}\dot{V}(t) &= \int_0^1 \left[\frac{\partial \Gamma}{\partial s}(s,t)^\top D(s,t)\frac{\partial \Gamma}{\partial s}(s,t) - 2\kappa\Psi(s,t)L_{22}\Psi(s,t)^\top\right] ds\\ &\leqslant \int_0^1 \left[\frac{\partial \Gamma}{\partial s}(s,t)^\top D(s,t)\frac{\partial \Gamma}{\partial s}(s,t) - 2\kappa\underline{\mu}\Psi(s,t)\Psi(s,t)^\top\right] ds\end{aligned}$$

where in the second step we used Assumption 4.1 and Lemma A.3. Therefore, by selecting $\kappa \geqslant \frac{\rho}{2\underline{\mu}}$ with ρ satisfying inequality (4.20) and $\underline{\mu} > 0$ given by Lemma A.3, we obtain the bound $\dot{V}(t) \leqslant -\lambda V(t)$. By Gronwall's Lemma, this implies $V(t) \leqslant \exp\left(-\lambda(t-t_0)\right) V(t_0)$ for all $t \in [t_0, t_0+\mathbf{T})$. Moreover, with (4.25) it yields

$$V_i(t) \geqslant \underline{p}\int_0^1 \frac{\partial \Gamma_i}{\partial s}(s,t)^\top \frac{\partial \Gamma_i}{\partial s}(s,t)ds \geqslant \underline{p}\,|\widetilde{X}_i(t)|^2\,. \quad (4.28)$$

Therefore, since

$$V(t_0) \leqslant \bar{p}\,|\tilde{x}_0|^2, \quad (4.29)$$

it yields for all t in $[t_0, t_0 + \mathbf{T})$ that

$$|\widetilde{X}(t)|^2 \leqslant e^{-\lambda(t-t_0)} \frac{\bar{p}}{\underline{p}} |\tilde{x}^\circ|^2 .$$

Hence, remembering the definition $\tilde{x}_i = x_i - z$ and the fact that the leader trajectory is well-defined for all positive times, this implies that the trajectories are complete in positive time (i.e. $\mathbf{T} = +\infty$). Moreover, by equivalence of norms in finite dimensional spaces it follows that there exist two strictly positive real numbers $\underline{c}, \bar{c} > 0$ such that $\underline{c}|\mathbf{X}(\mathbf{x}_0, t, t_0)|_{\mathcal{M}} \leqslant |\widetilde{X}(t)| \leqslant \bar{c}|\mathbf{X}(\mathbf{x}_0, t, t_0)|_{\mathcal{M}}$, for all $t \geqslant t_0$, which implies (4.19) and concludes the proof. □

Remark 4.1. *The results presented in Theorem 4.3 can be extended to the output feedback scenario as discussed in [S2]. However, in this section, we will focus exclusively on the state-feedback form of* (4.14).

It is important to note that the integrability condition (4.22) and the Killing vector assumption (4.21) may not be practical from a learning perspective. Specifically, the constraints imposed by (4.21)-(4.22) become exceedingly difficult to satisfy when using function approximators, particularly deep neural networks (DNNs). As a result, we propose a relaxation of these constraints, which allows for approximate solutions without compromising convergence guarantees.

Relaxing the integrability condition. To relax the integrability constraint, we introduce a bounded error between the Jacobian of α and the function Pg. Consequently, we propose the following alternative assumption in place of Assumption 4.4.

Assumption 4.4b. *There exist a C^2 function $\alpha : \mathbb{R}^{n_x} \times \mathbb{R} \mapsto \mathbb{R}^{n_u}$ and a scalar $\varepsilon > 0$ such that, for $\iota = 1, \ldots, n_u$, the following holds*

$$\left| \frac{\partial \alpha_\iota}{\partial x}(x,t) - g_\iota(x,t)^\top P(x,t) \right| \leqslant \varepsilon, \quad \forall (t,x) \in \mathbb{R} \times \mathbb{R}^{n_x}. \tag{4.30}$$

This relaxation offers two key advantages. Firstly, it eliminates the requirement for Pg to be integrable, thereby expanding the set of admissible metric functions and facilitating its learning process. Secondly, it simplifies the learning of α as its Jacobian only needs to approximate the function Pg. Under this relaxed assumption, the results of Theorem 4.3 adapt as follows.

Theorem 4.4. *Consider a network $G = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ of agents* (4.14) *and let Assumptions 4.1, 4.2, 4.3 and 4.4b hold. Moreover, assume there exists a positive real number $\bar{g}$ such that $|g_\iota(x,t)| \leqslant \bar{g}$ for all (x,t) in $\mathbb{R}^{n_x} \times \mathbb{R}$ and $\iota = 1, \ldots, n_u$. Let $\underline{\mu}, \bar{\mu} > 0$ be given by Lemma A.3 and let $\bar{\ell} = \max_{ij} |\ell_{ij}|$ where (ℓ_{ij}) is an element of the Laplacian matrix associated to the graph. Then, if ε in Assumption 4.4b is selected as $\varepsilon \in [0, \varepsilon^*)$ with $\varepsilon^* = (\lambda \underline{\mu} \underline{p})(\rho \bar{\mu} \bar{p} \bar{g} n_u)$, there exists κ^* such that for any $\kappa \in [\frac{\rho}{2\underline{\mu}}, \kappa^*)$, the distributed state-feedback control law* (4.16)-(4.23) *solves the synchronization Problem 4.1.*

Proof. The proof is identical to the proof of Theorem 4.3 up to equation (4.27). Recalling the compact notation $\zeta_i(s,t) = Z(t) + \Gamma_i(s,t)$, with Assumption 4.2, for each i in $\{2, \ldots, N\}$, the function V_i defined in (4.26) satisfies for all t in $[t_0, \mathbf{T})$

$$\dot{V}_i(t) = \int_0^1 \left[\frac{\partial \Gamma_i}{\partial s}(s,t)^\top \mathcal{L}_f P(\zeta_i(s,t),t) \frac{\partial \Gamma_i}{\partial s}(s,t) - 2\kappa \sum_{j=2}^{N} \ell_{ij} \Omega_i(s,t) \Omega_j(s,t)^\top + T_i(s,t) \right] ds,$$

where we used the compact notation introduced in the proof of Theorem 4.3 and

$$T_i(s,t) := -2\kappa \frac{\partial \Gamma_i}{\partial s}(s,t)^\top \sum_{j=2}^{N} \ell_{ij} \sum_{\iota=1}^{n_u} P(\zeta_i(s,t),t) g_\iota(\zeta_i(s,t),t) \Sigma(s,t) \tfrac{\partial \Gamma_j}{\partial s}(s,t),$$
$$\Sigma(s,t) := \tfrac{\partial \alpha_\iota}{\partial x}(\zeta_j(s,t),t) - g_\iota(\zeta_j(s,t),t)^\top P(\zeta_j(s,t),t).$$

With Assumption 4.4b and the bounds on P and g, the term T_i can be bound as follows

$$|T_i(t,s)| \leqslant 2\kappa n_u \bar{p}\bar{g}\varepsilon \left| \sum_{j=2}^{N} \ell_{ij} \tfrac{\partial \Gamma_i}{\partial s}(s,t)^\top \tfrac{\partial \Gamma_j}{\partial s}(s,t) \right| = 2\kappa c \left| \sum_{j=2}^{N} \ell_{ij} \tfrac{\partial \Gamma_i}{\partial s}(s,t)^\top \tfrac{\partial \Gamma_j}{\partial s}(s,t) \right|.$$

Consequently, with Assumption 4.1 and the fact that L_{22} is bounded as in Lemma A.3, by following the proof of Theorem 4.3 we obtain

$$\dot{V}(t) \leqslant \int_0^1 \left[\frac{\partial \Gamma}{\partial s}(s,t)^\top D(s,t) \frac{\partial \Gamma}{\partial s}(s,t) - 2\kappa \underline{\mu} \Psi(s,t) \Psi(s,t)^\top + 2\kappa c \frac{\partial \Gamma}{\partial s}(s,t)^\top L_{22} \frac{\partial \Gamma}{\partial s}(s,t) \right] ds.$$

Invoking again Lemma A.3 and the bounds on P, the above inequality gives

$$\dot{V}(t) \leqslant \int_0^1 \frac{\partial \Gamma}{\partial s}(s,t)^\top \operatorname{diag}(\Upsilon_i)_{i=2,\dots,N} \frac{\partial \Gamma}{\partial s}(s,t)\, ds,$$

where

$$\Upsilon_i(\zeta_i(s,t),t) = \mathcal{L}_f P(\zeta_i(s,t),t) + 2\kappa P(\zeta_i(s,t),t) \left[\bar{c}\, \mathrm{I}_n - \underline{\mu} g(\zeta_i(s,t),t)^\top g(\zeta_i(s,t),t) P(\zeta_i(s,t),t) \right],$$

with $\bar{c} = \frac{\bar{\mu} n_u \bar{p} \bar{g} \varepsilon}{\underline{p}}$. With Assumption 4.2, this implies

$$\Upsilon_i(\zeta,t) \leqslant (2\kappa\bar{c} - \lambda) P(\zeta,t) + (\rho - 2\kappa\underline{\mu}) P(\zeta,t) g(\zeta,t)^\top g(\zeta,t) P(\zeta,t).$$

Note that, with the choice of ε in the statement of the theorem, we have $\frac{\rho}{2\underline{\mu}} < \frac{\lambda}{2\bar{c}}$. Consequently, for each $\kappa \in (\frac{\rho}{2\underline{\mu}}, \frac{\lambda}{2\bar{c}})$ it implies $\dot{V}(t) \leqslant -\tilde{\lambda} V(t)$ for all $t \geqslant t_0$, where $\tilde{\lambda} = 2\kappa\bar{c} - \lambda$ is a positive real number. The proof completes following the lines of the one of Theorem 4.3. □

It is important to notice that the relaxation of the integrability condition comes at the price of infinite-gain margin properties. Nevertheless, the approach still presents (reduced) robustness properties with respect to the controller gain. Moreover, the result maintains its global convergence characteristic. Finally, we remark that the allowed approximation error ε in Theorem 4.4 is strongly dependent on the network structure, which is highly reminiscent of the discrete-time scenario of Section 3.2 with non-infinite gain margin properties.

Relaxing the Killing vector field assumption. When the equality constraint in Assumption 4.3 is replaced by an approximation, global synchronization may be lost. However, it is shown in the following theorem that provided $|L_g P|$ is small enough, a semi-global result can be obtained.

Theorem 4.5. *Consider a network $G = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ of systems* (4.14)*. Suppose Assumption 4.1, 4.2, and 4.4 hold, and let $\kappa \geqslant \frac{\rho}{\underline{\mu}}$ be fixed. Moreover, assume there exists a positive real number $\bar{g}$ such that $|g_\iota(x,t)| \leqslant \bar{g}$ for all (x,t) in $\mathbb{R}^{n_x} \times \mathbb{R}$ and $\iota = 1, \dots, n_u$. Then, for each $\bar{\mathbf{x}} > 0$ there exist $k, \varepsilon > 0$ such that, if the following holds*

$$|\mathcal{L}_{g_\iota} P(x,t)| \leqslant \varepsilon \ , \ \forall\, (x,t,\iota) \in \mathbb{R}^{n_x} \times [t_0, \infty) \times \{1, \dots, n_u\}, \tag{4.31}$$

then, for all $(\mathbf{x}_0, t_0)$ in $\mathbb{R}^{Nn_x} \times \mathbb{R}$ such that $|\mathbf{x}_0|_{\mathcal{M}} \leqslant \bar{\mathbf{x}}$, the solution of (4.18) *with the distributed state-feedback control law* (4.16)*,* (4.23) *is defined for all $t \geqslant t_0$ and*

$$|\mathbf{X}(\mathbf{x}_0, t, t_0)|_{\mathcal{M}} \leqslant k e^{-\frac{\lambda}{3}(t-t_0)} |\mathbf{x}_0|_{\mathcal{M}} \ , \quad \forall t \geqslant t_0. \tag{4.32}$$

Proof. Let $\bar{\mathbf{x}}, t_0 > 0$ and let $\mathbf{x}^\circ \in \mathbb{R}^{Nn}$ satisfy $|\mathbf{x}^\circ|_{\mathcal{M}} \leqslant \bar{\mathbf{x}}$. Assume that (4.31) is satisfied for some positive real number ε that will be selected later on. As in the proof of Theorem 4.3, consider the function V defined in (4.26). With (4.28) and (4.29), there exists two positive real numbers $(\underline{c}_V, \bar{c}_V)$ such that

$$\underline{c}_V \big|\mathbf{X}(\mathbf{x}_0, t, t_0)\big|^2_{\mathcal{M}} \leqslant V(t) \leqslant \bar{c}_V \big|\mathbf{X}(\mathbf{x}_0, t, t_0)\big|^2_{\mathcal{M}}.$$

With Assumption 4.2, and 4.4, for each i in $\{2, \dots, N\}$, the function V_i defined in (4.26) satisfies for all t in $[t_0, \mathbf{T})$

$$\dot{V}_i(t) = \int_0^1 \left[\frac{\partial \Gamma_i}{\partial s}(s,t)^\top \mathcal{L}_f P(\zeta_i(s,t),t) \frac{\partial \Gamma_i}{\partial s}(s,t) - 2\kappa \sum_{j=2}^{N} \ell_{ij} \Omega_i(s,t) \Omega_j(s,t)^\top + \widetilde{T}_i(s,t) \right] ds,$$

where we used the compact notation introduced in the proof of Theorem 4.3 and

$$\widetilde{T}_i(s,t) := -\kappa \frac{\partial \Gamma_i}{\partial s}(s,t)^\top \sum_{\iota=1}^{n_u} \mathcal{L}_{g_\iota} P(\zeta_i(s,t),t) \sum_{j=2}^{N} \ell_{ij} \widetilde{\alpha}_\iota(\zeta_j(s,t),t) \frac{\partial \Gamma_i}{\partial s}(s,t).$$

with $\widetilde{\alpha}_\iota(\zeta_j(s,t),t) := \alpha_\iota(\zeta_j(s,t),t) - \alpha_\iota(Z(t),t)$. Note that, with Assumption 4.4 we have

$$\begin{aligned} \widetilde{\alpha}_\iota(\zeta_j(s,t),t) &= \int_0^1 \frac{\partial \alpha_\iota}{\partial x}(Z(t) + r\Gamma_j(s,t),t)) \Gamma_j(s,t) dr \\ &= \int_0^1 \frac{\partial \alpha_\iota}{\partial x}(Z(t) + r\Gamma_j(s,t),t)) \int_0^s \frac{\partial \Gamma_j}{\partial s}(\nu,t) d\nu dr. \end{aligned}$$

Hence, exploiting Assumption 4.4 and the bounds on P and g, we have the following bound

$$|\widetilde{\alpha}_\iota(\zeta_j(s,t),t)| \leqslant \bar{p}\bar{g} \int_0^1 \left| \frac{\partial \Gamma_j}{\partial s}(\nu,t) \right| d\nu \leqslant \bar{p}\bar{g} \frac{1}{2} \int_0^1 1 + \left| \frac{\partial \Gamma_j}{\partial s}(\nu,t) \right|^2 d\nu.$$

Employing the fact that $\int_0^1 |\frac{\partial \Gamma_i}{\partial s}(s,t)|^2 ds \leqslant \frac{V_i(t)}{\underline{p}}$, by (4.31) we obtain

$$|\widetilde{T}_i(s,t)| \leqslant \kappa\varepsilon \frac{n_u \bar{g}\bar{p}}{2\underline{p}} \sum_{j=2}^{N} \ell_{ij} (\underline{p} + V_j(t)) \frac{\partial \Gamma_i}{\partial s}(s,t)^\top \frac{\partial \Gamma_i}{\partial s}(s,t).$$

Consequently, V_i is bounded as

$$\dot{V}_i(t) \leqslant \int_0^1 \left[\frac{\partial \Gamma_i}{\partial s}(s,t)^\top \left(\mathcal{L}_f P(\zeta_i(s,t),t) + \kappa\varepsilon c \sum_{j=2}^{N} \ell_{ij}(\underline{p} + V_j(t)) \right) \frac{\partial \Gamma_i}{\partial s}(s,t) \right. \\ \left. -2\kappa \sum_{j=2}^{N} \ell_{ij} \Omega_i(s,t)\Omega_j(s,t)^\top \right] ds,$$

where $c = \frac{n_u \bar{g} \bar{p}}{2\underline{p}}$. Consequently,

$$\dot{V}(t) \leqslant \int_0^1 \left[\frac{\partial \Gamma}{\partial s}(s,t)^\top D(s,t) \frac{\partial \Gamma}{\partial s}(s,t) - 2\kappa\underline{\mu}\Psi(s,t)\Psi(s,t)^\top + \kappa\varepsilon c \frac{\partial \Gamma}{\partial s}(s,t)^\top M(t) \frac{\partial \Gamma}{\partial s}(s,t) \right] ds$$

where $M(t) = \mathrm{diag}(L_{22}\mathbf{V}(t))$ with $\mathbf{V}(t) = \mathrm{col}(\underline{p} + V_i(t))_{i=2,\dots,N}$. Note that

$$M(t) \leqslant \max_i \left\{ \sum_{j=2}^{N} \ell_{ij}(\underline{p} + V_j(t)) \right\} \mathrm{I}_{N-1} \leqslant \bar{\ell}((N-1)\underline{p} + V(t))\, \mathrm{I}_{N-1},$$

with $\bar{\ell} = \max_{ij} \ell_{ij}$. Then

$$\dot{V}(t) \leqslant \int_0^1 \left[\frac{\partial \Gamma}{\partial s}(s,t)^\top D(s,t) \frac{\partial \Gamma}{\partial s}(s,t) - 2\kappa\underline{\mu}\Psi(s,t)\Psi(s,t)^\top + \right. \\ \left. \kappa\varepsilon c\bar{\ell}(N-1)\underline{p} \frac{\partial \Gamma}{\partial s}(s,t)^\top \frac{\partial \Gamma}{\partial s}(s,t) + \kappa\varepsilon c\bar{\ell} V(t) \frac{\partial \Gamma}{\partial s}(s,t)^\top \frac{\partial \Gamma}{\partial s}(s,t) \right] ds.$$

By using the bounds on P and following the proof of Theorem 4.3, with $\kappa \geqslant \frac{\rho}{2\underline{\mu}}$ and $\bar{c} = c\bar{\ell}$ we obtain

$$\dot{V}(t) \leqslant -(\lambda - \bar{c}\kappa\varepsilon(N-1))V(t) + \frac{\bar{c}\kappa\varepsilon}{\underline{p}} \sum_{i=2}^{N} V_i^2 \leqslant -\Big(\lambda - \bar{c}\kappa\varepsilon(N-1) - \frac{\bar{c}\kappa\varepsilon}{\underline{p}} V\Big) V.$$

Finally, selecting $\varepsilon < \max\left\{ \frac{\lambda}{3\bar{c}\kappa(N-1)}, \frac{\lambda \underline{p}}{3\bar{c}\kappa\bar{c}_V \mathbf{x}} \right\}$ it implies $\dot{V} \leqslant -\frac{\lambda}{3}V$ for all $t \geqslant t_0$, concluding the proof. □

4.2.2 Deep Learning for metric and controller estimation

As previously discussed, a drawback of the proposed approach lies in the fact that metrics may not be easy to find in the Riemannian scenario. Moreover, even when a metric has been found, designing a control law satisfying the integrability condition (4.22) may not be straightforward. One way to overcome such difficulties is to rely on machine learning tools to obtain approximate solutions, thus leveraging on Theorem 4.5 and Theorem 4.4. In what follows, we combine the proposed control design with neural network architectures. Hence, the idea is to set up and approximately solve an optimization problem aimed at circumventing the need for an analytic metric. Once a suitable metric has been found via a first DNN, we train a second one to satisfy the

integrability condition. We now describe the proposed algorithm. Let us consider the problem of finding a suitable approximation of the metric first. We enforce symmetry by solely learning the upper triangular part of $P(x,t)$. The neural metric is constructed as

$$P^\theta(x,t) = \begin{pmatrix} p_1^\theta(x,t) & \cdots & p_{n_x}^\theta(x,t) \\ \vdots & \ddots & \vdots \\ p_{n_x}^\theta(x,t) & \cdots & p_{n_p}^\theta(x,t) \end{pmatrix},$$

where $n_p = \frac{n_x(n_x+1)}{2}$ is the total number of elements to be learned, the vector of matrix entries $\mathbf{p} = \mathrm{col}(p_1^\theta(x,t), \dots, p_{n_p}^\theta(x,t))$ is the output of the neural network $\mathrm{DNN}_P : \mathbb{R}^{n_x} \times \mathbb{R}^{n_\theta} \times \mathbb{R} \mapsto \mathbb{R}^{n_p}$ and $\theta \in \mathbb{R}^{n_\theta}$ is the vector of DNN_P parameters. To train the DNN_P parameters, we rely on Theorem 4.4 to relax the existence of a primitive for $P^\theta g$ and on Theorem 4.5 to loosen the constraint posed by the Killing vector field property (4.21). We set up an optimization problem asking for the minimization with respect to the parameters θ of the following cost function

$$J_P(x,\theta,\mathbf{w},\mathbf{c},t) = \sum_{i=1}^{4} w_i J_i(x,\theta,\mathbf{c},t), \tag{4.33}$$

being $\mathbf{w} = (w_1, \dots, w_4)$ a vector of (positive) scalar weights and

$$\begin{array}{ll} J_i(x,\theta,\mathbf{c},t) = \ln\left(\max\left\{\bar\lambda(M_i(x,\theta,\mathbf{c},t)),0\right\}+1\right), & i=1,4 \\ J_i(x,\theta,\mathbf{c},t) = \sum_{\iota=1}^{n_u} \ln\left(\max\left\{\bar\lambda(M_{i,\iota}(x,\theta,\mathbf{c},t)),0\right\}+1\right), & i=2,3 \end{array} \tag{4.34}$$

with $i = 1, \dots, 4$, $\bar\lambda$ being the maximum eigenvalue of M_i and M_i defined as

$$\begin{aligned} M_1(x,\theta,\mathbf{c},t) &= \mathcal{L}_f P^\theta(x,t) - c_1 P^\theta(x,t) g(x,t) g^\top(x,t) P^\theta(x,t) + c_2\, \mathrm{I}_{n_x} \\ M_{2,\iota}(x,\theta,\mathbf{c},t) &= \mathcal{L}_{g_\iota} P^\theta(x,t) - c_3\, \mathrm{I}_{n_x}, \\ M_{3,\iota}(x,\theta,\mathbf{c},t) &= -\mathcal{L}_{g_\iota} P^\theta(x,t) - c_3\, \mathrm{I}_{n_x}, \\ M_4(x,\theta,\mathbf{c},t) &= -P^\theta(x,t) + c_4\, \mathrm{I}_{n_x} \end{aligned}$$

where $c_i > 0$, $i = 1, \dots, 4$ are positive scalars to be learned and $\mathbf{c} = (c_1, c_2, c_3, c_4)$. First, it is worth noting that all matrices M_i, $i = 1, \dots, 4$ are symmetric, ensuring that their eigenvalues are real. Then, each cost J_i serves a specific purpose in relation to the neural metric P^θ. For example, J_1 introduces a positive cost when the contraction condition (4.20) is not satisfied. On the other hand, J_2 and J_3 promote the boundedness of $\mathcal{L}_g P$, relaxing the Killing vector condition (4.21). Importantly, the presence of imaginary eigenvalues would render M_2 and M_3 unrelated to (4.31). Lastly, J_4 guides the solution towards positive definite matrices, as stated in (4.20). It is important to note that the upper bound on P^θ is always satisfied, given that our algorithm is optimized within a compact set $\mathcal{S}$. The natural logarithm serves as a regularization term between the costs J_i. It allows for the rescaling of costs with significantly different magnitudes to similar values, facilitating a more precise determination of their relative importance using the weight vector $\mathbf{w}$. In parallel with the training of DNN_P, we train a *parameter estimator* to output the vector $\mathbf{c} = \mathrm{col}(c_i)_{i=1,\dots,4}$. The estimator and DNN_P collaborate to the minimization of the loss function (4.33). Notably, if the cost reaches a value of 0, all the contraction conditions are satisfied for the dataset and the learned estimator output, allowing for the termination of the learning process. The second step is to find a suitable law approximating the integrability condition (4.22). We train the parameters $\phi \in \mathbb{R}^{n_\phi}$ of the second network $\mathrm{DNN}_\alpha : \mathbb{R}^{n_x} \times \mathbb{R}^{n_\phi} \mapsto \mathbb{R}^{n_u}$ such that

$$J_\alpha(x,\phi,t) = \left| \frac{\partial \mathrm{DNN}_\alpha}{\partial x}(x,\phi,t) - g(x,t)^\top P^\theta(x,t) \right|^2 \tag{4.35}$$

is minimized. The full learning procedure is summarized by Algorithm 6. Clearly, the DNNs can be trained only on a dataset $\mathcal{D}$ of finite size. Yet, DNNs are typically Lipschitz-continuous functions. Hence, similarly to [228, Section IV], we provide a verification tool to assess the satisfaction of contraction conditions over compact sets once the training is over.

Proposition 4.2. *Let $\mathcal{S} \subset \mathbb{R}^{n_x}$ be an arbitrary compact set and $\mathcal{D} \subset \mathbb{R}^{n_x}$ a set with a finite number of elements and let $r > 0$ be such that*

$$\mathcal{S} \subseteq \cup_{x_i \in \mathcal{D}} \mathcal{B}_r(x_i), \quad \mathcal{B}_r(x_i) := \{x \in \mathbb{R}^{n_x} : |x - x_i| < r\}.$$

Let $M : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x \times n_x}$ be a Lipschitz-continuous matrix-valued function, with Lipschitz constant l_M, taking symmetric values and such that $M(x_i) \preceq -2qI$ for all $x_i \in \mathbb{D}$ and for some $q > 0$. If q, r, l_M are such that $q > r l_M$, then $M(x) \preceq -qI$ for all $x \in \mathcal{S}$.

Proof. By Lipschitz-continuity of M we have

$$\begin{aligned} M(x) &= M(x_i) + M(x) - M(x_i) \preceq M(x_i) + |M(x) - M(x_i)|I \\ &\preceq M(x_i) + l_M |x - x_i| I \preceq -q \left(2 - \frac{|x - x_i|}{r}\right) I \end{aligned}$$

for an arbitrary $x_i \in \mathbb{D}$. Then, $M(x) \preceq -qI$ for all $x \in \mathcal{B}_r(x_i)$. The result follows from the fact that $\mathcal{S} \subseteq \cup_{x_i \in \mathbb{D}} \mathcal{B}_r(x_i)$. □

Proposition 4.2 implies that if the dataset is composed of points from a sufficiently fine grid, then the learned properties extend to the points in between. Hence, we can obtain a valid metric over a compact set by learning on a finite number of samples. Similar reasoning can be proposed for the feedback law α. Since the estimated metric is a DNN and $g \in C^2$, their product is Lipschitz continuous on a compact set. Since α is also modeled as a DNN, by selecting smooth activation functions its Jacobian is continuous. Following similar arguments to those used in Proposition 4.2, we can finally guarantee that a bounded error on a grid translates to a bounded error on a compact set including it.

These results can be extended to the time-varying domain by assuming the dataset contains state-time pairs (trajectories). However, it is also necessary to assume bounded variation of the dynamics over time, as we want the DNN to generalize properly. This would require the use of Recurrent Neural Networks (RNNs) in order to embed time information in the approximations. For all these reasons, in the following illustrative example we focus on networks of time-invariant nonlinear systems and employ multi-layer perceptrons (MLPs), i.e., fully-connected feed-forward deep neural networks.

Example. We apply the proposed algorithm to a leader-synchronization problem[2]. We consider a network of $N = 6$ identical Lorenz attractors. Such systems are particularly interesting, since they can present a chaotic behavior. Each agent $i = 1, \dots, N$ is described by the following dynamics

$$\begin{cases} \dot{x}_{i,1} = a(x_{i,2} - x_{i,1}) + u_i \\ \dot{x}_{i,2} = x_{i,1}(b - x_{i,3}) - x_{i,2} + (2 + \sin(x_{i,1}))u_i \\ \dot{x}_{i,3} = x_{i,1} x_{i,2} - c x_{i,3} \end{cases}$$

[2]The code for reproducing the experiments proposed in this section can be found at `https://github.com/SamueleZoboli/Control-learning-multiagent-lorenz.git`

Algorithm 6 DNN-based synchronizing controller

1: Input: Dataset of $\left(x\,, f(x,t)\,, g(x,t)\,, \frac{\partial f}{\partial x}(x,t)\,, \frac{\partial g}{\partial x}(x,t)\right)$, $\text{DNN}_P, \text{DNN}_\alpha$;
2: **while** $J_P(x,\theta,\mathbf{w},\mathbf{c},t) \neq 0$ **do**
3: Train DNN_P and the estimator with (4.33);
4: **end while**
5: Train the DNN_α with (4.35);
6: Set the distributed law $u_i = -\kappa \sum_{j=1}^{N} \ell_{ij} \text{DNN}_\alpha(x_j, \phi, t)$.

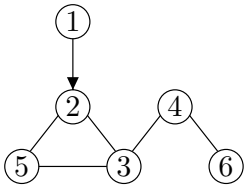
with $x_i = (x_{i,1}, x_{i,2}, x_{i,3}) \in \mathbb{R}^3$ and where $a = 10$, $b = 28$, $c = \frac{8}{3}$, guaranteeing the chaotic behavior. We consider the control matrix $g(x) = (1, 2 + \sin(x_{i,1}), 0)$ to exclude the possibility of feedback linearizing solutions. The agents communicate with each other following the leader-connected graph represented in Figure 4.1a.

We code and train our fully-connected DNNs and estimator using PyTorch [162]. For the metric network, we select an architecture composed of 4 hidden layers, with size $30, 20, 20, 10$ respectively and $\tanh$ activation functions. The output layer passes through a saturation function as a final activation, limiting the single elements of the metric. The second network presents 3 hidden layers, with size $30, 20, 10$ respectively and $\tanh$ activation functions. We select the identity function as output layer activation function. We select a weight vector $\mathbf{w} = (1, 10, 10, 20)$, directing the learning towards positive matrices first and successively satisfying the Killing-less assumptions and the contraction condition. We train both the networks and the estimator using Adam optimizer [114]. The learning rate for the metric network and the estimator is set as 3×10^{-3}, while DNN_α uses a learning rate of 5×10^{-3}. The DNNs learning rate are scheduled according to a cosine annealing policy [135], while the estimator one remains constant. We train the neural metric and the estimator over 100 epochs (yet stopped after 15 epochs due to the cost reaching 0) and the second DNN over 200 epochs. For both of the learning phases (the metric learning and the controller learning), the dataset is composed by 2×10^5 samples coming from a Gaussian distribution $\mathcal{N}(0, 10)$. We use 80% of the dataset as the training set, with a batch size of 512. The remaining 20% is used as test set. We select a $\kappa = 5$ and we apply the controller in a noisy-measurements scenario, i.e., $u_i = \varphi(x_i + \nu_i)$ where $\nu_i \sim \mathcal{N}(0, 0.2)$ represents some Gaussian measurement noise. This allows testing the robustness properties of the proposed neural control law. Each agents' initial condition is randomly sampled from a Gaussian distribution $\mathcal{N}(0, 20)$. Figures 4.1b and Figure 4.1c show the controller performances once the DNNs have been trained. Figure 4.1b presents the mean and standard deviation between agents of the norm of the error with respect to the leader trajectory. Figure 4.1c directly shows the state trajectories of each agent. As synchronization is achieved, we can see that the DNN optimized with (4.33) provides a suitable estimated metric, while the one trained with (4.35) effectively learns an approximate primitive of $P^\theta g$. The parameter estimator provided a decay rate $c_3 \approx 4.7$ and $c_1 \approx 36.3$. From Figure Figure 4.1c it is possible to see that the agents quickly synchronize, despite having significantly different initial conditions.

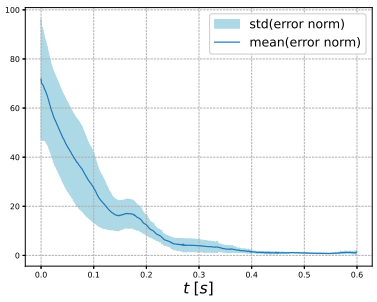
4.3 Learning tracking controllers for nonlinear systems

Building upon the results of the previous section, in this section we tackle the output tracking problem for continuous-time nonlinear systems by means of contraction analysis tools.

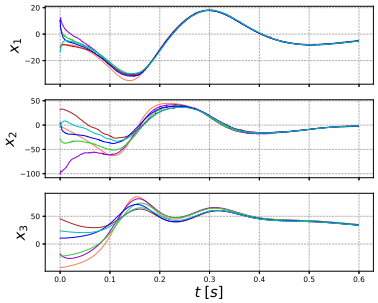
Output tracking is arguably among the most versatile applications of control theory, ranging



(a) Considered network of Lorentz oscillators.



(b) Evolution of the mean error norm between agents with respect to the leader.



(c) Evolution of the states (x_1, x_2, x_3) of agents a_i, $i = 1, \ldots, N$ (leader in orange).

Figure 4.1: Synchronization of Lorentz oscillators.

from vertical take-off in aerospace [141] and naval ships trajectories [234] to electronics [237] and mechanics [238]. The task consists in designing a control action leading the output of a dynamical system to track an arbitrary reference signal. Such a trajectory may be generated from manual design or any external source, depending on the control task. While being fairly simple to address on linear systems [77], output tracking remains an open problem for most general nonlinear dynamical systems. In this case, existing approaches either rely on heavy online computation or demand precise dynamical models.

Existing solutions canonically address the output tracking problem by exploiting one of the following tools: (i) model inversion, (ii) regulation theory or (iii) optimization. (i) The first method looks for an inverse model mapping the current state-target couple to the input transporting the former to the latter. As an example, we point the reader to solutions based on feedback linearization [101, Chapter 5.2] or [62] and references therein. (ii) The second design generalizes the linear method: the controller is divided in a dynamical part (the internal model) and a stabilizer [103]. The internal model component includes a generator of the steady-state solution where the tracking error is zero. The stabilizer provides convergence to such a solution (see e.g. [86] for the case of constant references). The overall control law guarantees stability, attractivity and forward-invariantness of a manifold where tracking is achieved. As for (i), the approach is strongly model-dependent. Related issues can be alleviated with possible countermeasures such as adaptive techniques [192]. Unfortunately, these tricks often leverage challenging analytical considerations, as they usually require a well-defined change of coordinates to bring the system in normal form and, in most cases, a minimum-phase assumption. (iii) Finally, the third method considers output tracking as an optimization problem, which motivates the use of the corresponding tools.

The latter is promising in many ways: it can cope with (small) model errors and it requires moderate theoretical analysis to be deployed, such as Model Predictive Control (MPC) [130]. The main drawback is the computational effort. MPC usually requires solving an online optimization problem for any point in the trajectory. The complexity of such a task drastically increases for systems presenting significant nonlinearities. To avoid the need of online computation, another class of optimization tools have emerged from the use of convex programming paired with DNN-powered interpolation [218]. Deep learning-based controllers are fast, versatile, and very efficient in most situations, but are very data intensive, and often requires a preexisting expert

agent to build a sufficiently informative dataset. A different option is to rely on the reinforcement learning philosophy [28], which leverages exploration guided by a weak reward signal. However, theoretical guarantees such as stability are challenging to obtain, which frequently limits existing results to simpler classes of systems, e.g., linear ones [42].

Similar to Section 4.2, in what follows we develop a solution to the output tracking problem which intertwines techniques from machine learning and control theory. To do so, we propose a DNN-based algorithm whose backbone comes from output-regulation theory. Formally, we propose a two-step controller. First, we estimate the solution of the regulator equations for a given reference signal, resulting in steady-state trajectory $\pi(t)$ and control action $\psi(t)$ minimizing the tracking error. Then, we design a stabilizer making trajectories asymptotically converge to the reference. To this aim, we rely on the results of Section 4.2. Hence, we revisit the output tracking problem in a multi-agent framework where a single follower has to synchronize to the leader. We design a stabilizer that makes the closed-loop a contraction and approximate its analytical solution with DNNs, adapting the approach of Section 4.2.2.

The proposed control structure is inspired by the results in [166, Section 5.4]. However, we highlight four main differences in our approach. (i) In [166, Section 5.4] the authors propose the design $u = \beta(x, \pi) = K(x - \pi)$, with K being a constant matrix and π the steady-state solution to the regulator equation. In our design, through the notion of Killing vector field (see Definition 4.3), we provide a more general structure for the controller β. (ii) We show that approximate, rather than asymptotic, tracking can be achieved under a non-perfect knowledge of the solution of the regulator equations. (iii) We provide a DNNs-based algorithm for the estimation of the unknown quantities that generalizes over references. (iv) We link the performances of the DNNs to the tracking error. The complete algorithm is modified with respect to Algorithm 6 in two main aspects. First, we add a "virtual-leader estimation" step by taking advantage of results coming from output regulation theory. The regulator equations are solved thanks to an end-to-end weakly supervised DNN that directly estimates steady-state variables from the reference. We show that our model performs well even on challenging references. Second, we introduce an auxiliary objective in the training of the neural metric. This secondary loss aims at refining and tightening the constraints imposed by the primary loss (4.33).

4.3.1 Problem statement and proposed approach

In what follows, we consider a system of the form

$$\dot{x} = f(x) + g(x)u \tag{4.36a}$$

$$e = h(x) - r(t) \tag{4.36b}$$

where $x \in \mathbb{R}^{n_x}$ is the state, $u \in \mathbb{R}^{n_u}$ is a control action, f, g, h are sufficiently smooth functions and $e \in \mathbb{R}^{n_e}$ is the error between an output $y = h(x)$ and a known smooth time-varying reference $r(t)$ taking values on a compact set $\mathcal{R} \subset \mathbb{R}^{n_e}$. In order to simplify the analysis, we assume forward completeness of the trajectories for all times $t \geqslant t_0, t_0 \in \mathbb{R}$ inside a forward invariant compact set $\mathcal{F} \subset \mathbb{R}^{n_x}$ and we define $\overline{g} := \sup_{x \in \mathcal{F}} |g(x)|$. Our goal is to design a feedback control action u such that the error e asymptotically goes to zero. We formalize our problem as follows. Let $c \geqslant 0$. Assume to know a smooth function $\gamma : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_u}$ such that the system (4.36) in closed-loop with the feedback control $u = \gamma(x, t)$ has bounded trajectories and such that $\lim_{t \to +\infty} |e(t)| \leqslant c$. Then:

- if $c = 0$, we say that the *asymptotic* output tracking control problem is solved;

- if $c > 0$, we say that the *approximate* output tracking control problem is solved.

From a regulation theory viewpoint, asymptotic output tracking can be achieved only if there exist two sufficiently smooth mappings $\pi : \mathbb{R}^{n_e} \to \mathbb{R}^{n_x}$ and $\psi : \mathbb{R}^{n_e} \to \mathbb{R}^{n_u}$ solution of the so-called *regulator equations* (see for instance [38, 103])

$$\begin{aligned} \dot{\pi}(r(t)) &= f(\pi(r(t))) + g(\pi(r(t)))\psi(r(t)) \\ 0 &= h(\pi(r(t))) - r(t)\,. \end{aligned} \tag{4.37}$$

Here, the mapping π represents the steady-state manifold on which the tracking error is zero. Hence, the state x must converge asymptotically to it. Similarly, the mapping ψ represents the steady-state control action making such a manifold forward invariant along the trajectories of the system. In order to have a well-posed problem, we assume that if $r(t) \in \mathcal{R}$ for all $t \geqslant t_0$, then $\pi(t)$ solving (4.37) is bounded and satisfies $\pi(t) \in \mathcal{F}$ for all $t \geqslant t_0$. Under perfect knowledge of the model and of the solution of the regulator equations, we look for a controller solving the asymptotic regulation problem and taking the form

$$u = \gamma(x, t) = \psi(t) + \beta(x, \pi, t), \tag{4.38}$$

where ψ solves (4.37) and β is any function that forces the dynamics of x to converge to the steady-state $\pi(t)$ and that is asymptotically vanishing. In other words, β is any function such that the trajectories of the closed-loop (4.36), (4.38) satisfy $\lim_{t\to+\infty}|x(t) - \pi(t)| = 0$ and such that $\beta(x, x, t) = 0$ for all $(x, t) \in \mathbb{R}^{n_x} \times \mathbb{R}$. Following these lines, we design β according to the leader-follower multi-agent synchronization strategy proposed in Section 4.2. As such, we consider a trivial directed graph whose network is composed by two agents. In particular, the leader is the steady-state dynamics in (4.37) and the follower is the closed-loop system (4.36), (4.38). With this approach, the problem of designing the function β in (4.38) can be seen as designing a control action achieving synchronization between two nonlinear systems whose open-loop dynamics is defined by

$$\dot{x} = \mathbf{f}(x, t) := f(x) + g(x)\psi(t)\,. \tag{4.39}$$

Hence, for controller design, we can exploits the results of Theorem 4.3. As reminded in Section 4.2.2, it is often hard to analytically compute contractive stabilizers for systems presenting significant nonlinearities. The same holds for the exact solution to the regulator equations. Hence, similarly to Section 4.2.2, we aim at solving approximate tracking with neural controllers equipped with convergence guarantees.

4.3.2 Approximate output tracking: the analytic solution

In order to justify the use of function approximators, we first highlight the robustness properties of the closed-loop system under the controller (4.38). The goal is to show that (i) under perfect knowledge of the system, of the regulation equations and of the control structure, the asymptotic output tracking problem is solved; (ii) it is still possible to achieve approximate output tracking by means of a approximate solution. Then, we show that the tracking error can be straightforwardly linked to the approximation errors of the estimated quantities, and we provide bounds guaranteeing approximate tracking up to arbitrary precision.
We start by assuming the following.

Assumption 4.5. *Consider system* (4.36)*,* (4.37) *and let* $\mathbf{f}(x, t) = f(x) + g(x)\psi(t)$*. There exist a* C^1 *matrix function* $P : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_x \times n_x}$ *taking symmetric positive definite values, a* C^2 *function*

$\beta : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_u}$, positive real numbers $\underline{p}, \bar{p}, \lambda, \rho > 0$ such that, for all $(x,t) \in \mathbb{R}^{n_x} \times \mathbb{R}$, the following holds:

(i) The matrix function P satisfies

$$\begin{aligned} &\mathcal{L}_{\mathbf{f}} P(x,t) - \rho P(x,t) g(x) g(x)^\top P(x,t) \preceq -\lambda P(x,t)\,, \\ &\underline{p} I \preceq P(x,t) \preceq \bar{p} I\,. \end{aligned} \tag{4.40}$$

(ii) There exists a function $\alpha : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_u}$ satisfying the integrability condition

$$\frac{\partial \alpha}{\partial x}(x,t)^\top = P(x,t) g(x)\,. \tag{4.41}$$

(iii) The Killing vector property holds, namely

$$\mathcal{L}_g P(x,t) = 0. \tag{4.42}$$

Clearly, Assumption 4.5 arises from the ones required in Theorem 4.3 and the integrability condition is assumed to simplify the analysis. However, we recall that such an assumption can be relaxed similarly to Section 4.2.2. We also highlight that (4.40) involves the closed-loop $\mathbf{f}$. Moreover, we assume the existence of a Riemannian metric P satisfying the Killing vector assumption (4.42). This is needed to ensure robust tracking with respect to input disturbances, which will be related to the approximation introduced by DNNs. In the case of relaxed Killing conditions as in Section 4.2.2, such robustness properties would hold only for small disturbances. Moreover, we remark that the Killing vector property would allow the definition of a time-invariant Riemannian metric P, as ψ enters φ via g. However, as perfect Killing conditions will not be achieved by the neural metric, we introduce time dependency of P by providing $\psi(t)$ as an additional entry to our network. We are now ready to present a first result on δISS-like properties of the closed-loop. We recall that a function $\omega \in \mathcal{L}_2$ if it is measurable and $\int_0^{+\infty} |w(s)|^2 ds < +\infty$.

Proposition 4.3. *Consider system* (4.36)*,* (4.37) *and let $\mathbf{f}(x,t) = f(x) + g(x)\psi(t)$. Let Assumption 4.5 hold and suppose $\omega : \mathbb{R} \to \mathbb{R}^{n_u}$ be in $\mathcal{L}_2$. Then, for any $\kappa \geqslant \frac{\rho}{2}$, the trajectories of system* (4.36) *in closed-loop with*

$$u = \psi(t) + \beta(x, \pi, t) + \omega(t) \tag{4.43a}$$

where

$$\beta(x, \pi, t) = -\kappa(\alpha(x,t) - \alpha(\pi,t)) \tag{4.43b}$$

satisfy

$$|X(x_0, t, t_0) - \Pi(\pi_0, t, t_0)| \leqslant k|x_0 - \pi_0| e^{-\lambda(t-t_0)} + \zeta(\sup_{s \in [t_0, t)} |\omega(s)|) \tag{4.44}$$

for all $(x_0, \pi_0, t, t_0) \in \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \times [t_0, \infty) \times \mathbb{R}$, for some $k, \lambda > 0$ and for some class-$\mathcal{K}_\infty$ function ζ, with $X(\cdot)$ being the trajectory of (4.36) *in closed-loop and $\Pi(\cdot)$ the trajectory of* (4.37)*.*

Proof. The proof follows the line of results in Section 4.2 and combines them with ISS-like arguments. Thus, we only highlight the main parts. Define the state-error $\tilde{x} := x - \pi$. Its dynamics read as

$$\dot{\tilde{x}} = \mathbf{f}(\pi + \tilde{x}, t) - \mathbf{f}(\pi, t) - \kappa g(\pi + \tilde{x})(\alpha(\pi + \tilde{x}, t) - \alpha(\pi, t)) + g(\pi + \tilde{x})\omega(t)\,. \tag{4.45}$$

Let $\widetilde{X}(\tilde{x}_0, t, t_0)$ be a solution defined for all $t \geqslant t_0$ and consider the function $\Gamma : [0,1] \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}^{n_x}$ satisfying $\Gamma(1, t_0, t_0) = \widetilde{X}(\tilde{x}_0, t_0, t_0)$, $\Gamma(0, t_0, t_0) = 0$ and $\Gamma(s, t_0, t_0) = \gamma(s)$, where

$\gamma : [0,1] \to \mathbb{R}^{n_x}$ is any C^1 curve and solution to

$$\frac{\partial \Gamma}{\partial t}(s,t,t_0) = \mathbf{f}(\Gamma+\Pi,t) - \mathbf{f}(\Pi,t) - \kappa g(\Gamma+\Pi)(\alpha(\Gamma+\Pi,t)-\alpha(\Pi,t)) + g(\Gamma+\Pi)w(s)$$

with $\Pi = \Pi(\pi_0,t,t_0)$ being the trajectory of (4.37) (arguments are dropped to ease notation) and $w(s) = s\omega$. Take the candidate Lyapunov function

$$V(t) = \int_0^1 \frac{\partial \Gamma}{\partial s}(s,t,t_0)^\top P(\Gamma+\Pi,t)\frac{\partial \Gamma}{\partial s}(s,t,t_0)\,ds \tag{4.46}$$

with P solving (4.40). Taking its time-derivative, by the Killing vector assumption and the integrability condition (4.41), we obtain

$$\dot{V}(t) \leqslant \int_0^1 \frac{\partial \Gamma}{\partial s}(s,t,t_0)^\top [T_1(s,t_0,t) + T_2(s,t_0,t)]\frac{\partial \Gamma}{\partial s}(s,t,t_0) + T_3(s,t_0,t)\,ds\,,$$

with

$$\begin{aligned}
T_1(s,t_0,t) &= \mathcal{L}_{\mathbf{f}} P(\Gamma+\Pi,t),\\
T_2(s,t_0,t) &= -2\kappa P(\Gamma+\Pi,t) g(\Gamma+\Pi) g^\top(\Gamma+\Pi) P(\Gamma+\Pi,t),\\
T_3(s,t_0,t) &= \mathrm{He}\left\{ \frac{\partial \Gamma}{\partial s}(s,t,t_0)^\top P(\Gamma+\Pi,t) g(\Gamma+\Pi,t)\omega(t) \right\}.
\end{aligned}$$

From the (generalized) Young's inequality [a] with parameters $a = \frac{\partial \Gamma}{\partial s}(s,t,t_0)^\top \sqrt{P(\Gamma+\Pi,t)}$, $b = \sqrt{P(\Gamma+\Pi,t)} g(\Gamma+\Pi)\omega(t)$ and $c = \frac{\lambda}{2}$ it follows that

$$T_3(s,t,t_0) \leqslant \frac{\lambda}{2}\frac{\partial \Gamma}{\partial s}(s,t,t_0)^\top P(\Gamma+\Pi,t)\frac{\partial \Gamma}{\partial s}(s,t,t_0) + \frac{2}{\lambda}\omega(t)^\top g(\Gamma+\Pi)^\top P(\Gamma+\Pi,t) g(\Gamma+\Pi)\omega(t)$$

Taking $\kappa \geqslant \frac{\rho}{2}$ and employing (4.40) we get

$$\dot{V}(t) \leqslant -\tfrac{\lambda}{2} V(t) + \tfrac{2}{\lambda}\overline{p}\,\overline{g}^2 |\omega(t)|^2\,.$$

From (4.40) and since $\widetilde{X}(\tilde{x},t,t) = \tilde{x}(t)$ for all t, it follows that, for any $t \geqslant t_0$

$$\underline{p}|\tilde{x}(t)|^2 \leqslant V(t) \leqslant \overline{p}|\tilde{x}(t)|^2\,. \tag{4.47}$$

Hence, the proof concludes by Gronwall's lemma and by following standard ISS-like arguments [204]. □

[a] Generalized inequality of Young: $2ab \leqslant ca^2 + \frac{b^2}{c}$ for any $c > 0$

The result of Proposition 4.3 shows that the control law (4.43) guarantees that trajectories of (4.36) in closed-loop remain close to the solution of (4.37). In particular, the error between the two depends on the component $\omega(t)$ in (4.43). Our objective is to approximate the control action with DNNs. Then, in our case $\omega(t)$ represents an approximation error. Without full knowledge of the model and of $(\pi(t),\psi(t))$ solution of (4.37), we end up using a control law of the form

$$u = \hat{\psi}(t) - \kappa(\hat{\alpha}(x,t) - \hat{\alpha}(\hat{\pi},t))\,, \tag{4.48}$$

where $\hat{\psi}, \hat{\pi}, \hat{\alpha}$ represent suitable approximations of the functions ψ, π, α in (4.43). In what follows, we link the error in the control action to the approximation capabilities of our DNN structure. More specifically, we show that if the functions $\hat{\psi}, \hat{\pi}, \hat{\alpha}$ are sufficiently close to ψ, π, α, then approximate regulation is still achieved. This lays strong foundations for the following section, as we exploit DNNs to learn an approximate version of the exact functions, which are not explicitly computable in general. Hence, via Proposition 4.3 and the following result, we highlight the link between the approximation and the tracking error.

Proposition 4.4. *Consider system* (4.36) *in closed-loop with the control law* (4.48). *Let* (π, ψ) *be a solution of* (4.37) *and let* $\mathbf{f}(x,t) := f(x) + g(x)\psi(t)$. *Let* (κ, α) *be chosen as in Proposition 4.3. Then, for any compact sets* $\mathcal{W}_{\tilde{x}} \subset \mathbb{R}^{n_x}$, $\mathcal{R} \subset \mathbb{R}^{n_e}$ *such that* $r(t) \in \mathcal{R}$ *for all* $t \geqslant t_0$ *and for any* $\delta \geqslant 0$, *there exist a compact set* $\mathcal{W}_x$ *and a scalar* $\mu_\delta \geqslant 0$ *such that, if the following holds for all* $(x,t) \in \mathcal{W}_x \times [t_0, \infty)$

$$|\hat{\alpha}(x,t) - \alpha(x,t)| \leqslant \mu_\delta\,, \quad |\hat{\psi}(t) - \psi(t)| \leqslant \mu_\delta\,, \quad |\hat{\pi}(t) - \pi(t)| \leqslant \mu_\delta\,, \tag{4.49}$$

then

$$\lim_{t \to +\infty} |\mathcal{X}(x_0, t, t_0) - \Pi(\pi_0, t, t_0)| < \delta.$$

for any (x_0, π_0) *satisfying* $(x_0 - \pi_0) \in \mathcal{W}_{\tilde{x}}$.

Proof. Consider the control law (4.43a). By adding and subtracting $\psi(t), \alpha(x,t), \alpha(\pi,t)$ and $\alpha(\hat{\pi},t)$, we rewrite the control (4.48) as $u(t) = u^\star(t) + \tilde{u}(t)$ with $u^\star := \psi(t) - \kappa(\alpha(x,t) - \alpha(\pi,t))$ and $\tilde{u}$ defined as

$$\tilde{u}(t) := \hat{\psi}(t) - \psi(t) - \kappa[(\hat{\alpha}(x,t) - \alpha(x,t)) + (\alpha(\hat{\pi},t) - \hat{\alpha}(\hat{\pi},t)) + (\alpha(\pi,t) - \alpha(\hat{\pi},t))]. \tag{4.50}$$

Consider the Lyapunov function (4.46) with $\tilde{x} = x - \pi$. Following the same lines as in the proof of Proposition 4.3, it follows that

$$\dot{V}(t) \leqslant -\tfrac{\lambda}{2} V(t) + \tfrac{2}{\lambda} \overline{p}\, \overline{g}^2 |\tilde{u}(t)|^2\,.$$

Consider now the reference r. Since $r(t) \in \mathcal{R}$ for all $t \geqslant t_0$, there exists a compact set $\mathcal{W}_\pi$ such that $\pi(t) \in \mathcal{W}_\pi$ for all $t \geqslant t_0$. Now, define

$$\eta_1 := \sup_{x \in \mathcal{W}_\pi} |x|, \quad \eta_2 := \sup_{x \in \mathcal{W}_{\tilde{x}}} |x|, \quad \eta_3 := \max\{\eta_2, \delta\}\,.$$

Then, $\mathcal{W}_{\tilde{x}} \subseteq \mathcal{B}_{\eta_3}$. Define $\overline{\mathcal{V}} := \left\{\tilde{x}_0 \in \mathbb{R}^n : V(t_0) \leqslant \bar{p}\eta_3^2\right\}$, and note that $\tilde{x}_0 \in \mathcal{B}_{\eta_3}$ implies $\tilde{x}_0 \in \overline{\mathcal{V}}$ due to (4.47). Differently put, $\mathcal{B}_{\eta_3} \subseteq \overline{\mathcal{V}}$. Pick $\mathcal{W}_x = \mathcal{B}_{\eta_4}$, where

$$\eta_4 = \eta_1 + \sup_{x \in \overline{\mathcal{V}}} |x|\,.$$

Then, $\overline{\mathcal{V}} \subseteq \mathcal{W}_x$. Moreover, if $\tilde{x}_0 \in \overline{\mathcal{V}}$, then $x_0 \in \mathcal{W}_x$. Pick

$$\mu_\delta = \frac{\lambda \delta \underline{p}}{2\sqrt{2}\, \overline{p}\, \overline{g}(1 + 2\kappa + \kappa \overline{p}\, \overline{g})}\,.$$

From (4.49), (4.50) and the relation (4.41), it follows that, for all times $t \geqslant t_0$ such that $x(t) \in \mathcal{W}_x$ the following holds

$$|\tilde{u}(t)| \leqslant \mu_\delta + 2\kappa\mu_\delta + \kappa\overline{p}\,\overline{g}|\pi(t) - \hat{\pi}(t)| \leqslant \mu_\delta(1 + 2\kappa + \kappa\overline{p}\,\overline{g}) \leqslant \frac{\lambda\delta\underline{p}}{2\sqrt{2}\,\overline{p}\,\overline{g}}\,.$$

Consider now the set $\underline{\mathcal{V}} := \{\tilde{x}_0 \in \mathbb{R}^{n_x} : V(t_0) < \underline{p}\delta^2\}$ and suppose $\tilde{x}_0 \in \underline{\mathcal{V}}$. By (4.47), if $\underline{p}|\tilde{x}_0|^2 < \underline{p}\delta^2$ then $|x_0 - \pi_0| < \delta$. Now, note that $\underline{\mathcal{V}} \subsetneq \mathcal{B}_{\eta_3} \subseteq \overline{\mathcal{V}}$ and suppose $\tilde{x}_0 \in \overline{\mathcal{V}} \setminus \underline{\mathcal{V}}$. This implies $|\tilde{x}_0|^2 \geqslant \frac{\underline{p}}{\overline{p}}\delta^2$. However, since $\tilde{x}_0 \in \overline{\mathcal{V}}$ implies $x_0 \in \mathcal{W}_x$, we have

$$\dot{V}(t_0) \leqslant -\lambda V(t_0) + \tfrac{2}{\lambda}\overline{p}\,\overline{g}^2|\tilde{u}(t_0)|^2 \leqslant \tfrac{\lambda}{2}(\tfrac{\underline{p}^2}{2\overline{p}}\delta^2 - V(t_0)) \leqslant \underline{p}\tfrac{\lambda}{2}(\tfrac{1}{2}\tfrac{\underline{p}}{\overline{p}}\delta^2 - |\tilde{x}_0|^2) < 0\,.$$

Hence, the level set $\overline{\mathcal{V}}$ is forward invariant. Moreover, in view of the above results, the set $\underline{\mathcal{V}}$ is attractive and forward invariant, with domain of attraction including $\overline{\mathcal{V}}$. Recall that $\mathcal{W}_{\tilde{x}} \subseteq \mathcal{B}_{\eta_3} \subseteq \overline{\mathcal{V}}$. Hence, for all $\tilde{x}_0 \in \mathcal{W}_{\tilde{x}}$, it holds $\lim_{t\to+\infty}|\mathcal{X}(x_0,t,t_0) - \Pi(\pi_0,t,t_0)| \leqslant \delta$. □

4.3.3 DNN-based output tracking controller

Our goal is to rely on DNNs to find approximate solutions for the output tracking problem [3]. DNNs are typically continuous functions by construction. Hence, if the approximation error over the training dataset is bounded, the error over a compact set including the dataset is also bounded by continuity. This allows the application of Proposition 4.4. Moreover, conditions related to the Lipschitz constant of the DNNs can be developed as per Lemma 4.2.

We split our controller into two parts: a steady-state component from solving the regulator equations and a stabilizing part based on Proposition 4.3. Instead of solving a time-consuming online optimization problem, the steady-state trajectory $\pi(t)$ is generated on the fly by our neural state reference generator. Such a generator is trained offline to solve the regulator equations. The stabilizer is derived from property (4.41) by modeling $\alpha(x,t)$ as a DNN and using Algorithm 6. Without loss of generality, we consider $t_0 = 0$. We also remark that the time-dependency of $\mathbf{f}$ is due only to the reference $r(t)$ (because of the tracking task). Thus, we discretize the problem using Euler scheme with a small timestep τ, yielding the time-discretized state $x(\mathfrak{t})$. In order to distinguish between feedforward and recurrent neural networks, we refer to $\mathfrak{m}(\cdot)$ as multi-layer perceptron (MLP) and $\upsilon^+ = \mathfrak{g}(\cdot,\upsilon)$ is a gated recurrent unit (GRU) with hidden state υ [45], where we omitted gating functions from the notation.

The state reference generator. The goal of the state reference estimator is to estimate state and command trajectories (π,ψ) verifying the dynamics (4.39) such that the observed part $h(\hat{\pi})$ remains as close as possible to the reference r. For a given time window $\mathbf{T} \in \mathbb{N}$, coherence with the dynamics can be ensured by forecasting solely the initial state $\pi(0) = \hat{\pi}_0$ and the set of steady-state inputs $\{\hat{\psi}_\mathfrak{t}\}_{\mathfrak{t}\in[0,\mathbf{T}]}$. The entire state trajectory can then be inferred readily from the dynamical model. Since the plant may not be fully observable from a single point, estimating the initial state can take advantage of longer reference signal $r = \{r_\mathfrak{t}\}_{\mathfrak{t}\in[0,\mathbf{T}]}$. We thus propose the following structure:

$$\left\{\begin{array}{ll} q^+ & = \mathfrak{g}_1\,(r,q,\phi_1) \\ \hat{\pi}_0 & = \mathfrak{m}_1(q(\mathbf{T}),\theta_1) \end{array}\right. , \quad \left\{\begin{array}{ll} z^+ & = \mathfrak{g}_2(\hat{\pi},r^+,z,\phi_2) \\ \hat{\psi} & = \mathfrak{m}_2(z,\theta_2) \end{array}\right. , \tag{4.51}$$

[3] Our code can be found at: `https://github.com/SteevenJanny/OutputTracking_contraction.git`

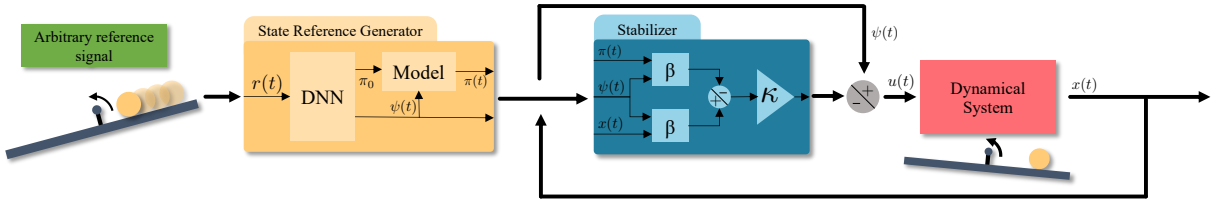


Figure 4.2: The *state reference generator* approximates the solution of the regulator equations and computes states and commands given an arbitrary reference signal. The *stabilizer* leverages a learned contraction based controller to force the dynamical system to track the reference.

with $\phi_1, \phi_2, \theta_1, \theta_2$ the parameters of the networks. We have two components: one dedicated to the estimation of the initial state $\pi(0)$ and another to the estimation of a one-step input. In the first one, the recurrent unit $\mathfrak{g}_1$ gathers temporal information from the reference into a single vector $q(\mathbf{T})$, which is then decoded to the initial state $\hat{\pi}_0$ through $\mathfrak{m}_1$. In the second component, the control signal $\hat{\psi}$ is inferred from the current state of the simulated system $\hat{\pi}$, the target reference observation r^+ and a latent memory vector z. The two parts combine as in the yellow box in Figure 4.2. The first component estimates $\pi(0)$ given the current reference. Then, the second component is invoked recursively to estimate a sequence of inputs $\{\hat{\psi}_t\}_{t\in[0,\mathbf{T}]}$. For each loop, successive states $\hat{\pi}_t$ are computed via the time-discretized steady-state dynamics $\hat{\pi}^+ = \hat{\pi} + (f(\hat{\pi}) + g(\hat{\pi})\hat{\psi})\tau$ with the estimated input $\hat{\psi}$. As such, the first component is used only once in the interval $[0, \mathbf{T}]$. However, note that the reference signal may change during this interval. In that case, a new estimate of π_0 is obtained by running through the state reference generator again. The model is trained to minimize the following objective:

$$\min_{\phi_1,\phi_2,\theta_1,\theta_2} \sum_{t=0}^{\mathbf{T}} |r_t - h(\hat{\pi}_t)|^2 \qquad \text{s.t.} \quad \hat{\pi}^+ = \hat{\pi} + (f(\hat{\pi}) + g(\hat{\pi})\hat{\psi})\tau\,. \tag{4.52}$$

The state reference generator is trained in an unsupervised manner, in the sense that no ground truth states and controls are needed for training. The references can be chosen arbitrarily insofar as these remain admissible by the dynamics. Nevertheless, the training requires prior knowledge of a model of the system (f, g, h). Since in many practical cases a good model may not be available, we robustify the state reference generator by training it with uniform noise on the dynamics. This exploits the generalization capabilities of DNNs by enriching the training set. Then, even if the model is faulty, the DNNs have more chances of producing suitable trajectories. Note that the bounds on the noise can be directly related to the confidence in the model.

Stabilizer. The stabilizer relies on the function $\hat{\alpha}$ modeled as an MLP. To include time dependence, the network inputs are both the system state and the steady-state input $\psi(t)$, namely $\hat{\alpha}(x,t) = \mathfrak{m}_3(x, \psi(t), \theta_3)$. This module is trained by following a two-step procedure, by using a modified version of Algorithm 6. The modified algorithm includes an auxiliary loss aimed at improving the estimation parameters $\mathbf{c} := (c_1, c_2, c_3, c_4)$ in (4.33). As in Section 4.2.2, we first look for a metric $P \in \mathbb{S}^{n_x}_{\succ 0}$ satisfying the synchronisation constraint (4.40) and the Killing vector property (4.42). This metric is also modeled as an MLP, i.e. $P(x,t) \approx \mathfrak{m}_4(x, \psi(t), \theta_4)$ with parameters θ_4, and it is initially trained according to the loss (4.33), where the time dependency is injected via ψ. Namely, the first component of training objective is

$$J_{P,1}(x, \theta_4, \mathbf{w}_1, \mathbf{c}, \psi) = \sum_{i=1}^{4} w_i J_i(x, \theta_4, \mathbf{c}, \psi),$$

where $\mathbf{w}_1 := (w_1, \dots, w_4)$ is a weight vector and J_i, $i = 1, \dots, 4$ are defined as in (4.34) with

$$\begin{aligned} M_1(x, \theta_4, \mathbf{c}, \psi) &= L_{\mathbf{f}} P^{\theta_4}(x, \psi) - c_1 P^{\theta_4}(x, \psi) g(x) g^\top(x) P^{\theta_4}(x, \psi) + c_2 \, \mathrm{I}_{n_x}, \\ M_{2,\iota}(x, \theta_4, \mathbf{c}, \psi) &= \mathcal{L}_{g_\iota} P^{\theta_4}(x, \psi) - c_3 \, \mathrm{I}_{n_x}, \\ M_{3,\iota}(x, \theta_4, \mathbf{c}, \psi) &= -\mathcal{L}_{g_\iota} P^{\theta_4}(x, \psi) - c_3 \, \mathrm{I}_{n_x}, \\ M_4(x, \theta_4, \mathbf{c}, \psi) &= -P^{\theta_4}(x, \psi) + c_4 \, \mathrm{I}_{n_x} \, . \end{aligned} \tag{4.53}$$

To the first loss component, we add a second one aimed at improving the vector of positive scalar parameters $\mathbf{c}$. Then, the objective is modeled as a switching loss function, composed by two interacting elements

$$J_P(x, \theta_4, \mathbf{w}_1, \mathbf{w}_2, \mathbf{c}, \psi) \quad = J_{P,1}(x, \theta_4, \mathbf{w}_1, \mathbf{c}, \psi) + \sigma J_{P,2}(\mathbf{w}_2, \mathbf{c}) \, , \tag{4.54}$$

with switch variable $\sigma = 0$ if $J_{P,1}(x, \theta_4, \mathbf{w}, \mathbf{c}, \psi) > 0$ for all pairs (x, ψ) in the dataset and $\sigma = 1$ otherwise. The second component activates once a suitable metric is found (i.e., once $J_{P,1}{=}0$). Its aim is to improve the estimation of $\mathbf{c}$, while looking for a better metric. Formally, it is defined as

$$J_{P,2}(\mathbf{w}_2, \mathbf{c}) = w_5 \ln(c_1^2 + 1) + w_6 \ln(c_3^2 + 1) - w_7 \ln(c_2^2 + 1) - w_8 \ln(c_4^2 + 1), \tag{4.55}$$

being $\mathbf{w}_2 := (w_5, \dots, w_8)$ a vector of (positive) scalar weights. The sub-objective (4.55) aims at minimizing c_1 and c_3, i.e., at reducing the controller dependence of the Riccati-like inequality (4.40) and at getting close to perfect Killing conditions. At the same time, it aims at maximizing c_2 and c_4, i.e., at increasing the contraction rate and the positivity of P. The composite objective (4.54) switches between metric search and contraction parameters optimization. First, it looks for a suitable metric along with set of parameters $\mathbf{c}$. Then, it freezes the metric estimator parameters θ_4 and tries improving the contraction parameters $\mathbf{c}$. If the metric still satisfies $J_{P,1}{=}0$, another step is taken in the direction of $\mathbf{c}$ improvement. If not, it unfreezes θ_4 and the loop starts again. Note that, by using $J_{P,1}$ as a discriminant, we can set the trained network to be the last one verifying $J_{P,1}(x, \theta_4, \mathbf{w}_1, \mathbf{c}, \psi){=}0$ for all (x, ψ) in the dataset.

There are multiple experimental advantages in using the switching objective (4.54). First, it achieves better estimation of parameters $\mathbf{c}$. Second, it improves controller robustness, e.g., smaller c_2 implies faster contraction, that is, better stability margins [203]. Third, it weakens the dependence of $\mathbf{c}$ from the initial condition. As a matter of fact, $\mathbf{c}$ can be initialized to looser bounds, which ease training. The second objective will then try to tighten the conditions (4.53) progressively. Finally, it can escape from local minima as the shape of the loss function drastically changes on switches.

Once a suitable metric is found, $\hat{\alpha}$ is learned according to (4.35). Each model of the stabilizer is trained on a training set composed of states and commands (x, ψ) from pre-trained state reference generator, thus ensuring consistency with future references generated at test time. We remark that the state reference generator and the stabilizer can be trained separately, as long as the training samples (x, ψ) for the stabilizer come from a similar distribution to the one of the output $\hat{\psi}(t)$ of the state reference generator. Training $\mathfrak{m}_3, \mathfrak{m}_4$ on the outputs of the state reference generator is a way to ensure this.

4.3.4 Illustrative example: ball and beam tracking Lorenz attractor

We test our solution on the well-known ball and beam setup, represented in Fig. 4.3 [96]. The plant can be described by a system of the form (4.36) where $x \in \mathbb{R}^4$, $u \in \mathbb{R}$ and

$$f(x){=}\,\mathrm{col}(x_2, b(x_1 x_4^2 - g_a \sin(x_3)), x_4, 0), \quad g(x){=}\,\mathrm{col}(0, 0, 0, 1), \quad h(x){=}x_1 \, .$$

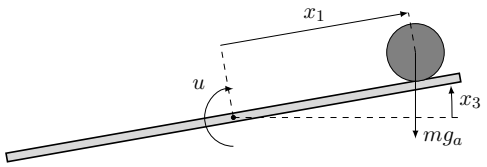


Figure 4.3: Block-scheme of the ball and beam system

From a physics viewpoint, $x{=}\,\mathrm{col}(x_1, x_2, x_3, x_4)$ with x_1, x_3 the ball position and beam angle respectively, b a constant depending on system parameters and g_a the gravitational acceleration. The interest of this setup lies in the fact that the relative degree[4] is not well-defined when the beam angular velocity and the ball position are zero. Therefore, input-output linearization and normal forms-based approaches fail to give a suitable controller. To make the problem harder, we sample the reference signal using the trajectory of the first component z_1 of a Lorenz oscillator whose dynamics is described by

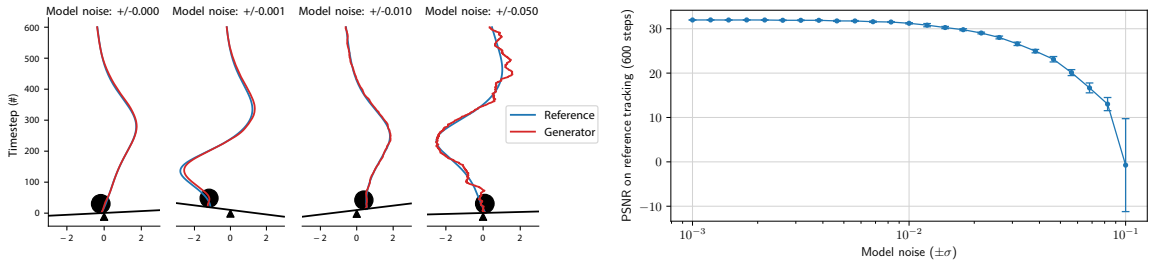
$$\dot{z}_1 = 10(z_2 - z_1), \quad \dot{z}_2 = z_1(28 - z_3) - z_2, \quad \dot{z}_3 = z_1 z_2 - \tfrac{8}{3} z_3, \tag{4.56}$$

with random initial conditions. As (4.56) is a chaotic oscillator, it is exponentially sensitive to initial conditions [155], making it hard to find analytical solutions to the regulator equations. Then, approaches as in [166] become unfeasible in practice. Here, $\mathfrak{m}_i\, i = 1, \ldots, 4$ are four layers, 64 hidden units and tanh-activated MLPs. Moreover, $\mathfrak{m}_1$ and $\mathfrak{m}_2$ use layer normalization on intermediate layers. All networks are trained with Adam optimizer [114].

State reference generator objective. The state reference generator objective is to estimate a trajectory in the state space as well as the commands allowing to follow it from a reference signal, which can potentially concern solely a part of the state. The sought solution is thus not necessarily unique. Nevertheless, our approach can exploit the long-range information of a reference sequence to estimate an adequate initial condition. We show examples of predictions from the state reference generator in Figure 4.4a. In a second time, we predict an input leading the system to follow the reference. This is a challenging task, especially since it is learned without direct supervision, that is, without knowledge of the optimal command for data in the training set. Our approach performs well, even in a very noisy environment. Figure 4.4b shows the evolution of the error when the dynamical model of the state reference generator is disturbed by a uniform noise whose amplitude is varied. The results are obtained on a set of new references absent from the training dataset. The standard deviation σ (represented by vertical bars) is obtained by averaging the results over five iterations. We empirically find that the state reference generator is consistently robust to model errors up to a significant intensity.

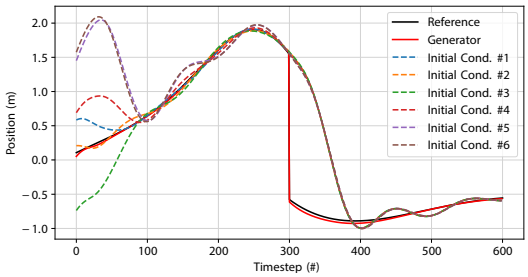
Behavior of the stabilizer. The stabilizer's closed-loop behavior is illustrated in Figure 4.5a. For a given new reference, the state reference generator estimates the solutions of the regulator equations (π, ψ). We then simulate the stabilizer starting from six random initial conditions x_0. Experimentally, we find that the system quickly converges towards the trajectory of the state reference generator. This is in accordance with the previous theoretical results. At timestep $\mathrm{t}{=}300$, we drastically change the reference signal. The generator reacts rapidly to such a change and estimates a new pair $(\pi(t), \psi(t))$. Thanks to the stabilizer, the system converges quickly to the new reference. Once finished, the trajectory remains close to the state reference without deviating

[4] see [101, Chapter IV]

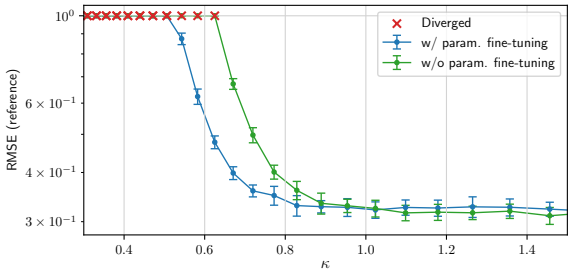


(a) (b)

Figure 4.4: State reference generator. (a) Four estimations from the state reference generator in different regimes where uniform noise is added to the model. (b) Peak signal to noise ratio (PSNR, dB) between the reference and the output for different noise range.



(a) Closed-loop behavior for six different initial conditions. The reference is changed at timestep t=300 timesteps.



(b) Admissible control gains κ with and without switching loss. Red crosses indicate that the system diverged from the reference signal.

from it. As mentioned above, the analytical solution to the output tracking problem is difficult to obtain with such a nonlinear system under chaotic references. Our approach, although it is an approximation of the analytical solution, experimentally demonstrates satisfactory performances. We report quantitative results in Table 4.1, considering a system perturbed by Gaussian noise modeling measurement errors. We observe that the learned stabilizer $\hat{\alpha}$ is robust even in noisy scenario. We also evaluate the advantages given by our switching objective, which reaches more stringent parameters $\mathbf{c}$. Table 4.1 compares noise robustness of our approach to the one without the fine-tuning component. We observe experimentally that our improved loss function leads to more robust control laws. Moreover, we also observe that fine-tuning allows for a wider set of control gains κ, (Figure 4.5b). This is linked to the approximate Killing vector property, see Theorem 4.5.

Parameter Finetuning	Noise StDev 0	0.01	0.05	0.1
With	**0.343** ± 0.015	**0.323** ± 0.015	**0.361** ± 0.011	**0.385** ± 0.014
Without	0.385 ± 0.015	0.366 ± 0.016	0.408 ± 0.012	0.439 ± 0.016

Table 4.1: Noise robustness with and without parameters fine-tuning. We measure RMSE from reference for different gaussian noise StDev. on state measurements.

5 From 1 to k-contraction

As previously stated, contraction theory is an emerging topic that has been used in numerous applications. Nonetheless, many systems cannot present classical contractivity properties, e.g. multi-stable systems. This motivated the study of suitable generalizations. Among others, some notable examples are horizontal contraction [72, Section VII], where contraction properties are required only on a subspace of the state space, transversal exponential stability [6], studying attractiveness of invariant manifolds via the analysis of exponential stability of the transverse plane, and p-dominance [74, 75], where convergence is imposed on a subspace by means of nonsingular matrices. Motivated by the results of Muldowney [151], recent works presented the notion of k-contraction [236], which generalizes the classical concept of shrinking distances between system trajectories to contraction of volumes. As such, k-contraction includes classical contraction as the special case $k = 1$. For $k > 1$, this property can be used to analyze asymptotic behavior of systems that are not contractive in the classical sense. For example, for 2-contractive time-invariant systems, every bounded solution converges to an equilibrium point (not necessarily unique).

Existing sufficient conditions for k-contraction are given in terms of a particular matrix compound of the Jacobian of the vector field dynamics [151, 236]. Although these conditions are adequate for system analysis, their application for feedback design is limited. First, matrix compounds rapidly explode in dimension for low value k and systems of large dimension. This fact strongly impacts the effectiveness of convex optimization tools for analysis, as conditions become computationally intractable. Moreover, it drastically increases the computational complexity of potential feedback design algorithms. Second, the use of matrix compounds tools prevents transforming the feedback design to a tractable LMI problem. Consequently, a k-contractive design methodology has yet to be developed.

Considering these limitations, in this chapter we propose alternative design-oriented conditions for k-contraction that do not rely on matrix compounds but rather on simple matrix inequalities on the given system dynamics. Due to this fact, our approach does not suffer from an explosion in computational complexity due to very different values of contraction order k and system dimension. Moreover, these inequalities can be solved via numerical method, thus opening multiple paths for future development of DNN-based controllers providing larger classes of guaranteed closed-loop behaviors. Finally, our conditions shed light on the strong connections between k-contraction, horizontal contraction and p-dominance [74, 75].

We proceed by formally presenting the definition of k-contraction. Our definition strongly focuses on geometrical interpretation and it is related to the notion presented in the works [151, 236]. Moreover, it directly translates to the definition of contraction presented in [6] when considering objects of dimension 1, i.e. when $k = 1$.

Consider nonlinear systems of the form

$$\dot{x} = f(t, x), \tag{5.1}$$

where $x \in \mathbb{R}^{n_x}$ and $f : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_x}$ is sufficiently smooth with respect to its first argument and continuous in the second. We denote the flow of (5.1) by ψ^t, so that $\psi^t(x_0)$ is the trajectory of (5.1) passing through x_0 at time 0. In [6], 1-contraction expresses the fact that the length of any C^1 curve from $[0, 1]$ to $\mathbb{R}^{n_x}$ decreases with time. To extend such a notion to any positive integer $k \in [1, n_x]$, with n_x being the state dimension of (5.1), we consider a set of sufficiently smooth functions $\mathcal{I}_k$ defined on $[0, 1]^k$, namely

$$\mathcal{I}_k := \Big\{ \Phi : [0, 1]^k \to \mathbb{R}^{n_x} \mid \Phi \text{ is a smooth immersion} \Big\}. \tag{5.2}$$

Let $P \in \mathbb{S}^{n_x}_{\succ 0}$. For each Φ in $\mathcal{I}_k$, we define the volume $\ell^k(\Phi)$ of Φ as

$$\ell^k(\Phi) := \int_{[0,1]^k} \sqrt{\det \left\{ \frac{\partial \Phi}{\partial s}(s)^\top P \frac{\partial \Phi}{\partial s}(s) \right\}} \, \mathrm{d}s \, . \tag{5.3}$$

Note that, since f in (5.1) is sufficiently smooth, for each forward invariant set and for each t in $\mathbb{R}_{\geqslant 0}$ it yields that the corresponding flow ψ^t is also sufficiently smooth in this set. Consequently, for each Φ in $\mathcal{I}_k$ such that $\mathrm{Im}(\Phi)$ is in a forward invariant set, $\psi^t \circ \Phi$ is in $\mathcal{I}_k$.

The relation between Definition 5.1 and Definition 4.1 is evident, since the 1-compound of a matrix is the matrix itself and the determinant of a scalar is the scalar itself. However, we highlight that the metric P in (4.1) is a matrix valued function, differently from (5.3). In other words, the proposed results will be restricted to Euclidean metric conditions and the Riemannian scenario is left for future research.

With these notions at hand, we can now define k-contraction properties for nonlinear systems of the form (5.1).

Definition 5.1 (k-contraction)**.** *Let k be a fixed integer between 1 and n_x. System* (5.1) *is said to be k-contractive on a forward invariant set $\mathcal{S} \subseteq \mathbb{R}^{n_x}$ if there exist strictly positive constants $\gamma, \eta > 0$ such that*

$$\ell^k(\psi^t \circ \Phi) \leqslant \gamma e^{-\eta t} \, \ell^k(\Phi), \qquad \forall t \in \mathbb{R}_{\geqslant 0},$$

for all $\Phi \in \mathcal{I}_k$ such that $\mathrm{Im}(\Phi) \subset \mathcal{S}$.

In simple words, we say that a system is k-contractive if, for any parametrized k-dimensional submanifold of $\mathbb{R}^{n_x}$ from which trajectories are complete, its volume is exponentially shrinking along the system dynamics. A scheme of this condition is depicted in Fig. 5.1. When $k = 1$, this means that the length of any sufficiently smooth curve is exponentially decreasing, matching the definition in [6]. Moreover, this definition includes the ones in [151], and [236, Section 3.2].

Remark 5.1. *When Φ is injective and P is the identity matrix,* (5.3) *gives the Euclidean k-volume of the submanifold $\Phi([0, 1]^k) \subset \mathbb{R}^{n_x}$. Note that 1-volumes are lengths, 2-volumes are areas and 3-volumes are standard volumes.*

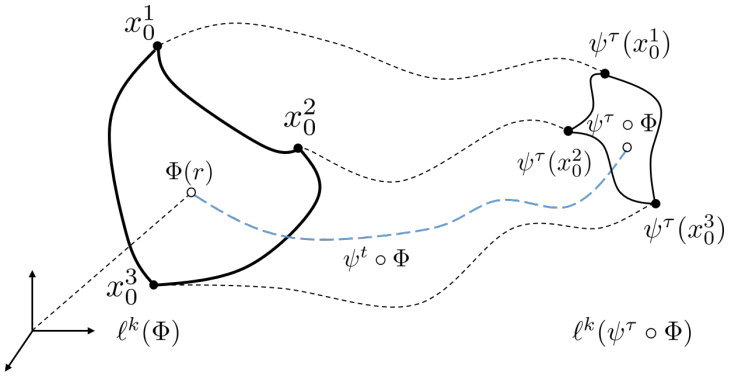


Figure 5.1: Scheme of a 2-contractive system. The initial submanifold, described by Φ, is some surface with vertices at x_0^1, x_0^2 and x_0^3. The volume of this submanifold $\ell^k(\cdot)$ decreases exponentially along the trajectories of the system.

5.1 k-contraction in nonlinear systems

Sufficient conditions for k-contraction of nonlinear systems originally appeared in the seminal work by Muldowney [151]. Recently, they have been rediscovered and reworked, e.g. [11,236]. However, the definition of k-contraction presented in the above works does not perfectly match the one in Definition 5.1. Hence, in order to clarify existing results, we first focus on briefly describing these sufficient conditions in the context of k-contraction as presented in Definition 5.1. Then, we proceed by presenting the recent concept of p-dominance and some related results. The section is concluded by linking p-dominance tools to k-contraction analysis, thus obtaining sufficient conditions that do not involve the computation of matrix compounds.

5.1.1 A sufficient condition based on matrix compounds

As previously stated, sufficient conditions provided in [151, 236] strongly depend on the use of matrix compounds. Hence, in order to understand such conditions, we start by introducing the notion of multiplicative and additive compound of a matrix. We recall that a minor of order k of a matrix $Q \in \mathbb{R}^{n\times m}$ is the determinant of some $k \times k$ submatrix of Q.

Definition 5.2 (Multiplicative Compound [151])**.** *Consider a matrix $Q \in \mathbb{R}^{n\times m}$ and let $k \in \mathbb{N}$ such that $k \in [1, min\{n, m\}]$. The k^{th} multiplicative compound of Q, denoted as $Q^{(k)} \in \mathbb{R}^{q\times q}$ with $q = \binom{n}{k}$, is the matrix of all the minors of order k of Q in a lexicographic order.*

Definition 5.3 (Additive Compound [151])**.** *Consider a matrix $Q \in \mathbb{R}^{n\times n}$ and let $k \in \mathbb{N}$ such that $k \in [1, n]$. The k^{th} additive compound of Q, denoted as $Q^{[k]} \in \mathbb{R}^{q\times q}$ with $q = \binom{n}{k}$, is defined as*

$$Q^{[k]} := \frac{\mathrm{d}}{\mathrm{d}\varepsilon}(I + \varepsilon Q)^{(k)}\big|_{\varepsilon=0}.$$

To give further intuition, we provide an example. Consider a 3×3 matrix Q with entries q_{ij} for $i, j = 1, \dots, 3$. The 2^{nd} multiplicative compound $Q^{(2)}$ is

$$Q^{(2)} = \begin{bmatrix} \det\left(\begin{smallmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{smallmatrix}\right) & \det\left(\begin{smallmatrix} q_{11} & q_{13} \\ q_{21} & q_{23} \end{smallmatrix}\right) & \det\left(\begin{smallmatrix} q_{12} & q_{13} \\ q_{22} & q_{23} \end{smallmatrix}\right) \\ \det\left(\begin{smallmatrix} q_{11} & q_{12} \\ q_{31} & q_{32} \end{smallmatrix}\right) & \det\left(\begin{smallmatrix} q_{11} & q_{13} \\ q_{31} & q_{33} \end{smallmatrix}\right) & \det\left(\begin{smallmatrix} q_{12} & q_{13} \\ q_{32} & q_{33} \end{smallmatrix}\right) \\ \det\left(\begin{smallmatrix} q_{21} & q_{22} \\ q_{31} & q_{32} \end{smallmatrix}\right) & \det\left(\begin{smallmatrix} q_{21} & q_{23} \\ q_{31} & q_{33} \end{smallmatrix}\right) & \det\left(\begin{smallmatrix} q_{22} & q_{23} \\ q_{32} & q_{33} \end{smallmatrix}\right) \end{bmatrix}.$$

The additive compound can be explicitly computed using the entries q_{ij} of the matrix Q. More details on this computation can be found in [71].

Bearing these definitions in mind, we now reframe the sufficient condition for k-contraction presented in [151, 236] in the framework of Definition 5.1.

Theorem 5.1. *Assume there exist a forward invariant set $\mathcal{A} \subseteq \mathbb{R}^{n_x}$, a symmetric positive definite matrix $Q \in \mathbb{R}^{q \times q}$ with $q = \binom{n_x}{k}$ and a real number $\mu > 0$ such that for all $(x,t) \in \mathcal{A} \times \mathbb{R}_{\geqslant 0}$*

$$Q\left(\frac{\partial f}{\partial x}(x,t)^{[k]}\right) + \left(\frac{\partial f}{\partial x}(x,t)^{[k]}\right)^{\top} Q \preceq -\mu Q. \tag{5.4}$$

Then, system (5.1) *is k-contractive on $\mathcal{S} := \mathcal{A}$.*

Proof. Consider a smooth immersion $\Phi \in \mathcal{I}_k$, with $\mathcal{I}_k$ defined as in (5.2). To simplify notation, for all (s,t) in $[0,1]^k \times \mathbb{R}_{\geqslant 0}$, let us denote

$$\Gamma(s,t) = \psi^t \circ \Phi(s), \quad \Gamma_s(s,t) = \frac{\partial \Gamma}{\partial s}(s,t), \quad v(s,t) = \left(\Gamma_s(s,t)^{(k)}\right)^{\top} P^{(k)}\, \Gamma_s(s,t)^{(k)}.$$

For all (s,t) in $[0,1]^k \times \mathbb{R}_{\geqslant 0}$, we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\Gamma(s,t) = f(\Gamma(s,t),t).$$

Then, by the chain rule, it follows that the point $\Gamma_s(s,t)$ evolves according to

$$\frac{\mathrm{d}}{\mathrm{d}t}\Gamma_s(s,t) = \frac{\partial^2 \Gamma}{\partial s \partial t}(s,t) = \frac{\partial f}{\partial x}(\Gamma(s,t),t)\Gamma_s(s,t)$$

for all (s,t) in $[0,1]^k \times \mathbb{R}_{\geqslant 0}$. Since these dynamics are linear, following similar steps to the ones presented in [236, Section 2.5], we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\Gamma_s(s,t)^{(k)} = \frac{\partial f}{\partial x}(\Gamma(s,t),t)^{[k]}\Gamma_s(s,t)^{(k)}. \tag{5.5}$$

Fix a symmetric positive definite matrix $P \in \mathbb{S}^{n_x}_{\succ 0}$ such that $Q = P^{(k)}$. Then, since $\Gamma_s(s,t) \in \mathbb{R}^{n_x \times k}$, from the Cauchy-Binet formula [67, Chapter 1] the following equality holds

$$\det\left(\Gamma_s(s,t)^{\top} P\, \Gamma_s(s,t)\right) = \left(\Gamma_s(s,t)^{(k)}\right)^{\top} P^{(k)}\, \Gamma_s(s,t)^{(k)} = v(s,t). \tag{5.6}$$

By, using the previous notation, the volume $\ell^k(\psi^t \circ \Phi)$ of $\psi^t \circ \Phi$ computed according to (5.3) takes the form

$$\ell^k(\psi^t \circ \Phi) = \int_{[0,1]^k} \sqrt{v(s,t)}\mathrm{d}s\,.$$

In turn, for all (s,t) in $[0,1]^k \times \mathbb{R}_{\geqslant 0}$, it evolves according to

$$\frac{\mathrm{d}}{\mathrm{d}t}\ell^k(\psi^t \circ \Phi) = \int_{[0,1]^k} \frac{\mathrm{d}}{\mathrm{d}t}\sqrt{v(s,t)}\,\mathrm{d}s = \int_{[0,1]^k} \frac{\mathrm{He}\left\{\left(\Gamma_s(s,t)^{(k)}\right)^{\top} Q \frac{\mathrm{d}}{\mathrm{d}t}\Gamma_s(s,t)^{(k)}\right\}}{2\sqrt{v(s,t)}}\mathrm{d}s\,.$$

Hence, for all (s,t) in $[0,1]^k \times \mathbb{R}_{\geqslant 0}$, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\ell^k(\Phi) = \int_{[0,1]^k} \frac{1}{2\sqrt{v(s,t)}} \left(\Gamma_s(s,t)^{(k)}\right)^\top \mathrm{He}\left\{Q\frac{\partial f}{\partial x}(\Gamma(s,t))^{[k]}, t\right\} \Gamma_s(s,t)^{(k)}\,\mathrm{d}s\,.$$

Invoking inequality (5.4), the previous relation implies

$$\frac{\mathrm{d}}{\mathrm{d}t}\ell^k(\psi^t \circ \Phi) \leqslant \int_{[0,1]^k} -\frac{\mu\, v(s,t)}{2\sqrt{v(s,t)}}\,\mathrm{d}s \leqslant -\frac{\mu}{2}\int_{[0,1]^k} \sqrt{v(s,t)}\,\mathrm{d}s \leqslant -\frac{\mu}{2}\ell^k(\psi^t \circ \Phi)$$

for all (s,t) in $[0,1]^k \times \mathbb{R}_{\geqslant 0}$. The result follows by Grönwall's lemma. □

Remark 5.2. *Inequality* (5.4) *is equivalent to the condition in [236, Theorem 9] using the logarithmic norm induced by the weighted ℓ_2 norm (e.g. [37, Equation 2.56]). However, in our statement, we allow the set $\mathcal{A}$ to be non-convex. Furthermore, for the case $k = 1$, we recover the well-known Demidovich conditions (e.g. [57, 163]) and the proof in [6] for contraction of lengths in the context of Euclidean metrics.*

5.1.2 A sufficient condition based on p-dominance

It is clear that computation of matrix compounds may become a difficult task, especially when $1 < k << n_x$. Moreover, as it will be clarified later on, the result in Theorem 5.1 do not easily translate to feedback design procedures and their use is restricted to analysis. Hence, we aim at proposing different conditions for k-contraction analysis which do not rely on matrix compounds. To this aim, we introduce the concept of *lifted system*. Then, inspired by classical works on contraction theory [134], we provide a result linking the exponential partial stability properties of the lifted system to the k-contraction property proposed in Definition 5.1.

The definition of k-contraction for the lifted system was used in [235] to study the link between k-contraction and other forms of generalized contraction. We now recall this definition and provide further geometrical interpretation of it. For an arbitrary Riemannian manifold $\mathcal{M}$, we denote by $T_x\mathcal{M}$ the tangent plane of $\mathcal{M}$ attached to a point x. The lifted system (also called *variational system*) is a linear dynamical system evolving according to dynamics obtained via linearization of (5.1) about the trajectory $\psi^t(x_0)$. Differently put, its trajectories evolve on the tangent boundle (the union of all tangent planes) of the space of trajectories of the nonlinear system. Mathematically, it takes the form

$$\dot{\delta_x} = \frac{\partial f}{\partial x}(\psi^t(x_0), t)\delta_x, \tag{5.7}$$

where δ_x belongs to the tangent space $T_{\psi^t(x_0)}\mathbb{R}^{n_x} = \mathbb{R}^{n_x}$. Then, $\frac{\partial \psi}{\partial x}^t(x_0)\delta_{x_0}$ is a trajectory of (5.7) at time t initialized at δ_{x_0} at $t = 0$. From linearity, it can be deduced that $\frac{\partial \psi}{\partial x}^t(x_0)$ is the state transition matrix of (5.7). Then, $\frac{\partial \psi}{\partial x}^t(x_0)\delta_{x_0}$ depicts the infinitesimal displacement with respect to the solution $\psi^t(x_0)$ induced by the initial condition $x_0 + \delta_{x_0}$. Hence, in order to define k-contraction of the lifted system (which we will call *infinitesimal k-contraction)*, pick any $x_0 \in \mathbb{R}^{n_x}$ and k initial conditions of the variational system in (5.7) $\delta^1_{x_0}, \ldots, \delta^k_{x_0}$. Following [235], we define

$$X_{\mathrm{NL}}(x_0,t) := \left[\begin{array}{ccc} \frac{\partial \psi}{\partial x}^t(x_0)\delta^1_{x_0} & \cdots & \frac{\partial \psi}{\partial x}^t(x_0)\delta^k_{x_0} \end{array}\right]. \tag{5.8}$$

Note that $X_{\mathrm{NL}}(x_0, 0) = \frac{\partial \Phi_{\mathrm{loc}}}{\partial r}(r)$, where Φ_{loc} is a function whose image is an infinitesimal k^{th}-order parallelotpe with vertices at x_0 and $\delta^i_{x_0} + x_0$, namely

$$\Phi_{\mathrm{loc}}(s) = \sum_{i=1}^{k} s_i(\delta^i_{x_0} + x_0) + \left(1 - \sum_{i=1}^{k} s_i\right) x_0, \ s \in [0,1]^k.$$

We have the following result relating [235] to Definition 5.1.

Lemma 5.1 (Infinitesimal k-contraction)**.** *Suppose there exist a forward invariant set $\mathcal{A} \subseteq \mathbb{R}^{n_x}$ and strictly positive constants γ and η such that the following holds for all $(x_0, t) \in \mathcal{A} \times \mathbb{R}_{\geqslant 0}$*

$$\left|(X_{\mathrm{NL}}(x_0, t))^{(k)}\right| \leqslant \gamma e^{-\eta t} \left|(X_{\mathrm{NL}}(x_0, 0))^{(k)}\right|, \tag{5.9}$$

Then, system (5.1) *is k-contractive on $\mathcal{S} := \mathcal{A}$.*

Proof. Following [236, Section 2.5], it can be shown that the compound matrix of $X_{\mathrm{NL}}(x_0, t)$ evolves according to the linear dynamics

$$\frac{d}{dt}\left(X_{\mathrm{NL}}(x_0, t)\right)^{(k)} = \frac{\partial f}{\partial x}(t, X_{\mathrm{NL}}(x_0, t))^{[k]} \left(X_{\mathrm{NL}}(x_0, t)\right)^{(k)}. \tag{5.10}$$

By (5.9), such dynamics are globally exponentially stable. Consider now an arbitrary $\Phi \in \mathcal{I}_k$. By following the first steps of the proof of Theorem 5.1, dynamics (5.5) and uniformity of (5.10) imply

$$\left|\Gamma_s(s, t)^{(k)}\right| \leqslant \gamma e^{-\eta t} \left|\Gamma_s(s, 0)^{(k)}\right|.$$

Then, by selecting P in (5.3) as the identity matrix, by (5.6) we obtain

$$\begin{aligned}\ell^k(\psi_t \circ \Phi) &= \int_{[0,1]^k} \left|\Gamma_s(s, t)^{(k)}\right| \, \mathrm{d}s \int_{[0,1]^k} \gamma e^{-\eta t} \left|\Gamma_s(s, 0)^{(k)}\right| \, \mathrm{d}s \\ &\leqslant \gamma e^{-\eta t} \int_{[0,1]^k} \left|\Gamma_s(s, 0)^{(k)}\right| \, \mathrm{d}s \leqslant \gamma e^{-\eta t} \ell^k(\Phi),\end{aligned}$$

and this concludes the proof. □

Remark 5.3. *Notice that the sufficient condition* (5.4) *also implies exponential stability of* (5.10)*, in turn implying infinitesimal k-contraction.*

With the above notions at hand, we can start introducing conditions for k-contraction which do not involve the computation of matrix compounds. As a matter of fact, k-compounds will be used in our proofs but, thanks to Lemma 5.1, they will not appear in the conditions. Our goal is to impose partial stability conditions allowing the presence of unstable directions, but to limit their rate of explosion. Indeed, if the rate of convergence is faster then the explosion one, k^{th}-order volume shrinks. To ease intuition, consider a rectangle with an expanding and a contracting side. If the contracting side shrinks faster than the expanding one, the area of the rectangle goes to zero. Conversely, if the contracting side shrinks at a lower rate, the area diverges in time. To impose such a constraint, we will leverage the theory of p-dominance [74, 75]. Hence, we rely on matrices with specific numbers of positive and negative eigenvalues. This generalizes classical conditions for 1-contraction based on Euclidian norms by allowing some directions of the state space to be unstable. Then, we first introduce the concept of inertia of a matrix with respect to the imaginary axis, see e.g. [208, Definition 2.1].

Definition 5.4 (Matrix inertia wrt the imaginary axis [208])**.** *The inertia of a matrix P with respect to the imaginary axis is defined by the triplet of integers $\mathrm{In_c}(P) := (\pi_-(P), \pi_0(P), \pi_+(P))$, where $\pi_-(P)$, $\pi_+(P)$ and $\pi_0(P)$ denote the numbers of eigenvalues of P with negative, positive and zero real part, resp., counting multiplicities.*

Since our conditions arise from the union between p-dominance and k-contraction analysis, we need to recall some results from the literature. However, we will restrict our analysis to the case of time-invariant dynamics. Consider nonlinear systems of the form

$$\dot{x} = f(x), \tag{5.11}$$

where $x \in \mathbb{R}^n_x$ and f is sufficiently smooth with respect to its argument. As before, the flow of (5.11) is denoted by ψ^t, so that $\psi^t(x_0)$ is the trajectory of (5.11) passing through x_0 at time 0. We start by recalling the definition of p-dominance, which has been related to various differential properties [75, Section V], such as differential positiveness [73] and monotonicity [12].

Definition 5.5 (p-dominance)**.** *System* (5.11) *is said to be strictly p-dominant on $\mathcal{S} \subseteq \mathbb{R}^{n_x}$ if there exist a real number $\mu > 0$ and a symmetric matrix $P \in \mathbb{R}^{n_x \times n_x}$ with inertia with respect to the imaginary axis $\mathrm{In_c}(P) = (p, 0, n-p)$ such that*

$$P\frac{\partial f}{\partial x}(x) + \frac{\partial f}{\partial x}(x)^\top P \prec -2\mu P\,, \quad \forall x \in \mathcal{S}. \tag{5.12}$$

Then, we recall (with a mild reformulation) the following result on p-dominance [75, Theorem 1], which plays a fundamental role in our findings.

Theorem 5.2. *Suppose that system* (5.11) *is strictly p-dominant on a compact forward invariant set $\mathcal{A} \subset \mathbb{R}^{n_x}$ with rate $\mu > 0$ and symmetric matrix P with inertia with respect to the imaginary axis $\mathrm{In_c}(P) = (p, 0, n-p)$. Then, for each $x \in \mathcal{A}$, there exists an invariant splitting $T_x\mathbb{R}^{n_x} = \mathcal{V}_x \oplus \mathcal{H}_x$, i.e. there exists a continuous mapping $\mathbf{T} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x \times n_x}$ invertible for any $x \in \mathcal{A}$ and satisfying*

$$\mathbf{T}(x) := \begin{bmatrix} \mathbf{T_h}(x) & \mathbf{T_v}(x) \end{bmatrix}, \tag{5.13a}$$

where $\mathbf{T_h} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x \times n_x - p}$ and $\mathbf{T_v} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x \times p}$ satisfy

$$\mathrm{Im}\ \mathbf{T_h}(x) = \mathcal{H}_x, \quad \mathrm{Im}\ \mathbf{T_v}(x) = \mathcal{V}_x. \tag{5.13b}$$

Moreover, there exist a scalar $c_h > 0$ such that

$$\left| \frac{\partial \psi^t}{\partial x}(x) \begin{bmatrix} \mathbf{T_h}(x) & 0 \end{bmatrix} \delta_x \right| \leqslant c_h e^{-\mu t} \left| \begin{bmatrix} \mathbf{T_h}(x) & 0 \end{bmatrix} \delta_x \right| \tag{5.13c}$$

holds for all $(t, x, \delta_x) \in \mathbb{R}_{\geqslant 0} \times \mathcal{A} \times T_x\mathbb{R}^{n_x}$.

Theorem 5.2 says that the p-dominance condition (5.12) splits the tangent space in a vertical subspace of dimension p and a horizontal subspace of dimension $n_x - p$. More precisely, for each initial condition $x_0 \in \mathcal{S}$, the tangent space can be divided in a horizontal distribution $\mathcal{H}_x$ and a vertical distribution $\mathcal{V}_x$. Then, the property of p-dominance can be interpreted as a form of horizontal contraction [72, Section VII], in the sense that contraction is only imposed in the horizontal subspace. However, horizontal contraction is not a sufficient condition for k-contraction [235], and a bound on the expansion rate of the vertical subspace has to be imposed. To this aim, we present the following technical lemma.

Lemma 5.2. *Consider system* (5.11) *and assume there exist a forward invariant compact set* $\mathcal{A} \subset \mathbb{R}^{n_x}$*, a matrix* $P \in \mathbb{S}^{n_x}_{\succ 0}$ *and a scalar* μ *satisfying*

$$\frac{\partial f}{\partial x}(x)^\top P + P\frac{\partial f}{\partial x}(x) \prec 2\mu P, \tag{5.14}$$

for all $x \in \mathcal{A}$*. Then there exists a constant* $c_v > 0$ *such that*

$$\left|\frac{\partial \psi}{\partial x}^t(x)\begin{bmatrix}0 & \mathbf{T_v}(x)\end{bmatrix}\delta_x\right| \leqslant c_v e^{\mu t}\left|\begin{bmatrix}0 & \mathbf{T_v}(x)\end{bmatrix}\delta_x\right| \tag{5.15}$$

for all $(t, x, \delta_x) \in \mathbb{R}_{\geqslant 0} \times \mathcal{A} \times T_x\mathbb{R}^{n_x}$*, with* $\mathbf{T_v}$ *as in* (5.13b)*.*

Proof. Consider the function, $W := \delta_x^\top P \delta_x$. It satisfies

$$\underline{\lambda}(P)|\delta_x|^2 \leqslant W \leqslant \overline{\lambda}(P)|\delta_x|^2, \tag{5.16}$$

where $\underline{\lambda}(\cdot)$ and $\overline{\lambda}(\cdot)$ represent the minimum and maximum eigenvalue of their argument, respectively. By (5.7), its time-derivative satisfies

$$\begin{aligned}\dot{W} &= \delta_x^\top\left(P\frac{\partial f}{\partial x}(x) + \frac{\partial f}{\partial x}(x)^\top P\right)\delta_x \\ &\leqslant 2\mu_0 \delta_x^\top P \delta_x = 2\mu W.\end{aligned}$$

Then, by Grönwall–Bellman inequality, we obtain

$$W(t) \leqslant W(0)e^{\int_0^t 2\mu d\tau} = e^{2\mu t}W(0), \quad \forall t \in \mathbb{R}_{\geqslant 0}.$$

Invoking (5.16), we obtain for all $(t, x, \delta_x) \in \mathbb{R}_{\geqslant 0} \times \mathcal{A} \times T_x\mathbb{R}^{n_x}$

$$\left|\frac{\partial \psi}{\partial x}^t(x)\delta_x\right| \leqslant \sqrt{\frac{\overline{\lambda}(P)}{\underline{\lambda}(P)}}e^{\mu t}|\delta_x|.$$

As $\begin{bmatrix}0 & \mathbf{T_v}(x)\end{bmatrix}\delta_x \in T_x\mathbb{R}^{n_x}$, the result trivially follows. □

Lemma 5.2 provides bounds on the expansion rate of the vertical subspace defined by the p-dominance condition. Then, given the above results, we can now relate expansion and contraction rates of the subspaces defined by Theorem 5.2. Since our aim is to exploit Lemma 5.1, we first present a technical lemma related to matrix compounds.

Lemma 5.3. *Consider a time-varying matrix* $M(t) \in \mathbb{R}^{n_x \times n_x}$

$$M(t) = \begin{bmatrix}H(t) & V(t)\end{bmatrix},$$

with $H(t) \in \mathbb{R}^{n_x \times n_x - p}$*,* $V(t) \in \mathbb{R}^{n_x \times p}$ *and* $p \in [0, n_x)$*. Assume there exist real numbers* $c_h, c_v, \alpha, \varphi > 0$ *such that*

$$|H(t)| \leqslant c_h e^{-\alpha t}, \quad |V(t)| \leqslant c_v e^{\varphi t}, \qquad \forall t \in \mathbb{R}_{\geqslant 0}. \tag{5.17}$$

If $\alpha > (k-1)\varphi$ *for some integer* $k \in [p+1, n_x]$*, there exist some real numbers* $c, \varepsilon > 0$ *such that*

$$|M(t)^{(k)}| \leqslant ce^{-\varepsilon t}, \qquad \forall t \in \mathbb{R}_{\geqslant 0}. \tag{5.18}$$

Proof. Consider the elements of the compound matrix $M(t)^{(k)}$. Each one is a k^{th}-order minor of the original matrix $M(t)$, i.e., it is the determinant of a $k \times k$ submatrix of $M(t)$, see Definition 5.2. Since $k \geqslant p+1$, each $k \times k$ submatrix contains at least one column composed of elements of $H(t)$. That is, in the minimum case

$$M_k(t) = \begin{bmatrix} h(t) & v_1(t) & \dots & v_{k-1}(t) \end{bmatrix}, \tag{5.19}$$

where $M_k(t) \in \mathbb{R}^{k\times k}$ is a submatrix of $M(t)$, $h(t) \in \mathbb{R}^k$ is a vector with components of $H(t)$ and $v_i(t) \in \mathbb{R}^k$ for $i = 1, \dots, k-1$ is a vector with components of $V(t)$. In what follows, we show the elements of $M(t)^{(k)}$ are bounded. Hence, we focus on submatrices of the form (5.19), since their determinant represents the worst-case scenario in a stability sense. Recall that, by definition of the wedge product,

$$\det(M_k(t)) = h(t) \wedge v_1(t) \wedge \dots \wedge v_{k-1}(t).$$

The wedge product can be represented using a basis e^i, where e^i depicts the i^{th} canonical vector of $\mathbb{R}^{n_x}$. More specifically, by bilinearity of the wedge product, we have

$$\det(M_k(t)) = \sum_{i=1}^{n} h^i(t)(e^i \wedge v_1(t) \wedge \dots \wedge v_{k-1}(t)),$$

where $h^i(t)$ is the ith element of $h(t)$. By performing similar operations on the remaining vectors we deduce

$$\det(M_k(t)) = \sum_{i_1=1}^{k} \dots \sum_{i_k=1}^{k} h^{i_1}(t) v_2^{i_2}(t) \dots v_{k-1}^{i_k}(t) E_k, \tag{5.20}$$

where $E_k := (e^{i_1} \wedge e^{i_2} \wedge \dots \wedge e^{i_k})$. By (5.17), we have

$$|h^i(t)| \leqslant c_h e^{-\alpha t}, \quad |v^i(t)| \leqslant c_v e^{\varphi t}.$$

Moreover, the factor E_k will be either zero or an element of the canonical basis in $\mathbb{R}^{n_x}$ multiplied by plus or minus one. Thus, using the triangle inequality, one obtains

$$|\det(M_k(t))| \leqslant \kappa c_h c_v e^{(-\alpha+(k-1)\varphi)t}$$

where $\kappa > 0$ is a positive constant related to the number of non-zero instances of E_k. Now, since $\alpha - (k-1)\varphi > 0$ by assumption, by continuity there always exists $\varepsilon > 0$ such that $\alpha - (k-1)\varphi - \varepsilon > 0$. Then,

$$|M(t)^{(k)}| = |e^{-\varepsilon t} e^{\varepsilon t} M(t)^{(k)}| \leqslant e^{-\varepsilon t} |e^{\varepsilon t} M(t)^{(k)}|.$$

By considering the worst-case (5.19), we have

$$e^{\varepsilon t} |\det(M_k(t))| \leqslant \bar{c} e^{(-\alpha+(k-1)\varphi+\varepsilon)t},$$

for some $\bar{c} > 0$. Hence, since $\alpha-(k-1)\varphi-\varepsilon > 0$, each element of $e^{\varepsilon t} M(t)^{(k)}$ is exponentially decreasing and the norm $|e^{\varepsilon t} M(t)^{(k)}|$ is uniformly bounded for all $t \in \mathbb{R}_{\geqslant 0}$, thus concluding the proof. □

Leveraging on the previous lemmas, we provide a bound on the k multiplicative compound of the state transition matrix of the lifted system (5.7).

Lemma 5.4. *Consider system* (5.11) *and assume there exist a forward invariant compact set* $\mathcal{A} \subset \mathbb{R}^{n_x}$*, constants* μ_0, μ_{k-1} *and matrices* $P_0, P_{k-1} \in \mathbb{R}^{n_x \times n_x}$ *such that*

$$\frac{\partial f}{\partial x}(x)^\top P_0 + P_0 \frac{\partial f}{\partial x}(x) \prec 2\mu_0 P_0, \tag{5.21a}$$

$$\frac{\partial f}{\partial x}(x)^\top P_{k-1} + P_{k-1} \frac{\partial f}{\partial x}(x) \prec 2\mu_{k-1} P_{k-1} \tag{5.21b}$$

$$\mu_{k-1} + (k-1)\mu_0 < 0 \tag{5.21c}$$

hold for all $x \in \mathcal{A}$*. Then, there exist* $\varepsilon, c > 0$ *such that*

$$\left| \frac{\partial \psi^t}{\partial x}(x)^{(k)} \right| \leqslant c e^{-\varepsilon t}, \quad \forall (t, x) \in \mathbb{R}_{\geqslant 0} \times \mathcal{A}. \tag{5.22}$$

Proof. Consider (5.13a) in Theorem 5.2. Invertibility of $\mathbf{T}(x)$ yields

$$\frac{\partial \psi^t}{\partial x}(x) = \frac{\partial \psi^t}{\partial x}(x)\mathbf{T}(x)\mathbf{T}(x)^{-1} = \Psi^t(x)\mathbf{T}(x)^{-1},$$

with $\Psi^t(x) := \left[\frac{\partial \psi^t}{\partial x}(x)\mathbf{T_h}(x) \quad \frac{\partial \psi^t}{\partial x}(x)\mathbf{T_v}(x) \right]$. Given any $\delta_x \in T_x\mathbb{R}^{n_x}$, consider the decomposition $\delta_x = (\delta_x^h, \delta_x^v)$, where $\delta_x^h \in \mathbb{R}^{n_x - p}$ and $\delta_x^v \in \mathbb{R}^p$. Then, for an arbitrary δ_x^h, inequality (5.13c) of Theorem 5.2 implies

$$\left| \frac{\partial \psi^t}{\partial x}(x)\mathbf{T_h}(x)\delta_x^h \right| \leqslant c_h e^{-\mu_{k-1}} \left| \mathbf{T_h}(x)\delta_x^h \right|.$$

Recall the definition of matrix norm,

$$\left| \frac{\partial \psi^t}{\partial x}(x)\mathbf{T_h}(x) \right| := \max_{|u|=1} \left| \frac{\partial \psi^t}{\partial x}(x)\mathbf{T_h}(x)u \right|.$$

By selecting vector $u^\star$ such that $|u^\star| = 1$, the previous exponential relation and the triangular inequality yield

$$\left| \frac{\partial \psi^t}{\partial x}(x)\mathbf{T_h}(x) \right| = \left| \frac{\partial \psi^t}{\partial x}(x)\mathbf{T_h}(x)u^\star \right| \leqslant c_h e^{-\mu_{k-1}} |\mathbf{T_h}(x)u^\star| \leqslant c_h e^{-\mu_{k-1}} |\mathbf{T_h}(x)|.$$

Since $\mathcal{A}$ is compact and $\mathbf{T}$ is continuous, $|\mathbf{T_h}(x)|$ is bounded for all $x \in \mathcal{A}$. Then, by (5.13c), and by (5.15) we obtain

$$\left| \frac{\partial \psi^t}{\partial x}(x)\mathbf{T_h}(x) \right| \leqslant c_h e^{-\mu_{k-1}} |\mathbf{T_h}(x)| \leqslant \bar{c}_h e^{-\mu_{k-1}}, \quad \left| \frac{\partial \psi^t}{\partial x}(x)\mathbf{T_v}(x) \right| \leqslant c_v e^{\mu_0} |\mathbf{T_v}(x)| \leqslant \bar{c}_v e^{\mu_0}$$

for all $x \in \mathcal{A}$. Finally, by boundedness of $\mathbf{T}(x)$ and Lemma 5.3, we obtain

$$\left|\frac{\partial \psi}{\partial x}^{t}(x)^{(k)}\right| \leqslant |\Psi^t(x)^{(k)}||\mathbf{T}(x)^{-1^{(k)}}| \leqslant ce^{-\varepsilon t}$$

for all $x \in \mathcal{A}$, concluding the proof. □

Thanks to all the above results, we are finally ready to present the main result of this section, namely, a set of sufficient conditions for k-contraction which do not involve matrix compounds.

Theorem 5.3. *Let $\mathcal{A} \subset \mathbb{R}^{n_x}$ be a compact forward invariant set and assume there exist symmetric matrices $P_0, P_{k-1} \in \mathbb{R}^{n_x \times n_x}$ of inertia with respect to the imaginary axis $\mathrm{In_c}(P_0) = (0, 0, n)$, $\mathrm{In_c}(P_{k-1}) = (k-1, 0, n_x - k + 1)$ and $\mu_0, \mu_{k-1} \in \mathbb{R}$ such that*

$$\begin{gathered}\frac{\partial f}{\partial x}(x)^\top P_0 + P_0 \frac{\partial f}{\partial x}(x) \prec 2\mu_0 P_0,\\ \frac{\partial f}{\partial x}(x)^\top P_{k-1} + P_{k-1} \frac{\partial f}{\partial x}(x) \prec 2\mu_{k-1} P_{k-1}\\ \mu_{k-1} + (k-1)\mu_0 < 0\end{gathered}$$

hold for all $x \in \mathcal{A}$. Then, system (5.11) *is k-contractive on $\mathcal{S} := \mathcal{A}$.*

Proof. Consider the k^{th} multiplicative compound of matrix $X_{\mathrm{NL}}(t, x_0)$ defined as in (5.8). A simple computation shows

$$X_{\mathrm{NL}}(x_0, t)^{(k)} = \left[\tfrac{\partial \psi}{\partial x}^{t}(x_0)\delta^1_{x_0} \quad \dots \quad \tfrac{\partial \psi}{\partial x}^{t}(x_0)\delta^k_{x_0}\right]^{(k)} = \frac{\partial \psi}{\partial x}^{t}(x_0)^{(k)} X_{\mathrm{NL}}(x_0, 0)^{(k)},$$

where the second inequality is derived from the Cauchy-Binet formula [67, Chapter 1]. From (5.21) and Lemma 5.4 we obtain

$$\left|(X_{\mathrm{NL}}(x_0, t))^{(k)}\right| \leqslant ce^{-\varepsilon t}\left|(X_{\mathrm{NL}}(x_0, 0))^{(k)}\right|.$$

Hence, the system is infinitesimally k-contractive and the result follows by Lemma 5.1. □

It is natural to see the similarities between matrix inequalities (5.12) and (5.21). Indeed condition (5.21b) is imposing p-dominance with a specific contraction rate, defined by (5.21c). Hence, this condition sheds light on the relationship between k-contraction and p-dominance. This relationship between p-dominance and k-contraction explains why both properties share similar convergence results for systems evolving in a bounded set. Consider system (5.11) and assume $\mathcal{S}$ is compact and forward invariant. In [124] it is shown that any bounded solution converges to an equilibrium point if the system is 2-contractive. Similarly, in [75, Corollary 1], it is proven that any bounded solution converges to a fixed point if the system is 1-dominant.

5.2 k-contraction in LTI systems: a necessary and sufficient condition

The results of Section 5.1.2 are encouraging, even if we had to restrict the analysis to forward invariant compact sets and time-invariant systems in order to exploit the findings of Theorem 5.2. However, the conditions in Theorem 5.3 are only sufficient. As a matter of facts, they can be

seen as imposing the slowest rate of convergence among the convergence directions to be $k-1$ times faster then the fastest exploding rate. This is clearly a conservative result, as ideally we would like to perfectly dominate the sum of exploding rates. Unfortunately, this turns out to be a difficult task in nonlinear systems. However, for linear systems we can exploit the relation between eigenvalues and convergence rates. As a matter of fact, this relation allows us to separately consider each unstable direction and, consequently, provide more refined conditions. Then, in what follows, we will focus on linear time-invariant system dynamics. Consider a linear system of the form

$$\dot{x} = Ax, \qquad x \in \mathbb{R}^{n_x}. \tag{5.24}$$

Recall that a spectral property of the additive compound matrix is that the eigenvalues of the matrix $A^{[k]}$ are all the possible sums of the form $\lambda_{i_1} + \lambda_{i_2} + \cdots + \lambda_{i_k}$, with $1 \leqslant i_1 \leqslant \ldots \leqslant i_k \leqslant n_x$, see [236]. Moreover, $A^{[k]}$ defines the continuous-time dynamics of volumes in the linear framework, see (5.10). Thus, a necessary and sufficient condition for k-contraction is

$$\sum_{i=1}^{k} \lambda_i < 0. \tag{5.25}$$

We now provide a set of sufficient and necessary conditions to establish the k-contractivity property of (5.24) which do not involve matrix compounds. First, we recall a technical result which will be instrumental for showing and understanding the main result of the section. This lemma can be seen as the linear counterpart of Theorem 5.2. The result can be found, e.g., in [75, Proposition 1], [121, Section 13.1], [201, Lemma 1, Section 3].

Lemma 5.5. *Assume there exists a symmetric matrix $P \in \mathbb{R}^{n_x \times n_x}$ with $\mathrm{In_c}(P) = (p, 0, n_x - p)$ and a constant μ such that*

$$A^\top P + PA \prec 2\mu P\,. \tag{5.26}$$

Then, matrix A has p eigenvalues with real part strictly bigger than μ and $n_x - p$ eigenvalues real part strictly smaller or equal than μ.

We can now move to the main result. We start by considering the case of a matrix A having only distinct real eigenvalues. This simplifies the statement and its readability. The general case will be given as a corollary, whose proof will trivially follow given the one for the distinct eigenvalue scenario.

Proposition 5.1. *Assume that A has only distinct real eigenvalues. Then, system* (5.24) *is k-contractive on $\mathcal{S} := \mathbb{R}^{n_x}$ if and only if there exist a set of symmetric matrices $P_i \in \mathbb{R}^{n_x \times n_x}$, $i = 0, \ldots, k-1$, with $\mathrm{In_c}(P_i) = (i, 0, n_x - i)$, and a set of real numbers $\mu_i \in \mathbb{R}$, $i = 0, \ldots, k-1$, such that*

$$A^\top P_i + P_i A \prec 2\mu_i P_i \qquad \forall i = 0, \ldots, k-1\,, \tag{5.27a}$$

$$\sum_{i=0}^{k-1} \mu_i < 0\,. \tag{5.27b}$$

Proof. As per (5.25), a necessary and sufficient condition for linear k-contraction is the stability of $A^{[k]}$. Then, the remainder of the proof is based on showing that (5.27a)- (5.27b) in Proposition 5.1 are equivalent to $A^{[k]}$ being Hurwitz.
Sufficiency. Let the eigenvalues of A be ordered such that $\lambda_1 > \lambda_2 > \cdots > \lambda_{n_x}$. By Lemma 5.5, inequality (5.27a) implies $\lambda_{i+1} < \mu_i$ for all $i = 0, \ldots, k-1$. Then, by (5.27b) and since

the eigenvalues are scalars, we have

$$\sum_{i=1}^{k} \lambda_i = \sum_{i=1}^{k} \lambda_i < \sum_{i=0}^{k-1} \mu_i < 0\,,$$

and (5.25) is satisfied.

Necessity. As stated in the previous step of the proof, if $A^{[k]}$ is Hurwitz, (5.25) is verified. Hence, by continuity, there exists a set of scalars $\varepsilon_i > 0$ such that $\sum_{i=1}^{k}(\varepsilon_i + \lambda_i) < 0$. Select $\mu_{i-1} = \varepsilon_i + \lambda_i > \lambda_i$, for $i = 1, \ldots, k-1$. We have

$$\sum_{i=0}^{k-1} \mu_i = \sum_{i=1}^{k} (\varepsilon_i + \lambda_i) < 0.$$

Now, define matrices $\hat{A}_i = A - \mu_i I$ with $i = 0, \ldots, k-1$. It is clear that, by the definition of μ_i, each matrix $\hat{A}_i$ has i negative eigenvalues and $n_x - i$ positive eigenvalues. Then, by [208, Theorem 2.5], there exist symmetric matrices P_i such that

$$\hat{A}^\top P_i + P_i \hat{A} = -G_i \qquad \forall i = 0, \ldots, k-1\,,$$

with $G_i \succ 0$ and $\mathrm{In_c}(P_i) = \mathrm{In_c}(\hat{A}_i) = \{i, 0, n_x - i\}$. Then, as $G_i \succ 0$ and by using the definition of $\hat{A}_i$, we have

$$A^\top P_i + P_i A \prec 2\mu_i P_i \qquad \forall i = 0, \ldots, k-1\,,$$

thus concluding the proof. □

In the linear case, the previous interpretation of inequalities (5.21) bounding expansion and contraction rates directly maps to μ_i bounding the k largest eigenvalues of matrix A. Then, condition (5.27b) states that the sum of the k largest eigenvalues of A is negative.

For the general case of arbitrary eigenvalues (possibly complex conjugate), the previous result needs to be slightly modified. More specifically, some of the inequalities (5.27a) cannot be satisfied in the case of multiple eigenvalues having identical real parts. Hence, some conditions need to be skipped. To this end, consider the matrix A in (5.24) and let us order the eigenvalues of A as $\mathfrak{Re}(\lambda_1) \geqslant \mathfrak{Re}(\lambda_2) \geqslant \ldots \geqslant \mathfrak{Re}(\lambda_{n_x})$. We define

$$\Delta_k := \{i \in \{1, \ldots, k-1\} \mid \mathfrak{Re}(\lambda_i) > \mathfrak{Re}(\lambda_{i+1})\} \cup \{0\},$$

and we denote by d_j an element at position $j \in [1, \#\Delta_k]$:

$$\Delta_k := \{d_1, d_2, \ldots, d_{\#\Delta_k}\}\,, \quad d_1 < d_2 < \cdots < d_{\#\Delta_k},$$

with $\#\Delta_k$ being the number of elements of Δ_k. In simple words, Δ_k contains the orders of p-dominance that could be satisfied by A given a desired contraction order k.

Theorem 5.4. *Let $p = \max \Delta_k$. System (5.24) is k-contractive if and only if there exists a set of symmetric matrices $P_i \in \mathbb{R}^{n_x \times n_x}$, $i \in \Delta_k$ with inertia with respect to the imaginary axis $\mathrm{In_c}(P_i) =$*

$(i, 0, n_x - i)$ and a set of positive constants μ_i such that

$$A^\top P_i + P_i A \prec 2\mu_i P \qquad \forall i \in \Delta_k,$$
$$(k-p)\mu_p + \sum_{i \in \Delta_k \setminus \{p\}} h_i \, \mu_i < 0 \,,$$

where $h_i = d_{j+1} - d_j$ and j is an index such that $d_j = i$.

The proof of Theorem 5.4 is omitted as it follows from the one of Proposition 5.3. Intuitively, Theorem 5.4 is a rephrasing of Theorem 5.3 where inequalities corresponding to eigenvalues having the same real part are merged. To improve clarity, we propose an example. Consider system (5.24) with $n_x = 6$ and eigenvalues satisfying $\mathfrak{Re}(\lambda_1) = \mathfrak{Re}(\lambda_2) > \mathfrak{Re}(\lambda_3) > \mathfrak{Re}(\lambda_4) = \mathfrak{Re}(\lambda_5) > \mathfrak{Re}(\lambda_6)$. Consider now the conditions for 6-contraction. We have $\Delta_6 = \{0, 2, 3, 5\}$, $\#\Delta_k = 4$. Then, $p = 5, (d_1, d_2, d_3, d_4) = (0, 2, 3, 5)$ and $h_0 = d_2 - d_1 = 2, h_2 = d_3 - d_2 = 1, h_3 = d_4 - d_3 = 2$. Consequently, conditions in Theorem 5.4 read as

$$A^\top P_i + P_i A \prec 2\mu_i P_i, \; \forall \, i = 0, 2, 3, 5, \quad 2\mu_0 + \mu_2 + 2\mu_3 + \mu_5 < 0.$$

These conditions consider the fact that $\mathfrak{Re}(\lambda_1) = \mathfrak{Re}(\lambda_2)$ and $\mathfrak{Re}(\lambda_4) = \mathfrak{Re}(\lambda_5)$. Indeed, these eigenvalues are covered jointly by P_2 (resp. P_4), which has inertia with respect to the imaginary axis $(2, 0, 4)$ (resp. $(4, 0, 2)$). Similarly, μ_0 (resp. μ_3) is counted twice, as it upper-bounds the real part of both λ_1 (resp. λ_4) and λ_2 (resp. λ_5). Consider now the conditions for 2-contraction. Then, $p = 0$ and we obtain $A^\top P_0 + AP \prec 2\mu_0 P_0$, $\mu_0 < 0$. These conditions are identical to the ones obtained when $k = 1$. As a matter of fact, a sufficient condition for k-contractivity is $(k-1)$-contractivity. However, given the eigenvalues of A, the system cannot be 2-contractive without being 1-contractive. Hence, the condition becomes necessary.

5.2.1 Comparison with existing results

In this section, we elaborate further the relationship between our conditions and the ones of existing works in k-contraction analysis that exploit matrix compounds [151, 154, 236]. In particular, we highlight the main advantages of our result when compared to matrix compound methods, e.g. [236] and [54]. Moreover, we motivate why conditions as in Theorem 5.3 and Theorem 5.4 are more suited to feedback design. Finally, we compare results in Proposition 5.1 to other recent ones avoiding matrix compounds [54].

We start by comparing computational complexity of Theorem 5.1 to the one of Proposition 5.1. Our aim is to show that our results provide a set of conditions which is solvable by optimization tools for a wide range of contraction order k and system dimension n_x. We focus on the linear system framework, as Proposition 5.1 provides a larger set of matrix inequalities with respect to Theorem 5.3. We recall that sufficient conditions could be obtained by relaxing Proposition 5.1 to require only two conditions to be satisfied, as in Theorem 5.3. Moreover, we select Proposition 5.1 since Theorem 5.4 requires fewer conditions to be satisfied (as $\#\Delta_k \leqslant k$). First, notice that the inertia constraint in our condition can be relaxed to obtain an unconstrained LMI, see [75, Section VI.B]. Hence, as both results (Theorem 5.1 in the linear framework and Proposition 5.1) can be expressed in terms of unconstrained LMIs, we compare the number of variables to be estimated by each. Let $M \in \mathbb{R}^{r \times r}$ be an arbitrary square matrix and $Q \in \mathbb{R}^{r \times r}$ be a symmetric matrix. Since Q is symmetric, each condition of the form $QM + M^\top Q \preceq \mu Q$ requires the computation of $N = r(r-1)/2 + 1$ variables, namely the entries of the top triangular portion of Q and the scalar μ. Then, Theorem 5.1 requires $N_1 = \binom{n_x}{k}\left(\binom{n_x}{k} - 1\right)/2 + 1$ variables

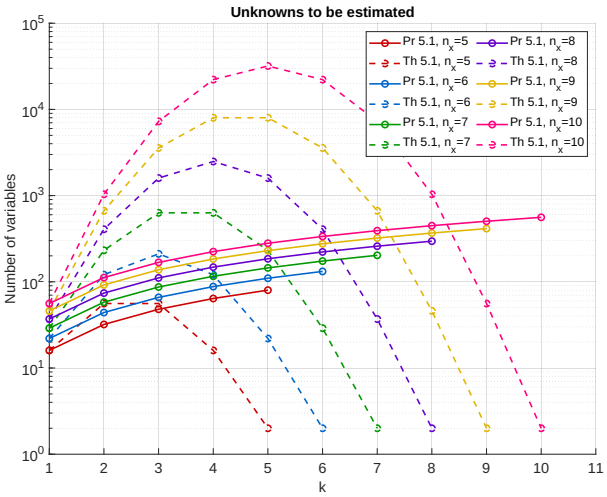


Figure 5.2: Number of variables to be estimated Proposition 5.1 (solid) and by Theorem 5.1 (dashed) in function of k. Different colors refer to different state dimensions n_x.

while Proposition 5.1 requires $N_2 = kn_x(n_x - 1)/2 + k$ variables. To better understand how the number of variables scales with different values of k and n_x, see Fig. 5.2. Clearly, for large dimensional systems and low k, the condition in (5.21) is of significantly smaller computational complexity. Moreover, even in the worst case of $k = n_x$, Proposition 5.1 typically requires between 10^2 and 10^3 variables. Differently, Theorem 5.1 can easily reach 10^4 variables in the worst case.

Now we compare the results in terms of feedback design. We claim that the lack of matrix compounds in Theorem 5.3 and Proposition 5.1 simplifies the process of k-contractive feedback design. Consider a linear system of the form

$$\dot{x} = Ax + Bu,$$

where $u \in \mathbb{R}^{n_u}$ is the control input. Assume we want to design a state-feedback controller of the form $u = -Kx$, with K a constant matrix of adequate dimension and such that the closed-loop system is k-contractive. On one hand, Theorem 5.1 reduces to designing K such that condition (5.4) is satisfied for the closed-loop system. That is,

$$Q\left((A - BK)^{[k]}\right) + \left((A - BK)^{[k]}\right)^{\top} Q \preceq -\mu Q.$$

However, this is a highly nonlinear and non-convex matrix inequality. Consequently, a design methodology for K cannot be straightforwardly derived. On the other hand, also Proposition 5.1 asks for conditions (5.21) to be verified by the closed-loop system. However, this can be transformed to a set of linear matrix inequalities by means of standard transformations [185]. For this reason, we believe that condition (5.21) will be crucial in the development of k-contraction design tools.

Finally, keeping our focus on the linear case, we compare to previous works that already investigated sufficient conditions for k-contraction that do not require the computation of the k-additive compound. As a matter of fact, [54, Theorem 17] shows that a system is k-contractive if there exist an invertible matrix $T \in \mathbb{R}^{n_x \times n_x}$ and $q \in \{1, 2, \infty\}$ such that

$$\mathrm{tr}(A) + (n_x - k)\mu_q(-TAT^{-1}) < 0 \tag{5.28}$$

where $\mu_q(\cdot)$ represents the logarithmic norm of a matrix, see e.g. [3,207], and $\mathrm{tr}(A)$ denotes the trace of the matrix A. We highlight that Proposition 5.1 provides necessary and sufficient conditions while the condition (5.28) is only sufficient. This is made evident by the first example presented in [54, Section V], where the authors consider a diagonal matrix $A = \mathrm{diag}(\lambda_1, \dots, \lambda_{n_x})$ satisfying $\mathfrak{Re}(\lambda_1) \geqslant \dots \geqslant \mathfrak{Re}(\lambda_{n_x})$. Then, for any invertible matrix T and any $q \in \{1, 2, \infty\}$, the left-hand side in (5.28) reads as

$$\mathrm{tr}(A) + (n_x - k)\mu_q(-TAT^{-1}) = -(n_x - k - 1)\lambda_n + \lambda_1 + \dots + \lambda_{n_x - 1}.$$

Thus, condition (5.28) reduces to

$$\lambda_1 + \dots + \lambda_{n_x - 1} < (n_x - k - 1)\lambda_{n_x} < 0. \tag{5.29}$$

Given the spectral property of the additive compound matrix, we recall that (5.25) is a necessary and sufficient condition for k-contraction. For $k = n_x - 1$, equation (5.29) reduces to (5.25). Consequently, (5.29) is necessary and sufficient for $(n_x - 1)$-contraction. However, for $n_x - k > 1$, it is always possible to fix a sufficiently negative eigenvalue λ_{n_x} such that (5.29) is not satisfied even if (5.25) is satisfied. Thus, condition (5.28) is sufficient but not necessary.

5.2.2 The discrete-time case

Unfortunately, k-contraction conditions similar to the ones presented in Section 5.1.2 are still missing in the discrete-time nonlinear scenario. This is due a lack of a parallel to Theorem 5.2 in such a framework. However, the linear case can leverage on eigenvalue arguments. Then, we now propose necessary and sufficient conditions for k-contraction for linear time invariant discrete-time systems equivalent to the ones in Proposition 5.1. We consider a discrete-time linear system evolving according to the difference equation

$$x^+ = Ax, \qquad x \in \mathbb{R}^{n_x}. \tag{5.30}$$

We start by recalling the concept of inertia of a matrix with respect to the unit circle.

Definition 5.6 (Matrix inertia wrt the unit circle [208])**.** *The inertia of a matrix P with respect to the unit circle is defined by the triplet of integers $\mathrm{In_d}(P) := (\pi_{>1}(P), \pi_1(P), \pi_{<1}(P))$, where $\pi_{<1}(P)$, $\pi_{>1}(P)$ and $\pi_1(P)$ denote the numbers of eigenvalues of P inside, outside and on the unit circle, resp., counting multiplicities.*

Then, before presenting our result for the linear discrete-time framework, we also recall the discrete-time parallel of Lemma 5.5, see e.g. [208, Theorem 3.4], [121, Section 13.2].

Lemma 5.6. *Assume there exists a symmetric matrix P with inertia with respect to the imaginary axis $\mathrm{In_c}(P) = (p, 0, n_x - p)$ and a positive constant μ such that*

$$A^\top P A - \mu^2 P \prec 0\,. \tag{5.31}$$

Then, matrix A has p eigenvalues with module strictly bigger than μ and $n_x - p$ eigenvalues with module smaller than μ. In other words, $\mathrm{In_d}(\mu^{-1}A) = \mathrm{In_c}(P)$.

We are now ready to state the discrete-time result of this section. Similarly to Proposition 5.1 we first focus on the simplified case of state matrices A in (5.30) with eigenvalues of different norms. This simplifies the analysis and intuition. Then, we generalize the result to arbitrary state matrices.

Proposition 5.2. *Assume A has no eigenvalue with the same norm. Then, system system* (5.30) *is k-contractive on $\mathcal{S} := \mathbb{R}^{n_x}$ if and only if there exist a set of symmetric matrices $P_i \in \mathbb{R}^{n_x \times n_x}$, $i = 0, \dots, k-1$ with inertia $\mathrm{In_c}(P_i) = (i, 0, n_x - i)$ and a set of positive constants $\mu_i, i = 0, \dots, k-1$ such that*

$$A^\top P_i A - \mu_i^2 P_i \prec 0 \qquad \forall i = 0, \dots, k-1\,, \tag{5.32a}$$

$$\prod_{i=0}^{k-1} \mu_i < 1\,. \tag{5.32b}$$

Proof. By the properties of the multiplicative compound, it is easy to show that $A^{(k)}$ defines the volume dynamics in the discrete-time linear framework, [236]. Define $\lambda_1, \dots, \lambda_{n_x} \in \mathbb{C}$ as the eigenvalues of the matrix A ordered according to their norm, that is

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_{n_x}|.$$

A spectral property of the multiplicative compound matrix is that the eigenvalues of the matrix $A^{(k)}$ are all the possible products of the form $\lambda_{i_1} \times \lambda_{i_2} \times \cdots \times \lambda_{i_k}$, with $1 \leqslant i_1 \leqslant \ldots \leqslant i_k \leqslant n_x$, [236]. This implies that a necessary and sufficient condition for k-contraction of discrete-time linear systems is

$$\left| \prod_{i=1}^{k} \lambda_i \right| < 1\,. \tag{5.33}$$

Sufficiency. By Lemma 5.6, inequality (5.32a) implies

$$|\lambda_{i+1}| < \mu_i, \qquad \forall i = 0, \dots, k-1.$$

Then, by (5.32b), we have

$$\left| \prod_{i=1}^{k} \lambda_i \right| = \prod_{i=1}^{k} |\lambda_i| < \prod_{i=0}^{k-1} \mu_i < 1\,.$$

Necessity. If $A^{(k)}$ is Schur stable, inequality (5.33) is verified. Hence, by continuity, there exist scalars $\varepsilon_i > 1$ such that

$$\left| \prod_{i=1}^{k} \varepsilon_i \lambda_i \right| < 1\,.$$

Select

$$\mu_{i-1} = \varepsilon_i |\lambda_i| > |\lambda_i|, \qquad i = 1, \dots, k-1.$$

We have

$$\prod_{i=0}^{k-1} \mu_i = \prod_{i=1}^{k} \varepsilon_i |\lambda_i| < 1.$$

Now, define matrices $\hat{A}_i = \frac{1}{\mu_i} A$ with $i = 0, \dots, k-1$. It is clear that, by the definition of μ_i, each matrix $\hat{A}_i$ has inertia with respect to the unit disc $\mathrm{In_d}(\hat{A}_i) = (n_x - i, 0, i)$ with respect to the unit circle. Then, by [208, Theorem 3.4], there exist symmetric matrices P_i such that

$$\hat{A}^\top P_i \hat{A} - P_i = -G_i \qquad \forall i = 0, \dots, k-1\,,$$

with G_i positive definite and P_i with inertia with respect to the imaginary axis $\mathrm{In_c}(P_i) = (i, 0, n_x - i)$. Finally, as $G_i \succ 0$ and by using the definition of $\hat{A}_i$ we have

$$A^\top P_i A - \mu_i^2 P_i \prec 0 \qquad \forall i = 0, \dots, k-1\,,$$

thus concluding the proof. □

In Proposition 5.2 we assumed eigenvalues to have different norms to simplify the readability of equations (5.32a), (5.32b) and ease the understanding of the concepts in the proof. However, similar to the continuous-time case, some conditions may not be verifiable in the case of eigenvalues having the same norm. This is clarified in the following example. Consider system (5.30) with state matrix $A \in \mathbb{R}^{6\times 6}$ having eigenvalues such that

$$|\lambda_1| = |\lambda_2| > |\lambda_3| > |\lambda_4| = |\lambda_5| = |\lambda_6|. \tag{5.34}$$

Consider now (5.32a) with $i = 1$, namely

$$A^\top P_1 A - \mu_1^2 P_1 \prec 0\,,$$

with P_1 having inertia with respect to the imaginary axis $\mathrm{In_c}(P_1) = (1, 0, n_x - 1)$. Since $|\lambda_1| = |\lambda_2|$, we have $\mu_1 > |\lambda_1|$. Then the inertia with respect to the unit disc of $\hat{A}_1 = \frac{1}{\mu_1} A$ is equal to the inertia with respect to the unit disc of $\hat{A}_0 = \frac{1}{\mu_0} A$. In turn, by [208, Theorem 3.4], P_1 and P_0 must have the same inertia with respect to the imaginary axis, thus contradicting our assumption on the inertia of P_1. Therefore, paralleling Theorem 5.4, we propose the following slight modification to Proposition 5.2 to tackle the issue.

Consider a system of the form (5.30) where the eigenvalues of A satisfy $|\lambda_1| \geqslant |\lambda_2| \geqslant \ldots \geqslant |\lambda_{n_x}|$. We define

$$\Delta_k := \{i \in \{1, \dots, k-1\} \mid |\lambda_i| > |\lambda_{i+1}|\} \cup \{0\},$$

and we denote by d_j an element at position $j \in [1, \#\Delta_k]$:

$$\Delta_k := \{d_1, d_2, \dots, d_{\#\Delta_k}\}\,, \quad d_1 < d_2 < \cdots < d_{\#\Delta_k}\,,$$

with $\#\Delta_k$ being the number of elements of Δ_k. As per the continuous-time scenario, Δ_k selects the conditions (5.32a) that can be satisfied by A given a desired contraction order k.

Theorem 5.5. *Let $p = \max \Delta_k$. System (5.24) is k-contractive if and only if there exists a set of symmetric matrices $P_i \in \mathbb{R}^{n_x \times n_x}$, $i \in \Delta_k$ with inertia with respect to the imaginary axis $\mathrm{In_c}(P_i) = (i, 0, n_x - i)$ and a set of positive constants μ_i such that*

$$\begin{aligned} A^\top P_i A - \mu_i^2 P_i &\prec 0 \qquad \forall i \in \Delta_k, \\ \mu_p^{k-p} \prod_{i \in \Delta_k \setminus \{p\}} \mu_i^{h_i} &< 1\,, \end{aligned}$$

where $h_i = d_{j+1} - d_j$ and j is an index such that $d_j = i$.

The proof of Theorem 5.5 is omitted as it follows from the one of Proposition 5.2. To improve clarity, we go back to the previous example where A satisfies (5.34). Consider the conditions for 3-contraction. We have $\Delta_k = \{0, 2\}$. Then, $p = 2$ and $h_0 = 2$. Consequently, the conditions in Theorem 5.5 read as

$$\begin{aligned} &A^\top P_0 A - \mu_0^2 P_0 \prec 0, \\ &A^\top P_2 A - \mu_2^2 P_2 \prec 0, \\ &\mu_0^2 \mu_2 < 1. \end{aligned}$$

As in the continuous-time case, it is clear that these conditions consider the fact that $|\lambda_1| = |\lambda_2|$. These parallel conditions with respect to the continuous-time framework are promising for a possible future extension of the nonlinear results in the discrete-time scenario.

6 Conclusions and perspectives

In this thesis, we derived multiple novel results on stability and robustness of system dynamics, and we studied several methods to pair them with deep neural network approximations. Ranging from results on robust stabilization of equilibrium points in discrete-time to trajectory following in the continuous-time framework, we showed the potential of the interconnection between the increasingly overlapping fields of control theory and machine learning. Via the combination of strong theoretical results and extremely adaptable learning techniques, we derived practical methods to obtain fast and complex DNN-based controllers equipped with strong guarantees. Moreover, we provided theoretical foundations for future research that will necessitate expressive function approximators (such as DNNs) to reach their full application potential.

In Chapter 1, we focused on learning locally asymptotically stabilizing laws for discrete-time nonlinear systems. We introduced a novel approach involving the integration of locally stabilizing behaviors into DNN-based controllers. To this aim, we presented the formulation of a discounted learning objective that characterizes a closed-loop behavior aligned with the selected local controller. Additionally, we derived finite optimality bounds that leverage stability properties to mitigate the impact of the discount factor. We comprehensively tested the proposed methodology to validate its effectiveness.

In Chapter 2, we discussed the use of integral action in the discrete-time nonlinear domain. We introduced the concept of discrete-time total stability and we derived conditions for the transfer of stability properties between sufficiently similar systems, solely based on the norms of differences between models. We showed that integral action allows for robust regulation under constant signals in the discrete-time nonlinear settings, and we applied the findings to learned controllers. In particular, we showed the effectiveness of integral action for DNN-based controllers by learning a robust policy for plasma-shape control in tokamak reactors.

In Chapter 3, our attention turned to the investigation of discrete-time incremental stability properties arising from contraction analysis. We presented sufficient conditions for the design of controllers guaranteeing incremental exponential ISS properties of the closed-loop. Moreover, we proved optimality of such laws with respect to an incremental quadratic cost, thereby paving the way to future learning-based designs. The results were applied to the multi-agent setting, and we provided sufficient contraction-based conditions for robust synchronization of discrete-time nonlinear agents. Our findings were further validated through experimental testing.

In Chapter 4, we derived learning-based approaches to compute contractive controllers based on Riemannian metrics in the nonlinear continuous-time framework. We relaxed existing conditions to allow for DNN approximation errors, and we proposed an optimization problem to learn control laws guaranteeing contractivity of the closed-loop. We specialized the proposed methodology to the multi-agent synchronization and the output tracking problems, and we proposed a switching objective function to improve the controller performance. Finally, we experimentally validated the controllers' effectiveness by learning a synchronizing controller for Lorenz oscillators and an output tracking controller for a ball-and-beam system under chaotic references.

In Chapter 5, we proposed generalized Lyapunov conditions for k-contraction that do not rely on matrix compounds. In the linear setting, we derived necessary and sufficient conditions for both the continuous and discrete-time frameworks. In the nonlinear scenario, we obtained sufficient condition for k-contraction of continuous-time dynamics. Finally, we presented a thorough comparison between our findings and existing results, highlighting the advantages of our approach in eliminating the need for complex matrix compound computations.

There are multiple research directions that can be explored as a continuation of the works presented in this manuscript. In what follows, we enlist some of the most thrilling ones, from both the practical and the theoretical sides.

Total stability in sampled-data or hybrid systems. With the introduction of total stability results in the discrete-time framework, Chapter 2 opens the possibility of exploring scenarios where continuous-time and discrete-time analysis overlap. This is the case of sampled-data and hybrid systems. In the former case, it may be interesting to compare conditions arising from the discretized continuous framework to the ones proposed in Chapter 2. In the latter scenario, an interesting opportunity lies in the formulation of hybrid total stability conditions, possibly leading to the derivation of results on robust output regulation, similar to the continuous and discrete-time framework.

Improving DRL generalization via internal models. Chapter 2 and Chapter 1 offer valuable insights in the generalization properties offered by the combination of deep reinforcement learning and output regulation theory. However, the results presented in this thesis are limited to the context of output regulation in presence of constant references. An interesting direction is the generalization of this concept to any periodic reference trajectory. By relying on results of output regulation based on internal model [103], one can envision learning optimal guaranteed asymptotically tracking controllers trained with model-free deep reinforcement learning algorithms.

DRL for incrementally stabilizing feedbacks. As shown in Chapter 3, discrete-time contractive controllers are solutions to an infinite-horizon optimization problem under quadratic costs. This interesting finding suggests that such controllers can be learned in a data-driven fashion. However, a first fundamental question is how close suboptimal policies have to be to an optimal solution (in value function difference terms) in order to preserve the guaranteed stabilizing properties. Another thrilling open question is whether the synchronizing solution proposed in Section 3.2 is also optimal. Finally, an exciting direction is the derivation of deep reinforcement learning approaches to learn such controllers.

Riemannian metric-based conditions for discrete-time incrementally stabilizing feedbacks. The developments in Chapter 3 are limited to Euclidean metrics. This limits the classes of nonlinear systems that can satisfy the required conditions. As a matter of fact, discrete-time contraction analysis based on Riemannian metrics is still limited to the autonomous framework [109]. The main challenge lies in the definition of a discrete-time parallel to the Killing vector field condition. Then, an intriguing research direction is the study of discrete-time feedback designs based on Riemannian metrics, paralleling the continuous-time framework. These results would also allow the development of DNN-based algorithms for learning such metrics and controllers also in the discrete-time nonlinear scenario.

Feedback design for k-contraction. Chapter 5 is a preliminary step for the design of k-contractive feedback laws. These controllers may provide reliable behaviors in systems for which classical stability theory cannot be used. As an example, 2-contractive dynamics evolving in bounded sets converge to an equilibrium point and cannot present limit cycles. Then, future research will focus on the derivation of such controllers and observers [S1]. Moreover, we believe that this investigation can lead to novel insights on the connection between different tools for the analysis of partial stability, and it will involve the definition of generalized concepts of the classical notions of stabilizability.

Discrete-time k-contraction. The results of k-contraction in the nonlinear framework have been derived only for the continuous-time scenario. However, it may be useful to translate the nonlinear results of Chapter 5 in discrete time. The motivation lies in the fact properties and asymptotic behaviors of discrete-time k-contractive systems are still unclear. By investigating this translation, we can provide useful insights in the analysis of such systems. Moreover, most learning approaches typically work in discrete-time scenarios. Then, the analysis may provide novel techniques for obtaining learning-based closed-loop systems equipped with valuable guarantees. Moreover, it is known that optimization algorithms can be interpreted as discrete-time dynamical systems. Then, similarly to [56], these new contraction analysis tools could be used for studying convergence properties of the learning algorithms themselves.

Appendices

A Some theoretical background

A.1 Deep reinforcement learning concepts

In this subsection, we briefly introduce some fundamental concepts of DRL and, more generally, reinforcement learning. Typically, in the reinforcement learning framework, the learning agent goal is to learn a mapping from state to actions generating a sequence of optimal inputs, called *control policy*. Through the interaction with the *environment*, it observes the current state and selects an *action*. In response, it receives a *reward* (which may be either positive or negative). Based on such a reward, it adjusts its policy, guided by an optimization objective. Then, the loop starts again and the agent observes the next state in order to select the next action. The control policy $\boldsymbol{\pi}$ can be deterministic ($\boldsymbol{\pi}(x)$ is a deterministic action) or stochastic ($\boldsymbol{\pi}(u|x)$ is a probability of action u given x, i.e. $\boldsymbol{\pi}(x)$ is a probability distribution). A stochastic policy draws actions from a random distribution, whose state-dependent momenta are learned by the agent. Stochastic policies are useful in adversarial contexts, where multiple agents compete with each others. As a matter of fact, deterministic policies can be proved to be optimal only in cooperative scenarios [174, Section 6.2.4]. In the framework of this manuscript, randomness of the policy will be useful only in the training context for exploring different state-action pairs and improve the policy. At test time, without loss of generality, we consider deterministic ones. Moreover, we focus on the common framework of discrete-time, time-invariant systems under infinite-horizon discounted objective functions.

In order to provide an intuition on the learning mechanisms behind DRL algorithms, we first introduce some concepts of Dynamic Programming (DP), whose theory is the backbone of modern DRL. In the discrete-time *deterministic* DP framework, the reinforcement learning problem is mapped to the problem of dealing with discrete-time dynamical systems (environments) generating a sequence of states $(x(\mathfrak{t}))_{\mathfrak{t}\in\mathbb{N}}$, $x \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$, under the influence of control inputs $(u(\mathfrak{t}))_{\mathfrak{t}\in\mathbb{N}}$, $u \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$ (actions). As previously stated, the objective is to find a mapping from states to inputs, the control policy $\boldsymbol{\pi}$, generating a control sequence leading to the maximum (or minimum) of a γ-discounted function (also known as *state-value function* under $\boldsymbol{\pi}$), $\mathbf{J}_{\boldsymbol{\pi}} : \mathcal{X} \to \mathbb{R}$, defined as $\mathbf{J}_{\boldsymbol{\pi}}(x(\mathfrak{t})) = \sum_{k=0}^{\infty} \gamma^k \mathbf{r}(x(k+\mathfrak{t}), \boldsymbol{\pi}(x(k+\mathfrak{t})))$, where $\gamma \in (0,1)$ and $\mathbf{r} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ is the instantaneous reward function, see, e.g., [28]. The state-value function under an optimal policy is the *optimal* state-value function $\mathbf{J}^\star = \max_{\boldsymbol{\pi}} \mathbf{J}_{\boldsymbol{\pi}}$, i.e. the one that provides the maximum sum of future rewards for each state.

However, typical reinforcement learning methods address a stochastic scenario, where dynamics are modeled by Markov Decision Processes (MDPs) [210]. Then, the *state-value function* $\mathbf{J}_{\boldsymbol{\pi}}$ corresponds to the expected total discounted reward starting from state $x(\mathrm{t})$ and then following the policy $\boldsymbol{\pi}$, namely

$$\mathbf{J}_{\boldsymbol{\pi}}(x(\mathrm{t})) = \mathbb{E}\left[\mathbf{r}(x(\mathrm{t}),u(\mathrm{t})) + \sum_{k=1}^{\infty}\gamma^k \mathbf{r}(x(k+\mathrm{t}),u(k+\mathrm{t}))\Big| x(\mathrm{t}), u(\cdot)\sim\boldsymbol{\pi}(x(\cdot))\right], \tag{A.1}$$

where $\mathbb{E}[\cdot \mid \cdot]$ stands for the conditional expectation. We remark that the value function profoundly depends on the policy $\boldsymbol{\pi}$. Given a policy $\boldsymbol{\pi}$ and an initial state $x(\mathrm{t})$, the value function is given. However, the optimal state-value function $\mathbf{J}^\star$ is independent from the policy, due to the presence of a maximization operation over the policies. In other words, the optimal policies may not be unique. Indeed, in light of the above discussion, it is clear that an optimal policy $\boldsymbol{\pi}^\star$ is a policy such that $\mathbf{J}_{\boldsymbol{\pi}^\star}(x) = \mathbf{J}^\star(x)$ for all $x \in \mathcal{X}$.

In the reinforcement learning framework, the *action-value function*, or *Q-function*, $\mathbf{Q}_{\boldsymbol{\pi}} : \mathcal{X} \times \mathcal{U} \to \mathbb{R}$, turns out to be more practical when dealing with unknown dynamics. The Q-function corresponds to the expected total discounted reward when the action $u(\mathrm{t})$ is taken in state $x(\mathrm{t})$, and then the policy $\boldsymbol{\pi}$ is followed from there on. Therefore, the Q-function is given by

$$\mathbf{Q}_{\boldsymbol{\pi}}(x(\mathrm{t}),u(\mathrm{t})) = \mathbb{E}\left[\mathbf{r}(x(\mathrm{t}),u(\mathrm{t})) + \sum_{k=1}^{\infty}\gamma^k \mathbf{r}(x(k+\mathrm{t}),u(k+\mathrm{t}))\Big| x(\mathrm{t}), u(\mathrm{t}), u(k+\mathrm{t})\sim\boldsymbol{\pi}(x(k+\mathrm{t}))\right].$$

As for the optimal action-value function, the optimal Q-function is given by

$$\mathbf{Q}^\star(x,u) = \max_{\boldsymbol{\pi}} \mathbf{Q}_{\boldsymbol{\pi}}(x,u), \qquad \forall (x,u) \in \mathcal{X}\times\mathcal{U},$$

and stands for the expected total discounted reward when the agent picks possible non-optimal action $u(\mathrm{t})$ in $x(\mathrm{t})$, and then behaves optimally from there on. The relation between the optimal action-value function and optimal state-value function directly comes from Bellman's principle of optimality [23] and it is expressed by

$$\mathbf{J}^\star(x) = \max_{u\in\mathcal{U}} \mathbf{Q}^\star(x,u), \qquad \forall (x,u) \in \mathcal{X}\times\mathcal{U}.$$

Then, if the optimal Q-function is known, the optimal action $u^\star$ for each state x can be extracted by choosing the action u that maximizes $\mathbf{Q}^\star(x,u)$, namely

$$u^\star = \arg\max_{u\in\mathcal{U}} \mathbf{Q}^\star(x,u).$$

In reinforcement learning, the action is chosen by a policy to maximize the expected total discounted reward. Most DP methods deal with problems where the state and action spaces are discrete, i.e. finite. However, when dealing with control of physical systems, the state and action spaces are continuous, i.e., infinite. To tackle the problem, DRL algorithms make use of function approximators, more specifically, DNNs. One major advantage of DRL algorithms is their direct applicability to a large family of complex systems, especially in the case of model-free approaches, e.g. [94, 129, 147]. In most cases, the environment to be controlled is considered a black box. In order to estimate the future performances of the policy without knowing the environment, many DRL algorithms exploit actor-critic architectures [117] (see Figure A.1). This family of methods exploits two or more DNNs (see e.g. [80]). The first is used as the policy,

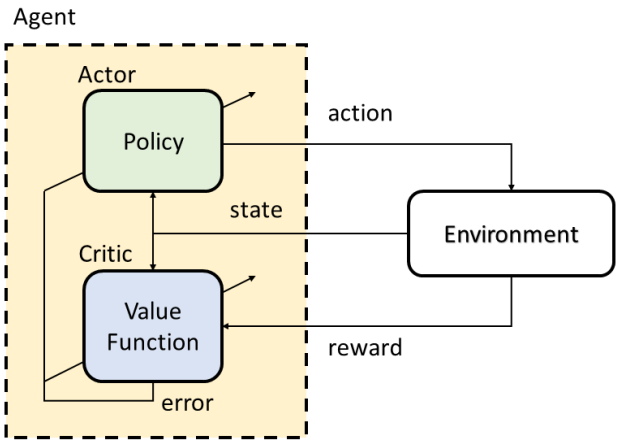


Figure A.1: Actor-critic structure

while the others typically predict its performance by estimating the action-value or state-value function. The *actor* is in charge of improving the policy DNN based on the value function that is estimated by the *critic*. Hence, the critic's role is to provide a DNN evaluating the current policy prescribed by the actor. While the policy network is updated based on the critic's estimation, the estimator itself is updated according to the observed reward.

More formally, in actor-critic algorithms, the reward signal is used to update the parameter vector $\phi \in \mathbb{R}^{n_\phi}$ of the action-value function estimator $\hat{\mathbf{Q}}^{\phi}_{\boldsymbol{\pi}}$, according to a criterion $J^\phi : \mathcal{X} \times \mathcal{U} \times \mathbb{R}^{n_\phi} \to \mathbb{R}$. The parameter vector $\theta \in \mathbb{R}^{n_\theta}$ of the actor policy $\boldsymbol{\pi}^\theta$ are subsequently updated according to a different cost function $J^\theta : \mathcal{X} \times \mathcal{U} \times \mathbb{R}^{n_\theta} \to \mathbb{R}$ depending on the new critic estimation. Then, for our purposes, we define the class of actor-critic algorithms $\mathbb{A}$ as the set of operators $\mathbb{A} := \{\mathbf{a} : \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_\phi} \to \mathbb{R}^{n_\theta} \times \mathbb{R}^{n_\phi}\}$ with input (θ_0, ϕ_0) and output $(\theta_{\bar{\kappa}}, \phi_{\bar{\kappa}})$ such that, for any $\kappa \in [0, \bar{\kappa}] \subseteq \mathbb{N}$, the parameters ϕ and θ are updated via some form of Stochastic Gradient Descent [210, Chapter 9.3]. The classic formulation imposes $\theta_{\kappa+1} = \theta_\kappa - \lambda^a_\kappa \nabla_\theta J^\theta$ and $\phi_{\kappa+1} = \phi_\kappa - \lambda^c_\kappa \nabla_\phi J^\phi$, where $\lambda^a_\kappa, \lambda^c_\kappa$ are the positive learning rates, subscript κ identifies a parameter vector at iteration κ, ∇_θ is the gradient with respect to the policy parameters, ∇_ϕ is the gradient with respect to the critic value function parameters and $\bar{\kappa} \in \mathbb{N}$ is the iteration index such that $\nabla_\theta J^\theta \leqslant \epsilon$ and $\nabla_\phi J^\phi \leqslant \epsilon$ for some small $\epsilon > 0$ and for all $\kappa \geqslant \bar{\kappa}$.

Training the critic. Inspired by the Bellman's equation, a common critic parameters objective is to minimize the one-step prediction error for all $(x, u) \in \mathcal{X} \times \mathcal{U}$, namely

$$J^\phi(x,u) = \mathbb{E}\left[\left(\mathbf{r}(x,u) + \max_{u^+ \sim \pi(x^+)} \hat{\mathbf{Q}}^{\phi}_{\boldsymbol{\pi}}(x^+,u^+) - \hat{\mathbf{Q}}^{\phi}_{\boldsymbol{\pi}}(x,u)\right)^2\right]. \tag{A.2}$$

where x^+, u^+ represent the state and action at the next step respectively. This objective comes from Q-learning theory and it can be seen it asks for a good estimation of the instantaneous reward over an infinite trajectory defined by the current control policy, assuming the estimation is correct from the successive step onward.

Training the policy. Many DRL update procedures exploit policy gradient theorems [196,211], which compute the gradient of the value function with respect to the policy parameters. Algo-

rithms learning a deterministic policy exploit deterministic policy gradient theorem [196]. Given a parametrized deterministic policy $\boldsymbol{\pi}^\theta$ and a set $\mathcal{X} \subset \mathbb{R}^{n_x}$, deterministic policy gradient updates take the form

$$\begin{aligned} \nabla_\theta \mathbf{J}^\pi(x) &= \int_{\mathcal{X}} \mathbf{d}^{\boldsymbol{\pi}}(x) \nabla_\theta \boldsymbol{\pi}^\theta(x) \nabla_u \hat{\mathbf{Q}}^\phi \boldsymbol{\pi}(x,u) \mid_{u=\boldsymbol{\pi}(x)} dx \\ &= \mathbb{E}_{x\sim\mathbf{d}^{\boldsymbol{\pi}}}[\nabla_\theta \boldsymbol{\pi}^\theta(x) \nabla_u \hat{\mathbf{Q}}^\phi \boldsymbol{\pi}(x,u) \mid_{u=\boldsymbol{\pi}(x)}], \end{aligned} \tag{A.3}$$

where ∇_u the gradient with respect to the actions $u = \boldsymbol{\pi}^\theta(x)$ and the distribution function $\mathbf{d}^{\boldsymbol{\pi}}(x) := \int_{\mathcal{X}} \sum_{\mathfrak{t}=1}^{\infty} \gamma^{\mathfrak{t}-1} \Pr(x_0) \Pr(x_0 \to x, \mathfrak{t}, \boldsymbol{\pi}^\theta) dx$ is the discounted state distribution, being $\Pr(x_0 \to x, \mathfrak{t}, \boldsymbol{\pi}^\theta)$ the probability of reaching the state $x \in \mathcal{X}$ after transitioning for $\mathfrak{t}$ time steps with initial condition x_0, following policy $\boldsymbol{\pi}^\theta$. For stochastic policies, updates are based on the stochastic policy gradient theorem [211, Theorem 2]

$$\begin{aligned} \nabla_\theta \mathbf{J}^\pi(x) &= \int_{\mathcal{X}} \mathbf{d}^{\boldsymbol{\pi}}(x) \int_{\mathcal{U}} \nabla_\theta \boldsymbol{\pi}^\theta(u|x) \hat{\mathbf{Q}}^\phi \boldsymbol{\pi}(x,u) du\, dx \\ &= \mathbb{E}_{x\sim\mathbf{d}^{\boldsymbol{\pi}}, u\sim\boldsymbol{\pi}^\theta}[\nabla_\theta \log \boldsymbol{\pi}^\theta(u|x) \hat{\mathbf{Q}}^\phi_{\boldsymbol{\pi}}(x,u)], \end{aligned} \tag{A.4}$$

It can be proven that the stochastic policy gradient is the generalization of its deterministic counterpart [195, Section 3.3].

A.2 Highlights on graph theory

In multi-agent control, a communication graph is typically described by a triplet $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{A}\}$ where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ is a set of $N \subset \mathbb{N}$ *vertices* (or nodes), $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of *edges* ϵ_{jh} modeling the interconnection between such nodes, and $\mathbf{A} \in \mathbb{R}^{N\times N}$ is the *adjacency matrix*, whose entries $a_{jh} \geqslant 0$ *weight* the flow of information from vertex j to vertex h. The *Laplacian matrix* $L \in \mathbb{R}^{N\times N}$ of the graph $\mathcal{G}$ is defined as

$$\ell_{jh} = -a_{jh} \quad \text{for } j \neq h, \qquad \ell_{jh} = \sum_{i=1}^{N} a_{ji} \quad \text{for } j = h,$$

where ℓ_{jh} is the (j,h)-th entry of L. We denote with $\mathcal{N}_i$ the set of in-neighbors of node i, i.e. the set $\mathcal{N}_i := \{j \in \{1,\ldots,N\} \,|\, \epsilon_{ij} \in \mathcal{E}\}$. A graph is said to be *undirected* if information exchange between agents can flow in any direction. If for some edges communication is not bi-directional, the graph is said to be *directed*. A graph is said to be *leader-connected* if there exists a leader node that can only send information, while bidirectional communication is allowed in the rest of the network. If some entries a_{ij} of the adjacency matrix are different from ± 1 or 0, the graph is said to be *weighted*. For general directed, weighted graphs, the Laplacian matrix is not diagonalizable and admits complex eigenvalues. If the graph is undirected, the Laplacian matrix is symmetric and, hence, has only real eigenvalues [89]. We define connectivity following [102, Definition 5.1].

Definition A.1 (Connected graph)**.** *A graph $\mathcal{G}$ is connected if there is a node v such that, for any other node $v_k \in \mathcal{V} \setminus \{v\}$, there is a path from v to v_k.*

Based on this definition, we recall the following property, see e.g., [89, 102, 161].

Lemma A.1. *A directed weighted graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ is connected if and only if its Laplacian matrix L has only one trivial eigenvalue $\lambda_1 = 0$ and all other eigenvalues $\lambda_2, \ldots, \lambda_N$ have positive real parts.*

Lemma A.1 implies that, for connected graphs, the vector $\mathbf{1}$ is the eigenvector of L associated to $\lambda_0 = 0$, namely $L\,\mathbf{1} = \mathbf{0}$. Note that, in general, the Laplacian of the network can be partitioned as

$$L = \begin{pmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{pmatrix} \tag{A.5}$$

where L_{11} is a scalar, L_{12} is a $N-1$ row vector, L_{21} is a $N-1$ column vector and L_{22} is a $(N-1)\times(N-1)$ matrix. Given the above partition, for general directed, weighted graphs, the following holds.

Lemma A.2. *Let the weighted graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ be directed and connected, with Laplacian L and L_{11}, L_{12} defined as in* (A.5)*. Then, there exist $M \in \mathbb{S}^{N-1}_{\succ 0}$ and constants $\underline{m}, \overline{m}, \underline{\mu}, \bar{\mu} > 0$, $\rho_M \in (0,1]$ such that*

$$\underline{m}\,\mathrm{I}_{N-1} \preceq M \preceq \overline{m}\,\mathrm{I}_{N-1}, \quad \rho_M \leqslant \frac{\underline{m}}{\overline{m}} \tag{A.6a}$$

$$\mathrm{He}\,\{M(L_{22} - \mathbf{1}\,L_{12})\} \succeq 2\underline{\mu} M \tag{A.6b}$$

$$(L_{22} - \mathbf{1}\,L_{12})^\top M (L_{22} - \mathbf{1}\,L_{12}) \preceq \bar{\mu}^2 M. \tag{A.6c}$$

Proof. Since the graph is connected, Lemma A.1 ensures that the Laplacian L, as in (A.5), has one zero eigenvalue and $N-1$ eigenvalues with positive real part. Consider the transformation

$$T = T^{-1} := \begin{pmatrix} 1 & 0 \\ \mathbf{1} & -\mathrm{I}_{N-1} \end{pmatrix}$$

and consider the change of coordinates on the Laplacian defined by

$$\mathcal{L} := TLT^{-1} = T \begin{pmatrix} 0 & -L_{12} \\ \mathbf{0} & -L_{22} \end{pmatrix} = \begin{pmatrix} 0 & -L_{12} \\ \mathbf{0} & L_{22} - \mathbf{1}\,L_{12} \end{pmatrix},$$

where we exploited $L\,\mathbf{1} = \mathbf{0}$. Since T is full rank, by similarity transformation $\operatorname{spec}\mathcal{L} = \operatorname{spec} L$, namely, it has one zero eigenvalue and $N-1$ eigenvalues with positive real part. Then, due to the block-triangular structure of $\mathcal{L}$, all the eigenvalues of $L_{22} - \mathbf{1}\,L_{12}$ have positive real part. Define $\tilde{L} := \mathbf{1}\,L_{12} - L_{22}$. Since all eigenvalues of $\tilde{L}$ have negative real part, by the Lyapunov equation there exists $M \in \mathbb{S}^{N-1}_{\succ 0}$ satisfying

$$M\tilde{L} + \tilde{L}^\top M = -\,\mathrm{I}_{N-1} \prec 0.$$

In turn, this implies $M(-\tilde{L}) + (-\tilde{L})^\top M \succ 0$. As a consequence, there exists a sufficiently small scalar $\underline{\mu} > 0$ such that (A.6b) holds. We now move to the other inequalities in Lemma A.2. Since $M \in \mathbb{S}^{N-1}_{\succ 0}$, (A.6a) and (A.6c) trivially hold with of $\underline{m}$ and $\overline{m}$ being the smallest and largest eigenvalues of M respectively, $\rho_M = \underline{m}\overline{m}^{-1}$ and a sufficiently large $\bar{\mu}$. □

As a consequence, we recover the following property of leader-connected graphs, e.g. [89].

Lemma A.3. *Suppose the graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ is undirected and leader-connected . Then the Laplacian L can be partitioned as*

$$L = \begin{pmatrix} 0 & 0 \\ L_{21} & L_{22} \end{pmatrix}.$$

Moreover, there exists a strictly positive real numbers $\underline{\mu}, \bar{\mu} > 0$ such that $\underline{\mu}\,\mathrm{I}_{N-1} \preceq L_{22} \preceq \bar{\mu}\,\mathrm{I}_{N-1}$.

B Tokamak model and simulation algorithm

B.1 Safety factor and thermal energy control model

Consider the reaction-diffusion equation describing the magnetic flux dynamics

$$\frac{\partial \psi}{\partial t}(s,t) = \frac{D(s,t)}{a_\rho^2}\frac{\partial^2 \psi}{\partial s^2}(s,t) + \frac{G(s,t)}{a_\rho}\frac{\partial \psi}{\partial s}(s,t) + S(s,t). \tag{B.1}$$

The coefficients $D(s,t)$, $G(s,t)$, and $S(s,t)$ can be computed by following [36, eq. III-34] as

$$D(s,t) = \frac{\eta_\parallel C_2}{\mu_0 C_3} \quad G(s,t) = \frac{\eta_\parallel F a_\rho}{\mu_0 C_3}\frac{\partial}{\partial \rho}\left(\frac{C_2}{F}\right) \quad S(s,t) = L(\rho,t) j_{ni} \quad L(\rho,t) = \frac{\eta_\parallel V' B_{\phi 0}}{F C_3} \tag{B.2}$$

where $\eta_\parallel(\rho,t)$ is the resistivity, μ_0 is the permeability of the free space, F is the diamagnetic function, $V(\rho,t)$ is the plasma volume, $V' = \frac{\partial V}{\partial \rho}$ is the volume spatial derivative while C_2 and C_3 are space varying parameters depending on the considered plasma geometry configuration. $j_{ni}(s,t)$ is the non-inductive current source and includes the bootstrap current j_{bs} as well as the ECCD density currents j_{eccd}

$$j_{ni} = j_{bs} + j_{eccd}.$$

In this work, the bootstrap currents are computed according to [70]

$$j_{bs} = -\frac{k_{bs}}{\partial\psi/\partial\rho}\left(\mathcal{L}_{31}\frac{\partial \ln(n_e)}{\partial \rho} + R_{pe}(\mathcal{L}_{31}+\mathcal{L}_{32})\frac{\partial \ln(T_e)}{\partial \rho} + (1-R_{pe})(\mathcal{L}_{31}+\alpha\mathcal{L}_{34})\frac{\partial \ln(T_i)}{\partial \rho}\right) \tag{B.3}$$

where T_e is the electronic temperature, $T_i(s,t) \approx \alpha_{Ti}(t) T_e(s,t)$ is the ions temperature, n_e is the electron density, α is a constant parameter while k_{bs}, $\mathcal{L}_{31}$, $\mathcal{L}_{32}$, $\mathcal{L}_{34}$, R_{pe} are space varying parameters depending on the electronic and ion temperatures and on the plasma geometric configuration. The ion-to-electron temperature ratio can be fixed to $\alpha_{Ti} = 0.7$. The electron density can be approximated by

$$n_e(s,t) \approx \frac{\gamma_n + 1}{\gamma_n}(1 - s^{\gamma_n})\bar{n}_e$$

where $\bar{n}_e$ is the electron line average density, that in our case has been considered to be constant $\bar{n}_e = 1 \times 10^{-19}$.

Symbol	Description	Unit
a_ρ	Radius of the LCFS	m
B	Magnetic field	T
B_ϕ	Toroidal magnetic field	T
$\eta_\parallel$	Resistivity	$\Omega \times$ m
e	Electron charge 1.6022×10^{-19}	C
F	Diamagnetic function	T $\times$ m
ι	Inverse of the safety factor	
I_p	Total plasma current	A
j_{ni}	Non-inductive current	J/m^2
j_{bs}	Bootstrap current	J/m^2
j_{eccd}	Electron Ciclotron Current Drive density current	J/m^2
j_{tor}	Toroidal density current	J/m^2
μ_0	Permeability of the free space $4\pi \times 10^{-7}$	H/m
n_e	Electron density profile	m^{-3}
n_i	Ion density profile	m^{-3}
ϕ	Magnetic flux of the toroidal field	T/m^2
P_{eccd}	ECCD power	W
P_{OH}	Total ohmic power	W
ψ	Magnetic flux of the poloidal field	T/m^2
q	Safety factor	
R	Major plasma radius	m
R_0	Magnetic center location	m
ρ	Spatial index	
τ_{th}	Thermal energy confinement time	s
T_e	Electronic temperature	eV
T_i	Ion temperature	eV
U_{pl}	Toroidal loop voltage	
V	Plasma volume	m^3
W_{th}	Plasma thermal energy	J
s	Normalized spatial index	
χ_e	Electron thermal diffusivity	m^2/s
Z_{eff}	Effective Plasma charge	C

Table B.1: Table of symbols and corresponding units.

An appropriate and effective choice used in control-oriented plasma-dynamics simulators is to approximate the current density by a weighted Gaussian [233]. According to [70], the ECCD efficiency can be modelled heuristically as

$$j_{eccd,i}(\rho,t) = c_{cd,i} e^{\rho^2/0.5^2} \frac{T_e}{n_e} e^{-4(\rho-\rho_{dep,i})^2/w_{cd,i}^2} P_{eccd,i}(t) \tag{B.4}$$

where w_{dep} is the deposition width and ρ_{dep} is the location of the peak of the deposition, while $P_{eccd,i}$ is the power associated with the i-th antenna. The parameter c_{cd} is a machine-dependent parameter that can be chosen to scale the expression to the experimentally obtained current drive values. The total ECCD current is obtained as the sum of the different antennas, that in this work are considered to be two

$$j_{eccd}(\rho,t) = j_{eccd,1}(\rho,t) + j_{eccd,2}(\rho,t).$$

According to [182], the conductivity can be computed as

$$\eta_{\|} = \frac{1}{\sigma_{\|}} = \frac{1}{\sigma_{sptz} c_{neo}}. \tag{B.5}$$

The Spitzer conductivity σ_{sptz} depends on the electron temperature and on the effective value of the plasma charge Z_{eff}. This last parameter may in general vary spatially, but it is chosen here to be a fixed quantity for the whole plasma $Z_{eff} = 3.5$. The neoclassical correction c_{neo} depends on the electron and ion collisionality parameters as well as on Z_{eff}. Both σ_{sptz} and c_{neo} are space and time-varying.
Specific boundary conditions have to be considered both at the center and on the LCFS of the plasma. At the plasma center, the spatial variation of the flux is zero

$$\frac{\partial \psi}{\partial s}(0,t) = 0,$$

while at the LCFS, we consider a Neumann boundary condition

$$\frac{\partial \psi}{\partial s}(1,t) = -\frac{R_0 \mu_0 I_p(t)}{2\pi}. \tag{B.6}$$

where I_p is the total plasma current. The toroidal flux $\phi(s,t)$ is defined as the magnetic flux passing through a poloidal surface centered at R_0 and with radius ρ. Assuming that the toroidal magnetic field remains constant, it is possible to obtain an explicit formula for the toroidal flux [233]

$$\phi(s,t) = \frac{1}{2\pi}\int_{S_{pol}} B(R,Z) dS_{pol} = -\frac{1}{2\pi}\int_{S_{pol}} B_\phi dS_{pol} \approx -\frac{B_{\phi 0} a_\rho^2 s^2}{2}. \tag{B.7}$$

An instrumental quantity for the temperature dynamics is the ohmic power

$$P_{OH} = \int_0^1 \frac{1}{2\pi R_0} U_{pl} j_{tor} ds \tag{B.8}$$

In the previous equation, U_{pl} identifies the toroidal loop voltage while j_{tor} corresponds to the toroidal density and they can be computed as

$$U_{pl} = \frac{\partial \psi}{\partial t} \quad j_{tor} = \frac{1}{\eta_{\|}} \left(\frac{D(s,t)}{a_\rho^2} \frac{\partial^2 \psi}{\partial s^2} + \frac{G(s,t)}{a_\rho} \frac{\partial \psi}{\partial s} \right). \tag{B.9}$$

We consider thermal dynamics in the model, to account for the delay introduced by temperature diffusion. Rather than using a pure delay term, we chose to represent this delay with a dynamic equation for more accurate modeling. However, we do not to incorporate the full distributed thermal diffusion model in order to minimize simulation time and achieve a sufficiently short training time for the RL algorithm. The temperature diffusion equation writes

$$\frac{3}{2}\frac{\partial n_e T_e}{\partial t} = \frac{1}{\rho}\frac{\partial}{\partial \rho}\left(\rho n_e \chi_e(\rho,t)\frac{\partial T_e}{\partial \rho}\right) - \frac{3 n_e T_e}{2\tau_d} + S_T(\rho,t)$$

where $\chi_e(\rho,t)$ is the electron thermal diffusivity, τ_d is the time-varying damping modeling the losses and $S_T(\rho,t)$ is the source term. In our specific application, where we consider two ECCD inputs, we have

$$S_T(\rho,t) = S_{T,eccd,1}(\rho,t) + S_{T,eccd,2}(\rho,t).$$

It is worth remarking that for $i \in \{1,2\}$ the source term has an amplitude such that

$$\int_0^1 S_{T,eccd,i}(s,t)ds = P_{eccd,i}.$$

Because of the high uncertainty of the proposed temperature model and in order to effectively diminish the simulation time, we choose to use an empirical reduced-order model that approximates the actual temperature dynamics, similar to the one proposed in [232]. This model is composed of an ordinary differential equation representing the evolution of the thermal energy W_{th} and a DNN that takes as input the total power and the thermal energy and returns the distributed temperature profile. The plasma thermal energy is defined as

$$W_{th} = W_e(t) + W_i(t) = \frac{3e}{2}\int_V (n_e T_e + n_i T_i)dV = \frac{3e}{2}\int_V (1+\alpha_{Ti}\alpha_{ni}) n_e T_e dV$$

where $n_i \approx \alpha_{ni} n_e(s,t)$ is the ions density, e is the electron charge and W_e, W_i are the electrons and ions energy, respectively. The density ratio can be approximately computed as $\alpha_{ni} \approx (7 - Z_{eff})/6$.

The exponents in the τ_{th} expression in (2.58) have been obtained by applying linear regression on data obtained from RAPTOR simulations. In particular, the collected data together with the applied open-loop input, are organised in the vectors X and Y as following

$$X = \begin{pmatrix} 1 \\ \log(P_{tot}) \\ \log(1+P_{eccd,1}) \\ \log(1+P_{eccd,2}) \end{pmatrix} \quad Y = \tau_{th}.$$

Linear regression is then applied to the couple (X,Y) to obtain the power constant values k_0, k_1, k_2, k_3 such that

$$\tau_{th} = e^{k_0} P_{tot}^{k_1} (1+P_{eccd,1})^{k_2} (1+P_{eccd,2})^{k_3}.$$

The temperature profile is obtained as the output of a DNN

$$T_e(x,t) = f_{NN}(P_{tot}, W_{th}). \tag{B.10}$$

The neural network has been trained using a set of temperature profiles associated with the total power and the thermal energy obtained by some RAPTOR simulations. For both the τ_{th} linear regression and the DNN training, the RAPTOR simulations have been obtained by applying different constant open-loop inputs to the system and extracting the total power, the thermal energy, τ_{th} and the temperature profiles.

B.2 Simulation Algorithm

Employing a combination of implicit-explicit time discretization and fixed-step spatial discretization, as outlined in [233, Appendix A], system (B.1) can be approximated by the difference equation

$$\psi^+ = B_{\mathfrak{t}}^{-1} A_{\mathfrak{t}} \psi + B_{\mathfrak{t}}^{-1} S_{\mathfrak{t}} \tag{B.11}$$

where $\psi, S_{\mathfrak{t}} \in \mathbb{R}^{n_\psi}$ are n_ψ-dimensional vectors of the magnetic flux and the source term at n_ψ different point of the spatial domain at the $\mathfrak{t}$ time step. The matrices $A_{\mathfrak{t}} \in \mathbb{R}^{n_\psi \times n_\psi}$ and $B_{\mathfrak{t}} \in \mathbb{R}^{n_\psi \times n_\psi}$ depend on the plasma physical parameters and change at every iteration $\mathfrak{t}$. The time discretization step is fixed at $\delta t = 0.01$, with an implicit-explicit ratio of $h = 0.45$, and the total simulation time is referred to as T_{sim}. The space domain is divided into $N = 21$ discretization elements, with a fixed spatial discretization step of $\delta x_i = 0.05$. Similarly, the thermal energy dynamics can be approximated by the difference equation

$$W_{th}^+ = d_{\mathfrak{t}} W_{th} + s_{\mathfrak{t}} P_{tot,\mathfrak{t}}$$

where $W_{th}, P_{tot,\mathfrak{t}} \in \mathbb{R}$ are the thermal energy and the total power at discrete time $\mathfrak{t}$, $d_{\mathfrak{t}}$ and $s_{\mathfrak{t}}$ are coefficients depending on $\tau_{th,\mathfrak{t}}$. It is worth remarking that our study case is similar to the one considered in [145], where the two available inputs act on the same spatial point: the first antenna $P_{eccd,1}$ acts positively on z while the second $P_{eccd,2}$ acts negatively. The two input powers have limited maximum power $P_{eccd,i} \in [\underline{P}_{eccd,i}, \bar{P}_{eccd,i}]$. To control the magnetic flux gradient, the control action corresponds to the difference between the two antennas' power. Inversely, the control action for the temperature profile corresponds to the sum of the two antennas' power. Since in this work we are interested in magnetic control, in the following we define a function mapping from the desired power difference to the value of each antenna power. Firstly, we define the control input $u \in [0, 1]$ that is mapped to the desired difference $\alpha \in [\underline{\alpha}, \bar{\alpha}]$ between the two ECCD powers applied at the $\mathfrak{t}$ time iteration

$$\alpha = \underline{\alpha} + u(\bar{\alpha} - \underline{\alpha}) = P_{eccd,1} - P_{eccd,2}, \tag{B.12}$$

where $\underline{\alpha} = \underline{P}_{eccd,i} - \bar{P}_{eccd,2}$ and $\bar{\alpha} = \bar{P}_{eccd,1} - \underline{P}_{eccd,2}$. Given a desired power difference α, the control input powers are mapped to minimize their sum $P_{eccd,1} + P_{eccd,2}$. The mapping can be expressed by

$$\begin{cases} P_{eccd,1} = \underline{P}_{eccd,i} \\ P_{eccd,2} = \underline{P}_{eccd,i} - \alpha & \text{if} \quad \alpha < \underline{P}_{eccd,i} - \underline{P}_{eccd,2} \\ \\ P_{eccd,1} = \alpha + \underline{P}_{eccd,2} \\ P_{eccd,2} = \underline{P}_{eccd,2} & \text{if} \quad \alpha \geqslant \underline{P}_{eccd,i} - \underline{P}_{eccd,2}. \end{cases} \tag{B.13}$$

After the spatial discretization of the magnetic flux dynamics and the temporal discretization of both the magnetic flux and thermal energy dynamics, we obtain the difference equation

$$\xi^+ = \begin{pmatrix} B_{\mathfrak{t}}^{-1} A_{\mathfrak{t}} & 0 \\ 0 & d_{\mathfrak{t}} \end{pmatrix} \xi + \begin{pmatrix} B_{\mathfrak{t}}^{-1} S_{\mathfrak{t}} \\ s_{\mathfrak{t}} P_{tot,\mathfrak{t}} \end{pmatrix} = f(\xi, u). \tag{B.14}$$

in the state variable

$$\xi = \begin{pmatrix} \psi \\ W_{th} \end{pmatrix} \in \mathbb{R}^{n_\psi + 1}.$$

The steps for the plasma magnetic flux and temperature simulation are listed in Algorithm 7.

Algorithm 7 Tokamak environment simulation algorithm

Input data:
- Initialization of ψ_0, $W_{th,0}$ and $P_{OH,0}$ from RAPTOR simulation after the ramp-up phase
- Initialization of the constant physical parameters
- Initialization of the constant simulation parameters
- Initialization of the open-loop input U
- Simulation initialization $\mathfrak{t} = 0$

while $\mathfrak{t} < T_{sim}/\delta t$ **do**
  Input extraction : $u = U[\mathfrak{t}]$
  Temperature: $T_{e,\mathfrak{t}}, T_{i,\mathfrak{t}} \leftarrow f_{temp}(u, P_{OH}, W_{th})$ with (2.59), (B.10) and (B.12)-(B.13)
  Resistivity: $\eta_{\parallel,\mathfrak{t}} \leftarrow f_{resistivity}(T_{e,\mathfrak{t}}, \psi)$ with (B.5)
  Bootstrap current: $j_{bs,\mathfrak{t}} \leftarrow f_{boostrap}(T_{e,\mathfrak{t}}, T_{i,\mathfrak{t}}, \psi)$ with (B.3)
  ECCD deposit: $(j_{eccd,1})_{\mathfrak{t}}, (j_{eccd,2})_{\mathfrak{t}} \leftarrow f_{eccd}(u, T_{e,\mathfrak{t}})$ with (B.4)
  Non-inductive currents: $j_{ni,\mathfrak{t}} \leftarrow j_{bs,\mathfrak{t}} + (j_{eccd,1})_{\mathfrak{t}} + (j_{eccd,2})_{\mathfrak{t}}$
  Diffusion coefficients: $D_{i,\mathfrak{t}}, G_{i,\mathfrak{t}}, L_{i,\mathfrak{t}} \leftarrow f_{coeff}(\eta_{\parallel,\mathfrak{t}})$ with (B.2)
  Magnetic flux: $\psi^+ \leftarrow f_{\psi}(\psi, D_{i,\mathfrak{t}}, G_{i,\mathfrak{t}}, L_{i,\mathfrak{t}}, j_{ni,\mathfrak{t}})$ with (B.14)
  Thermal energy: $W_{th}^+ \leftarrow f_{thermal}(u, P_{OH}, W_{th})$ with (B.14)
  Ohmic Power: $P_{OH}^+ = f_{ohmic}(\eta_{\parallel,\mathfrak{t}}, T_{e,\mathfrak{t}}, T_{i,\mathfrak{t}}, u, \psi^+)$ with (B.8)-(B.9)
  $\mathfrak{t} \leftarrow \mathfrak{t} + 1$
end while

The constant parameters are fixed as follows

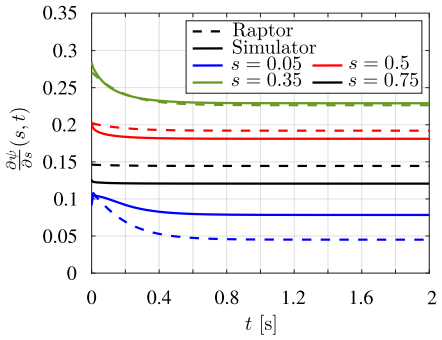
$$\begin{array}{cccccc} B_{\phi 0} = 1.44, & R_0 = 0.88, & a_\rho = 0.25, & Z_{eff} = 3.5, & \delta_0 = 0.3, \\ \bar{n}_e = 1 \times 10^{-19}, & \alpha_{Ti} = 0.7, & \mu_0 = 4\pi \times 10^{-7}, & \gamma_n = 2. & \end{array} \tag{B.15}$$

In the current experiment, we assume that the two ECCD actuators, described by the injected current density in (B.4), have the following parameters

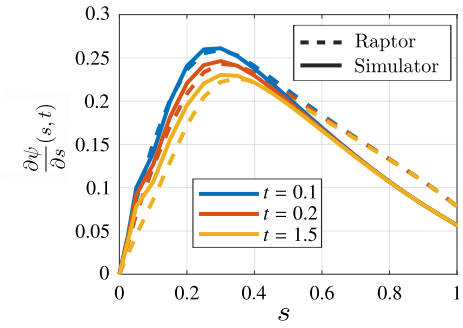
$$\begin{array}{cccccc} c_{cd,1} = 1, & \rho_{dep,1} = 0, & w_{cd,1} = 0.35, & \bar{P}_{eccd,1} = 900(MW), & \underline{P}_{eccd,i} = 360(MW), \\ c_{cd,2} = -1, & \rho_{dep,2} = 0, & w_{cd,2} = 0.35, & \bar{P}_{eccd,2} = 750(MW), & \underline{P}_{eccd,2} = 100(MW). \end{array} \tag{B.16}$$

It is worth noticing that the Ohmic power at time instant $\mathfrak{t}+1$ is computed with the variables $\eta_{\parallel,\mathfrak{t}}$, $T_{e,\mathfrak{t}}$, $T_{i,\mathfrak{t}}$, u, belonging to the time instant $\mathfrak{t}$, as well as ψ^+, belonging to the time instant $\mathfrak{t}+1$. With this simulation procedure, it is not possible to only use variables belonging to the time instant $\mathfrak{t}+1$ for the P_{OH}^+ calculation because P_{OH}^+ itself is needed to compute $T_{e,\mathfrak{t}+1}$ and $T_{i,\mathfrak{t}+1}$. This is an intrinsic property of this simulation procedure, introduced in [233], that avoids the implementation of a fixed point iteration research to find all the states at the time step $\mathfrak{t}+1$. The nonlinear components of the model are delayed by one sample while an implicit-explicit scheme is used for the linear components, thus avoiding the fixed-point iteration algorithm to obtain a faster simulation.

To test the simulator's precision with respect to a certain tokamak configuration, we compare the trajectories obtained with the application of some constant open-loop controls with the ones obtained through the application of the same controls with the RAPTOR simulator. The RAPTOR simulator is a real-time predictor of the Ψ and T_e profiles used as an observer in the TCV control environment [70]. The kinetic and magnetic profiles are obtained by simulating two coupled nonlinear reaction-diffusion PDEs. Therefore, the RAPTOR simulator provides fairly precise simulation results that can be taken as a reference for our simulator. Figure B.1 shows the $\frac{\partial \psi}{\partial \rho}(\rho, t)$ trajectories with the application of the open loop input $a = 0.15$. The initial condition for both



(a) $\partial\psi/\partial s$ open-loop trajectories.



(b) $\partial\psi/\partial s$ open-loop profiles at $t = 0.1, 0.2, 1.5$.

Figure B.1: Comparison between RAPTOR and training model open-loop simulations.

the RAPTOR and the training simulator corresponds to the steady state with the constant open loop input $a = 0.7$. In Figure B.1a are shown the trajectories of four points of the spatial domain $x = 0.05,\ 0.35,\ 0.5,\ 0.75$ of both the RAPTOR and training simulator. While in Figure B.1b are shown the $\frac{\partial \psi}{\partial \rho}(\rho, t)$ profiles at time instants $t = 0.1,\ 0.2,\ 1.5$. We remark that there exists a visible difference between the profiles obtained with the proposed simulation algorithm and the RAPTOR simulator. Nevertheless, we can observe similar trends:

- Small values of the input result to small values of the magnetic gradient peak,
- Small values of the input result to a right-shift of the magnetic gradient peak.

Bibliography

[1] M. Abdelrahem, C. M. Hackl, R. Kennel, and J. Rodriguez. Efficient direct-model predictive control with discrete-time integral action for pmsgs. *IEEE Transactions on Energy Conversion*, 34(2):1063–1072, 2019.

[2] R. P. Agarwal. *Difference equations and inequalities: theory, methods, and applications*. CRC Press, 2000.

[3] Z. Aminzare and E. D. Sontag. Contraction methods for nonlinear systems: A brief introduction and some open problems. In *53rd IEEE Conference on Decision and Control*, pages 3835–3847, 2014.

[4] Z. Aminzare and E. D. Sontag. Synchronization of diffusively-connected nonlinear systems: Results based on contractions with respect to general norms. *IEEE Transactions on Network Science and Engineering*, 1(2):91–106, 2014.

[5] V. Andrieu, B. Jayawardhana, and L. Praly. Transverse exponential stability and applications. *IEEE Transactions on Automatic Control*, 61(11):3396–3411, 2016.

[6] V. Andrieu, B. Jayawardhana, and L. Praly. Characterizations of global transversal exponential stability. *IEEE Transactions on Automatic Control*, 66(8):3682–3694, 2020.

[7] V. Andrieu, B. Jayawardhana, and S. Tarbouriech. Some results on exponential synchronization of nonlinear systems. *IEEE Transactions on Automatic Control*, 63(4):1213–1219, 2018.

[8] V. Andrieu and S. Tarbouriech. LMI conditions for contraction and synchronization. *IFAC-PapersOnLine*, 52(16):616–621, 2019.

[9] D. Angeli. A Lyapunov approach to incremental stability properties. *IEEE Transactions on Automatic Control*, 47(3):410–421, 2002.

[10] D. Angeli. Further results on incremental input-to-state stability. *IEEE Transactions on Automatic Control*, 54(6):1386–1391, 2009.

[11] D. Angeli, M. A. Al-Radhawi, and E. D. Sontag. A robust Lyapunov criterion for nonoscillatory behaviors in biological interaction networks. *IEEE Transactions on Automatic Control*, 67(7):3305–3320, 2022.

[12] D. Angeli and E. Sontag. Monotone control systems. *IEEE Transactions on Automatic Control*, 48(10):1684–1698, 2003.

[13] M. Arcak. Passivity as a design tool for group coordination. *IEEE Transactions on Automatic Control*, 52(8):1380–1390, 2007.

[14] M. Arcak. Certifying spatially uniform behavior in reaction–diffusion pde and compartmental ode systems. *Automatica*, 47(6):1219–1229, 2011.

[15] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017.

[16] D. Astolfi. *Observers and robust output regulation for nonlinear systems*. PhD thesis, Universitá di Bologna, 2016.

[17] D. Astolfi and L. Praly. Integral action in output feedback for multi-input multi-output nonlinear systems. *IEEE Transactions on Automatic Control*, 62(4):1559–1574, 2017.

[18] D. Astolfi, L. Praly, and L. Marconi. Harmonic internal models for structurally robust periodic output regulation. *Systems & Control Letters*, 161:105154, 2022.

[19] S. Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fundamenta mathematicae*, 3(1):133–181, 1922.

[20] J. E. Barton, E. Schuster, F. Felici, and O. Sauter. Closed-loop control of the safety factor profile in the tcv tokamak. In *53rd IEEE Conference on Decision and Control*, pages 5660–5665. IEEE, 2014.

[21] J. E. Barton, W. P. Wehner, E. Schuster, F. Felici, and O. Sauter. Simultaneous closed-loop control of the current profile and the electron temperature profile in the tcv tokamak. In *2015 American Control Conference (ACC)*, pages 3316–3321. IEEE, 2015.

[22] F. Bayer, M. Bürger, and F. Allgöwer. Discrete-time incremental ISS: A framework for robust NMPC. In *European Control Conference*, pages 2068–2073, 2013.

[23] R. Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

[24] M. S. Benachour, V. Andrieu, L. Praly, and H. Hammouri. Forwarding design with prescribed local behavior. *IEEE Transactions on Automatic Control*, 58(12):3011–3023, 2013.

[25] F. Berkenkamp, R. Moriconi, A. P. Schoellig, and A. Krause. Safe learning of regions of attraction for uncertain, nonlinear systems with gaussian processes. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 4661–4666. IEEE, 2016.

[26] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause. Safe model-based reinforcement learning with stability guarantees. *Advances in neural information processing systems*, 30, 2017.

[27] D. P. Bertsekas. Dynamic programming and optimal control 4th edition, volume ii chapter 4 noncontractive total cost problems. *Massachusetts Institute of Technology*, 2018.

[28] D. P. Bertsekas. *Reinforcement Learning and Optimal Control*. Athena scientific, 2nd printing edition, 2018.

[29] J. Blum. *Numerical simulation and optimal control in plasma physics*. New York, NY; John Wiley and Sons Inc., 1989.

[30] N. Bof, R. Carli, and L. Schenato. Lyapunov theory for discrete time systems. *arXiv preprint arXiv:1809.05289*, 2018.

[31] F. Bonassi, M. Farina, and R. Scattolini. Stability of discrete-time feed-forward neural networks in NARX configuration. *IFAC-PapersOnLine*, 54(7):547–552, 2021.

[32] M. D. Boyer, J. Barton, E. Schuster, M. L. Walker, T. C. Luce, J. R. Ferron, B. G. Penaflor, R. D. Johnson, and D. A. Humphreys. Backstepping control of the toroidal plasma current

profile in the DIII-D tokamak. *IEEE Transactions on Control Systems Technology*, 22(5):1725–1739, 2014.

[33] S. J. Bradtke. Reinforcement learning applied to linear quadratic regulation. In *Advances in neural information processing systems*, pages 295–302, 1993.

[34] F. Bribiesca Argomedo, E. Witrant, C. Prieur, S. Brémond, R. Nouailletas, and J. Artaud. Lyapunov-based distributed control of the safety-factor profile in a tokamak plasma. *Nuclear Fusion*, 53(3):033005, feb 2013.

[35] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym, 2016.

[36] R. Brégeon. *Évolution résistive du profil de courant dans les tokamaks, application à l'optimisation des echarges de Tore Supra*. PhD thesis, Université de Provence (Aix-Marseille I), CEA - Cadarache, 1998.

[37] F. Bullo. *Contraction Theory for Dynamical Systems*, volume 1. Kindle Direct Publishing, 2022.

[38] C. I. Byrnes and A. Isidori. Limit sets, zero dynamics, and internal models in the problem of nonlinear output regulation. *IEEE Transactions on Automatic Control*, 2003.

[39] G. Casadei, D. Astolfi, A. Alessandri, and L. Zaccarian. Synchronization in networks of identical nonlinear systems via dynamic dead zones. *IEEE Control Systems Letters*, 3(3):667–672, 2019.

[40] G. Casadei, A. Isidori, and L. Marconi. About disconnected topologies and synchronization of homogeneous nonlinear agents over switching networks. *International Journal of Robust and Nonlinear Control*, 28(3):901–917, 2018.

[41] V. Chellaboina and W. M. Haddad. Stability margins of discrete-time nonlinear nonquadratic optimal regulators. In *Proceedings of the 37th IEEE Conference on Decision and Control*, volume 2, pages 1786–1791 vol.2, 1998.

[42] C. Chen, L. Xie, K. Xie, F. L. Lewis, and S. Xie. Adaptive optimal output tracking of continuous-time systems via output-feedback-based reinforcement learning. *Automatica*, 2022.

[43] M. Z. Q. Chen, L. Zhang, H. Su, and G. Chen. Stabilizing solution and parameter dependence of modified algebraic Riccati equation with application to discrete-time network synchronization. *IEEE Transactions on Automatic Control*, 61(1):228–233, 2016.

[44] G. Chesi. *Domain of attraction: analysis and control via SOS programming*, volume 20. Springer, 2011.

[45] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

[46] J. Choi, F. Castaneda, C. J. Tomlin, and K. Sreenath. Reinforcement learning for safety-critical control under model uncertainty, using control Lyapunov functions and control barrier functions. *Robotics: Science and Systems (RSS)*, 2020.

[47] N. Chopra and M. Spong. Output synchronization of nonlinear systems with relative degree one. In *Recent advances in learning and control*, pages 51–64. Springer, 2008.

[48] Y. Chow, O. Nachum, E. Duenez-Guzman, and M. Ghavamzadeh. A Lyapunov-based approach to safe reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[49] Y. Chow, O. Nachum, A. Faust, E. Duenez-Guzman, and M. Ghavamzadeh. Lyapunov-based safe policy optimization for continuous control. *arXiv preprint arXiv:1901.10031, ICML Workshop RL4RealLife*, 2019.

[50] F. H. Clarke. *Optimization and nonsmooth analysis*. SIAM, 1990.

[51] M. Coraggio, P. DeLellis, S. J. Hogan, and M. Di Bernardo. Synchronization of networks of piecewise-smooth systems. *IEEE Control Systems Letters*, 2(4):653–658, 2018.

[52] E. Coumans and Y. Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. `http://pybullet.org`, 2016–2021.

[53] A. Cristofaro and M. Mattioni. Hybrid consensus for multi-agent systems with time-driven jumps. *Nonlinear Analysis: Hybrid Systems*, 43:101113, 2021.

[54] O. Dalin, R. Ofir, E. B. Shalom, A. Ovseevich, F. Bullo, and M. Margaliot. Verifying k-contraction without computing k-compounds. *arXiv preprint:2209.01046*, 2022.

[55] W. D'Amico, A. La Bella, and M. Farina. An incremental input-to-state stability condition for a generic class of recurrent neural networks. *arXiv preprint arXiv:2210.09721*, 2022.

[56] A. Davydov, V. Centorrino, A. Gokhale, G. Russo, and F. Bullo. Contracting dynamics for time-varying convex optimization. *arXiv preprint arXiv:2305.15595*, 2023.

[57] A. Davydov, S. Jafarpour, and F. Bullo. Non-euclidean contraction theory for robust nonlinear stability. *IEEE Transactions on Automatic Control*, 67(12):6667–6681, 2022.

[58] C. Dawson, S. Gao, and C. Fan. Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods for robotics and control. *IEEE Transactions on Robotics*, 2023.

[59] M. C. De Oliveira, J. Bernussou, and J. C. Geromel. A new discrete-time robust stability condition. *Systems & Control Letters*, 37(4):261–265, 1999.

[60] C. De Persis, E. R. A. Weitenberg, and F. Dörfler. A power consensus algorithm for dc microgrids. *Automatica*, 89:364–375, 2018.

[61] J. Degrave, F. Felici, J. Buchli, M. Neunert, B. Tracey, F. Carpanese, T. Ewalds, R. Hafner, A. Abdolmaleki, D. de Las Casas, C. Donner, L. Fritz, C. Galperti, A. Huber, J. Keeling, M. Tsimpoukelli, J. Kay, A. Merle, J. M. Moret, S. Noury, D. Pesamosca, F.and Pfau, O. Sauter, C. Sommariva, S. Coda, B. Duval, A. Fasoli, P. Kohli, K. Kavukcuoglu, D. Hassabis, and M. Riedmiller. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, 2022.

[62] S. Devasia, D. Chen, and B. Paden. Nonlinear inversion-based output tracking. *IEEE Transactions on Automatic Control*, 1996.

[63] M. Di Bernardo, C. Budd, A. R. Champneys, and P. Kowalczyk. *Piecewise-smooth dynamical systems: theory and applications*, volume 163. Springer Science & Business Media, 2008.

[64] M. Di Bernardo, D. Liuzza, and G. Russo. Contraction analysis for a class of nondifferentiable systems with applications to stability and network synchronization. *SIAM Journal on Control and Optimization*, 52(5):3203–3227, 2014.

[65] M. W. M. G. Dissanayake and N. Phan-Thien. Neural-network-based approximations for solving partial differential equations. *communications in Numerical Methods in Engineering*, 10(3):195–201, 1994.

[66] B. Eysenbach and S. Levine. Maximum entropy rl (provably) solves some robust rl problems. *10th International Conference on Learning Representations (ICLR)*, 2022.

[67] S. M. Fallat and C. R. Johnson. *Totally nonnegative matrices*. Princeton university press, 2022.

[68] A. Farahmand, S. Nabi, P. Grover, and D. N. Nikovski. Learning to control partial differential equations: Regularized fitted Q-iteration approach. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 4578–4585, 2016.

[69] A. Farahmand, S. Nabi, and D. N. Nikovski. Deep reinforcement learning for partial differential equation control. In *2017 American Control Conference (ACC)*, pages 3120–3127, 2017.

[70] F. Felici. *Real-Time Control of Tokamak Plasmas: from Control of Physics to Physics-Based Control*. PhD thesis, EPFL, Lausanne, 2011.

[71] M. Fiedler. *Special matrices and their applications in numerical mathematics*. Courier Corporation, 2008.

[72] F. Forni and R. Sepulchre. A differential Lyapunov framework for contraction analysis. *IEEE Transactions on Automatic Control*, 59(3):614–628, 2013.

[73] F. Forni and R. Sepulchre. Differentially positive systems. *IEEE Transactions on Automatic Control*, 61(2):346–359, 2016.

[74] F. Forni and R. Sepulchre. A dissipativity theorem for p-dominant systems. In *IEEE 56th Conference on Decision and Control*, pages 3467–3472, 2017.

[75] F. Forni and R. Sepulchre. Differential dissipativity theory for dominance analysis. *IEEE Transactions on Automatic Control*, 64(6):2340–2351, 2019.

[76] M. Franceschelli and P. Frasca. Proportional dynamic consensus in open multi-agent systems. In *IEEE Conference on Decision and Control (CDC)*, pages 900–905. IEEE, 2018.

[77] B. A. Francis and W. M. Wonham. The internal model principle of control theory. *Automatica*, 12(5):457–465, 1976.

[78] V. Fromion and G. Scorletti. The behaviour of incrementally stable discrete time systems. *American Control Conference ACC*, 6:4563 – 4567, 1999.

[79] H. Fu, X. Chen, W. Wang, and M. Wu. Data-based optimal synchronization control for discrete-time nonlinear heterogeneous multiagent systems. *IEEE Transactions on Cybernetics*, 2020.

[80] S. Fujimoto, H. Hoof, and D. Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pages 1587–1596. PMLR, 2018.

[81] A. Gahlawat, M. M. Peet, and E. Witrant. Control and verification of the safety-factor profile in tokamaks using sum-of-squares polynomials. *IFAC Proceedings Volumes*, 44(1):12556–12561, 2011. 18th IFAC World Congress.

[82] A. Gahlawat, E. Witrant, M. M. Peet, and M. Alamir. Bootstrap current optimization in tokamaks using sum-of-squares polynomials. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 4359–4365, 2012.

[83] M. Giaccagli. *Ph.D. thesis: "Incremental stability and applications for nonlinear control systems"*. PhD thesis, Universitè Claude Bernard Lyon 1, 2022.

[84] M. Giaccagli, V. Andrieu, S. Tarbouriech, and D. Astolfi. Infinite gain margin, contraction and optimality: An LMI-based design. *European Journal of Control*, 68:100685, 2022.

[85] M. Giaccagli, D. Astolfi, and V. Andrieu. Further results on incremental input-to-state stability based on contraction-metric analysis. In *Submitted to IEEE 62nd Conference on Decision and Control (CDC)*, 2023.

[86] M. Giaccagli, D. Astolfi, V. Andrieu, and L. Marconi. Sufficient conditions for global integral action via incremental forwarding for input-affine nonlinear systems. *IEEE Transactions on Automatic Control*, 67(12):6537–6551, 2022.

[87] P. Giesl, S. Hafstein, and C. Kawan. Review on contraction analysis and computation of contraction metrics. *arXiv preprint arXiv:2203.01367*, 2022.

[88] M. E. Gilmore, C. Guiver, and H. Logemann. Semi-global incremental input-to-state stability of discrete-time Lur'e systems. *Systems & Control Letters*, 136:104593, 2020.

[89] C. Godsil and G. Royle. *Algebraic graph theory*. Springer, 2001.

[90] C. A. Gonzaga, M. Jungers, and J. Daafouz. Stability analysis of discrete-time Lur'e systems. *Automatica*, 48(9):2277–2283, 2012.

[91] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[92] S. Grammatico, A. Subbaraman, and A. R. Teel. Discrete-time stochastic control systems: A continuous Lyapunov function implies robustness to strictly causal perturbations. *Automatica*, 49(10):2939–2952, 2013.

[93] F. Gu, H. Yin, L. El Ghaoui, M. Arcak, P. Seiler, and M. Jin. Recurrent neural network controllers synthesis with stability guarantees for partially observed systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 5385–5394, 2022.

[94] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870. PMLR, 2018.

[95] M. Han, L. Zhang, J. Wang, and W. Pan. Actor-critic reinforcement learning for control with stability guarantee. *IEEE Robotics and Automation Letters*, 5(4):6217–6224, 2020.

[96] J. Hauser, S. Sastry, and P. Kokotovic. Nonlinear control via approximate input-output linearization: The ball and beam example. *IEEE Transactions on Automatic Control*, 1992.

[97] K. Hengster-Movric, K. You, F. L. Lewis, and L. Xie. Synchronization of discrete-time multi-agent systems on graphs using riccati design. *Automatica*, 49(2):414–423, 2013.

[98] H. J. C. Huijberts, T. Lilge, and H. Nijmeijer. Nonlinear discrete-time synchronization via extended observers. *International Journal of Bifurcation and Chaos*, 11(07):1997–2006, 2001.

[99] M. Igl, L. Zintgraf, T. A. Le, F. Wood, and S. Whiteson. Deep variational reinforcement learning for POMDPs. In *International Conference on Machine Learning*, pages 2117–2126. PMLR, 2018.

[100] F. Imbeaux, M. Lennholm, A. Ekedahl, P. Pastor, T. Aniel, S. Brémond, J. Decker, P. Devynck, R. Dumont, G. Giruzzi, P. Maget, D. Mazon, A. Merle, D. Molina, P. Moreau, F. Saint-Laurent, J. Ségui, D. Zarzoso, and T. S. Team. Real-time control of the safety factor profile diagnosed by magneto-hydrodynamic activity on the tore supra tokamak. *Nuclear Fusion*, 51(7):073033, jun 2011.

[101] A. Isidori. *Nonlinear Control Systems*. Springer, 1995.

[102] A. Isidori. *Lectures in feedback design for multivariable systems*. Springer, 2017.

[103] A. Isidori and C. I. Byrnes. Output regulation of nonlinear systems. *IEEE Transactions on Automatic Control*, 1990.

[104] A. Isidori, L. Marconi, and G. Casadei. Robust output synchronization of a network of heterogeneous nonlinear agents via nonlinear regulation theory. *IEEE Transactions on Automatic Control*, 59(10):2680–2691, 2014.

[105] S. Jafarpour, P. Cisneros-Velarde, and F. Bullo. Weak and semi-contraction for network systems and diffusively coupled oscillators. *IEEE Transactions on Automatic Control*, 67(3):1285–1300, 2022.

[106] Z. Jiang and Y. Wang. A converse Lyapunov theorem for discrete-time systems with disturbances. *Systems & Control Letters*, 45(1):49–58, 2002.

[107] J. Jouffroy. Some ancestors of contraction analysis. In *44th IEEE Conference on Decision and Control*, pages 5450–5455, 2005.

[108] J. Kato and T. Yoshizawa. A relationship between uniformly asymptotic stability and total stability. *Funkcialaj Ekvacioj*, 12:233–238, 1969.

[109] Y. Kawano and Y. Hosoe. Contraction analysis of discrete-time stochastic systems. *IEEE Transactions on Automatic Control*, pages 1–16, 2023.

[110] Y. Kawano and K. Kashima. An LMI framework for contraction-based nonlinear control design by derivatives of Gaussian process regression. *Automatica*, 151:110928, 2023.

[111] C. M. Kellett and A. R. Teel. Smooth Lyapunov functions and robustness of stability for difference inclusions. *Systems & Control Letters*, 52(5):395–405, 2004.

[112] C. M. Kellett and A. R. Teel. On the robustness of $\mathcal{KL}$-stability for difference inclusions: Smooth discrete-time Lyapunov functions. *SIAM Journal on Control and Optimization*, 44(3):777–800, 2005.

[113] H. K. Khalil. *Nonlinear systems*. Pearson, 3rd ed edition, 2002.

[114] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations (ICLR)*, 2015.

[115] B. Kiumarsi, F. L. Lewis, and Z. Jiang. H∞ control of linear discrete-time systems: Off-policy reinforcement learning. *Automatica*, 78:144–152, 2017.

[116] S. Knorn, Z. Chen, and R. H. Middleton. Overview: Collective control of multiagent systems. *IEEE Transactions on Control of Network Systems*, 3(4):334–347, 2015.

[117] V. Konda and J. Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

[118] K. Königsberger. *Analysis 2*. Springer-Verlag, 2004.

[119] L. Laborde, D. Mazon, D. Moreau, A. Murari, R. Felton, L. Zabeo, R. Albanese, M. Ariola, J. Bucalossi, F. Crisanti, M. de Baar, G. de Tommasi, P. de Vries, E. Joffrin, M. Lennholm, X. Litaudon, A. Pironti, T. Tala, and A. Tuccillo. A model-based technique for integrated real-time profile control in the JET tokamak. *Plasma Physics and Controlled Fusion*, 47(1):155–183, dec 2004.

[120] I. E. Lagaris, A. Likas, and D. I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks*, 9(5):987–1000, 1998.

[121] P. Lancaster and M. Tismenetsky. *The theory of matrices: with applications*. Elsevier, 1985.

[122] F. L. Lewis and D. Vrabie. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE circuits and systems magazine*, 9(3):32–50, 2009.

[123] J. Li, J. Ding, T. Chai, F. L. Lewis, and S. Jagannathan. Adaptive interleaved reinforcement learning: Robust stability of affine nonlinear systems with unknown uncertainty. *IEEE Transactions on Neural Networks and Learning Systems*, 33(1):270–280, 2022.

[124] M. Y. Li and J. S. Muldowney. On R.A. Smith's Autonomous Convergence Theorem. *Rocky Mountain Journal of Mathematics*, 25(1):365 – 378, 1995.

[125] Y. Li. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.

[126] Z. Li, Z. Duan, G. Chen, and L. Huang. Consensus of multiagent systems and synchronization of complex networks: A unified viewpoint. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 57(1):213–224, 2009.

[127] D. Liberzon. *Switching in systems and control*, volume 190. Springer, 2003.

[128] T. Lilge. Nonlinear discrete-time observers for synchronization problems. In *New Directions in nonlinear observer design*, pages 491–510. Springer, 1999.

[129] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *4th International Conference on Learning Representations (ICLR)*, 2016.

[130] D. Limon and T. Alamo. Tracking model predictive control. In *Encyclopedia of Systems and Control*. Springer, 2021.

[131] W. Lin. Feedback stabilization of general nonlinear control systems: a passive system approach. *Systems & Control Letters*, 25(1):41–52, 1995.

[132] W. Lin and C. I. Byrnes. Design of discrete-time nonlinear control systems via smooth feedback. *IEEE Transactions on Automatic Control*, 39(11):2340–2346, 1994.

[133] Z. Liu, M. Zhang, A. Saberi, and A. A. Stoorvogel. Passivity based state synchronization of homogeneous discrete-time multi-agent systems via static protocol in the presence of input delay. *European Journal of Control*, 41:16–24, 2018.

[134] W. Lohmiller and J. J. E. Slotine. On contraction analysis for non-linear systems. *Automatica*, 34(6):683–696, 1998.

[135] I. Loshchilov and F. Hutter. SGDR: Stochastic gradient descent with warm restarts. *5th International Conference on Learning Representations (ICLR)*, 2017.

[136] E. Maljaars, F. Felici, T. C. Blanken, C. Galperti, O. Sauter, M. R. De Baar, F. Carpanese, T. P. Goodman, D. Kim, S. H. Kim, M. Kong, B. Mavkov, A. Merle, J. M. Moret, R. Nouailletas, M. Scheffer, A. A. Teplukhina, N. M. T. Vu, E. MST1-team, and TCV-team. Profile control simulations and experiments on tcv: a controller test environment and results using a model-based predictive controller. *Nuclear Fusion*, 57(12):126063, 2017.

[137] E. Maljaars, F. Felici, M. R. De Baar, J. Van Dongen, G. M. D. Hogeweij, P. J. M. Geelen, and M. Steinbuch. Control of the tokamak safety factor profile with time-varying constraints using mpc. *Nuclear Fusion*, 55(2):023001, 2015.

[138] I. G. Malkin. *Stability Theory of Motion*. Nauka, Moscow, 1966.

[139] I. R. Manchester, M. Revay, and R. Wang. Contraction-based methods for stable identification and robust machine learning: a tutorial. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 2955–2962. IEEE, 2021.

[140] I. R. Manchester and J. J. E. Slotine. Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design. *IEEE Transactions on Automatic Control*, 62(6):3046–3053, 2017.

[141] P. Martin, S. Devasia, and B. Paden. A different look at output tracking: Control of a VTOL aircraft. *Automatica*, 1996.

[142] M. Mattioni, S. Monaco, and D. Normand-Cyrot. Forwarding stabilization in discrete time. *Automatica*, 109:108532, 2019.

[143] M. Mattioni, S. Monaco, and D. Normand-Cyrot. A new connection protocol for multi-consensus of discrete-time systems. In *American Control Conference (ACC)*, pages 5179–5184. IEEE, 2022.

[144] B. Mavkov, E. Witrant, and C. Prieur. Distributed control of coupled inhomogeneous diffusion in tokamak plasmas. *IEEE Transactions on Control Systems Technology*, 27(1):443–450, 2019.

[145] B. Mavkov, E. Witrant, C. Prieur, E. Maljaars, F. Felici, O. Sauter, and TCV-Team. Experimental validation of a Lyapunov-based controller for the plasma safety factor and plasma pressure in the TCV tokamak. *Nuclear Fusion*, 58(5):056011, mar 2018.

[146] W. Mei and F. Bullo. Lasalle invariance principle for discrete-time dynamical systems: A concise and self-contained tutorial. *arXiv preprint arXiv:1710.03710*, 2017.

[147] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[148] D. Moreau, J. F. Artaud, J. R. Ferron, C. T. Holcomb, D. A. Humphreys, F. Liu, T. C. Luce, J. M. Park, R. Prater, F. Turco, and M. L. Walker. Combined magnetic and kinetic control of advanced tokamak steady state scenarios based on semi-empirical modelling. *Nuclear Fusion*, 55(6):063011, 2015.

[149] P. Moreau, S. Bremond, D. Douai, A. Geraud, P. Hertout, M. Lennholm, D. Mazon, F. Saint-Laurent, and T. S. Team. Plasma control in tore supra. *Fusion Science and Technology*, 56(3):1284–1299, 2009.

[150] J. Morimoto and K. Doya. Robust reinforcement learning. *Neural computation*, 17(2):335–359, 2005.

[151] J. Muldowney. Compound matrices and ordinary differential equations. *Rocky Mountain Journal of Mathematics*, 20(4):857–872, 1990.

[152] D. Nešić, A. R. Teel, and D. Carnevale. Explicit computation of the sampling period in emulation of controllers for nonlinear sampled-data systems. *IEEE transactions on Automatic Control*, 54(3):619–624, 2009.

[153] D. Nešić, A. R. Teel, and P. V. Kokotović. Sufficient conditions for stabilization of sampled-data nonlinear systems via discrete-time approximations. *Systems & Control Letters*, 38(4-5):259–270, 1999.

[154] R. Ofir, M. Margaliot, Y. Levron, and J. J. Slotine. A sufficient condition for k-contraction of the series connection of two systems. *IEEE Transactions on Automatic Control*, 67(9):4994–5001, 2022.

[155] E. Ott. *Chaos in dynamical systems*. Cambridge university press, 2002.

[156] Y. Ou, C. Xu, and E. Schuster. Robust control design for the poloidal magnetic flux profile evolution in the presence of model uncertainties. *IEEE Transactions on Plasma Science*, 38(3):375–382, 2010.

[157] Y. Ou, C. Xu, E. Schuster, T. C. Luce, J. R. Ferron, M. L. Walker, and D. A. Humphreys. Optimal tracking control of current profile in tokamaks. *IEEE Transactions on Control Systems Technology*, 19(2):432–441, 2011.

[158] D. Paccagnan, B. Gentile, F. Parise, M. Kamgarpour, and J. Lygeros. Nash and wardrop equilibria in aggregative games with coupling constraints. *IEEE Transactions on Automatic Control*, 64(4):1373–1388, 2019.

[159] A. Pajares and E. Schuster. Nonlinear robust safety factor profile control in tokamaks via feedback linearization and nonlinear damping techniques. In *2018 IEEE Conference on Control Technology and Applications (CCTA)*, pages 306–311. IEEE, 2018.

[160] Y. Pan, A. Farahmand, M. White, S. Nabi, P. Grover, and D. Nikovski. Reinforcement learning with function-valued action spaces for partial differential equation control. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3986–3995. PMLR, 10–15 Jul 2018.

[161] E. Panteley, A. Loria, and S. Sukumar. Strict Lyapunov functions for consensus under directed connected graphs. In *2020 European Control Conference (ECC)*, pages 935–940. IEEE, 2020.

[162] A. Paszke et al. Pytorch: an imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[163] A. Pavlov, A. Pogromsky, N. van de Wouw, and H. Nijmeijer. Convergent dynamics, a tribute to Boris Pavlovich Demidovich. *Systems & Control Letters*, 52(3-4):257–261, 2004.

[164] A. Pavlov, E. Steur, and N. van de Wouw. Nonlinear integral coupling for synchronization in networks of nonlinear systems. *Automatica*, 140:110202, 2022.

[165] A. Pavlov and N. van de Wouw. Convergent discrete-time nonlinear systems: the case of PWA systems. In *2008 American Control Conference*, pages 3452–3457. IEEE, 2008.

[166] A. Pavlov, N. Van De Wouw, and H. Nijmeijer. *Uniform output regulation of nonlinear systems: a convergent dynamics approach*, volume 205. Springer, 2006.

[167] I. R. Petersen. A stabilization algorithm for a class of uncertain linear systems. *Systems & Control Letters*, 8(4):351–357, 1987.

[168] I. R. Petersen and C. V. Hollot. A Riccati equation approach to the stabilization of uncertain linear systems. *Automatica*, 22(4):397–411, 1986.

[169] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta. Robust adversarial reinforcement learning. In *International Conference on Machine Learning*, pages 2817–2826. PMLR, 2017.

[170] R. Postoyan, L. Buşoniu, D. Nešić, and J. Daafouz. Stability analysis of discrete-time infinite-horizon optimal control with discounted cost. *IEEE Transactions on Automatic Control*, 62(6):2736–2749, 2017.

[171] L. Praly, R. Ortega, and G. Kaliora. Stabilization of nonlinear systems via forwarding mod$\{L_gV\}$. *IEEE Transactions on Automatic Control*, 46(9):1461–1466, 2001.

[172] L. Praly and Y. Wang. Stabilization in spite of matched unmodeled dynamics and an equivalent definition of input-to-state stability. *Mathematics of Control, Signals and Systems*, 9(1):1–33, 1996.

[173] D. C. Psichogios and L. H. Ungar. A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, 38(10):1499–1511, 1992.

[174] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 1994.

[175] Z. Qu. Robust control of nonlinear uncertain systems under generalized matching conditions. *Automatica*, 29(4):985–998, 1993.

[176] A. Raffin, A. Hill, M. Ernestus, A. Gleave, A. Kanervisto, and N. Dormann. Stable baselines3. `https://github.com/DLR-RM/stable-baselines3`, 2019.

[177] M. Raissi, P. Perdikaris, and G. Karniadakis. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[178] B. S. Rüffer, N. Van De Wouw, and M. Mueller. Convergent systems vs. incremental stability. *Systems & Control Letters*, 62(3):277–285, 2013.

[179] A. Saberi, A. A. Stoorvogel, M. Zhang, and P. Sannuti. *Synchronization of Multi-Agent Systems in the Presence of Disturbances and Delays*. Springer Nature, 2022.

[180] L. Salvadori and A. Schiaffino. On the problem of total stability. *Nonlinear Analysis: Theory, Methods & Applications*, 1(3):207–213, 1977.

[181] R. G. Sanfelice and L. Praly. Convergence of Nonlinear Observers on $\mathbb{R}^n$ With a Riemannian Metric (Part I). *IEEE Transactions on Automatic Control*, 57(7):1709–1722, 2012.

[182] O. Sauter, C. Angioni, and Y. R. Lin-Liu. Neoclassical conductivity and bootstrap current formulas for general axisymmetric equilibria and arbitrary collisionality regime. *Physics of Plasmas*, 6(7):2834–2839, 1999.

[183] L. Scardovi and R. Sepulchre. Synchronization in networks of identical linear systems. In *47th IEEE Conference on Decision and Control*, pages 546–551, 2008.

[184] L. Scardovi and R. Sepulchre. Synchronization in networks of identical linear systems. *Automatica*, 45(11):2557–2562, 2009.

[185] C. Scherer, P. Gahinet, and M. Chilali. Multiobjective output-feedback control via LMI optimization. *IEEE Transactions on Automatic Control*, 42(7):896–911, 1997.

[186] J. Schmidhuber. *Making the World Differentiable: On Using Self-Supervised Fully Recurrent Neural Networks for Dynamic Reinforcement Learning and Planning in Non-Stationary Environments*. Inst. für Informatik, 1990.

[187] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[188] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[189] G. Seifert. On total stability and asymptotic stability. *Tôhoku Math. Journ.*, 19(1), 1967.

[190] J. Seo, Y. S. Na, B. Kim, C. Y. Lee, M. S. Park, S. J. Park, and Y. H. Lee. Feedforward beta control in the KSTAR tokamak by deep reinforcement learning. *Nuclear Fusion*, 61(10):106010, sep 2021.

[191] R. Sepulchre, M. Jankovic, and P. V. Kokotovic. *Constructive nonlinear control*. Springer Science & Business Media, 2012.

[192] A. Serrani, A. Isidori, and L. Marconi. Semi-global nonlinear output regulation with adaptive internal model. *IEEE Transactions on Automatic Control*, 46(8):1178–1194, 2001.

[193] D. Sheng and G. Fazekas. A feature learning siamese model for intelligent control of the dynamic range compressor. In *IEEE International Joint Conference on Neural Networks*, pages 1–8, 2019.

[194] D. D. Šiljak and D. M. Stipanovic. Robust stabilization of nonlinear systems: The LMI approach. *Mathematical problems in Engineering*, 6(5):461–493, 2000.

[195] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

[196] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. PMLR, 2014.

[197] J. Simpson-Porco and F. Bullo. Contraction theory on Riemannian manifolds. *Systems & Control Letters*, 65:74–80, 2014.

[198] J. W. Simpson-Porco. Low-gain stability of projected integral control for input-constrained discrete-time nonlinear systems. *IEEE Control Systems Letters*, 6:788–793, 2021.

[199] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry. Kalman filtering with intermittent observations. *IEEE Transactions on Automatic Control*, 49(9):1453–1464, 2004.

[200] J. J. E. Slotine and W. Wang. A study of synchronization and group cooperation using partial contraction theory. In *Cooperative control*, pages 207–228. Springer, 2005.

[201] R. A. Smith. The poincaré–bendixson theorem for certain differential equations of higher order. *Proceedings of the Royal Society of Edinburgh Section A: Mathematics*, 83(1-2):63–79, 1979.

[202] T. Söderström and P. Stoica. *System Identification*. Prentice Hall International, 1989.

[203] E. Sontag. Contractive systems with inputs. In *Perspectives in mathematical system theory, control, and signal processing*, pages 217–228. Springer, 2010.

[204] E. D. Sontag and Y. Wang. On characterizations of the input-to-state stability property. *Systems & Control Letters*, 1995.

[205] G. Stan and R. Sepulchre. Analysis of interconnected oscillators by dissipativity theory. *IEEE Transactions on Automatic Control*, 52(2):256–270, 2007.

[206] A. A. Stoorvogel, A. Saberi, M. Zhang, and Z. Liu. Solvability conditions and design for synchronization of discrete-time multiagent systems. *International Journal of Robust and Nonlinear Control*, 28(4):1381–1401, 2017.

[207] T. Strom. On logarithmic norms. *SIAM Journal on Numerical Analysis*, 12(5):741–753, 1975.

[208] T. Stykel. Stability and inertia theorems for generalized Lyapunov equations. *Linear Algebra and its Applications*, 355(1-3):297–314, 2002.

[209] D. Sun, S. Jha, and C. Fan. Learning certified control using contraction metric. In *Conference on Robot Learning*, pages 1519–1539, 2021.

[210] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[211] R. S. Sutton, D. A. McAllester, S. P. Singh, Y. Mansour, et al. Policy gradient methods for reinforcement learning with function approximation. In *NIPs*, volume 99, pages 1057–1063. Citeseer, 1999.

[212] S. Takashi. *Riemannian geometry*, volume 149. American Mathematical Soc., 1996.

[213] S. Tarbouriech and M. Turner. Anti-windup design: an overview of some recent advances and open problems. *IET control theory & applications*, 3(1):1–19, 2009.

[214] C. Tessler, Y. Efroni, and S. Mannor. Action robust reinforcement learning and applications in continuous control. In *International Conference on Machine Learning*, pages 6215–6224. PMLR, 2019.

[215] D. N. Tran, B. S. Rüffer, and C. M. Kellett. Incremental stability properties for discrete-time systems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 477–482. IEEE, 2016.

[216] D. N. Tran, B. S. Rüffer, and C. M. Kellett. Convergence properties for discrete-time nonlinear systems. *IEEE Transactions on Automatic Control*, 64(8):3415–3422, 2019.

[217] N. M. Trang Vu, R. Nouailletas, E. Maljaars, F. Felici, and O. Sauter. Plasma internal profile control using ida-pbc: Application to tcv. *Fusion Engineering and Design*, 123:624–627, 2017. Proceedings of the 29th Symposium on Fusion Technology (SOFT-29) Prague, Czech Republic, September 5-9, 2016.

[218] H. Tsukamoto and S. Chung. Neural contraction metrics for robust estimation and control: A convex optimization approach. *IEEE Control Systems Letters*, 2020.

[219] H. Tsukamoto, S. Chung, and J. Slotine. Contraction theory for nonlinear stability analysis and learning-based control: A tutorial overview. *Annual Reviews in Control*, 52:135–169, 2021.

[220] G. Valmorbida and J. Anderson. Region of attraction estimation using invariant sets and rational lyapunov functions. *Automatica*, 75:37–45, 2017.

[221] F. J. Vargas and R. A. González. On the existence of a stabilizing solution of modified algebraic Riccati equations in terms of standard algebraic Riccati equations and linear matrix inequalities. *IEEE Control Systems Letters*, 4(1):91–96, 2020.

[222] T. Wakatsuki, T. Suzuki, N. Hayashi, N. Oyama, and S. Ide. Safety factor profile control with reduced central solenoid flux consumption during plasma current ramp-up phase using a reinforcement learning technique. *Nuclear Fusion*, 59(6):066022, may 2019.

[223] T. Wakatsuki, M. Yoshida, E. Narita, T. Suzuki, and N. Hayashi. Simultaneous control of safety factor profile and normalized beta for jt-60sa using reinforcement learning. *Nuclear Fusion*, 63(7):076017, 2023.

[224] M. L. Walker, P. De Vries, F. Felici, and E. Schuster. Introduction to tokamak plasma control. In *2020 American Control Conference (ACC)*, pages 2901–2918, 2020.

[225] S. Wang, E. Witrant, and D. Moreau. Robust control of q-profile and βp using data-driven models on east. *Fusion Engineering and Design*, 162:112071, 2021.

[226] W. Wang and J. E. Slotine. On partial contraction analysis for coupled nonlinear oscillators. *Biological cybernetics*, 92(1):38–53, 2005.

[227] L. Wei, R. Mccloy, and J. Bao. Control contraction metric synthesis for discrete-time nonlinear systems. *IFAC-PapersOnLine*, 54(3):661–666, 2021.

[228] L. Wei, R. McCloy, and J. Bao. Discrete-time contraction-based control of nonlinear systems with parametric uncertainties using neural networks. *Computers & Chemical Engineering*, 166:107962, 2022.

[229] J. Wesson and D. J. Campbell. *Tokamaks*, volume 149. Oxford university press, 2011.

[230] T. Wijnands, D. Van Houtte, G. Martin, X. Litaudon, and P. Froissard. Feedback control of the current profile on tore supra. *Nuclear Fusion*, 37(6):777–791, jun 1997.

[231] F. W. Wilson. The structure of the level surfaces of a Lyapunov function. *Journal of Differential Equations*, 3:323–329, 1967.

[232] E. Witrant and S. Brémond. Shape identification for distributed parameter systems and temperature profiles in tokamaks. In *50th IEEE Conference on Decision and Control and European Control Conference*, pages 2626–2631, 2011.

[233] E. Witrant, E. Joffrin, S. Bremond, G. Giruzzi, D. Mazon, O. Barana, and P. Moreau. A control-oriented model of the current profile in tokamak plasma. *Plasma Physics and Controlled Fusion*, 49:1075–1105, 2007.

[234] M. Wondergem, E. Lefeber, K. Pettersen, and H. Nijmeijer. Output feedback tracking of ships. *IEEE Transactions on Control Systems Technology*, 2010.

[235] C. Wu and D. V. Dimarogonas. From partial and horizontal contraction to k-contraction. *arXiv preprint:2208.14379*, 2022.

[236] C. Wu, I. Kanevskiy, and M. Margaliot. k-contraction: Theory and applications. *Automatica*, 136:110048, 2022.

[237] Y. Wu and Q. Zou. Robust inversion-based 2-DOF control design for output tracking: Piezoelectric-actuator example. *IEEE Transactions on Control Systems Technology*, 2009.

[238] X. J. Xie and N. Duan. Output tracking of high-order stochastic nonlinear systems with application to benchmark mechanical system. *IEEE Transactions on Automatic Control*, 2010.

[239] B. Yi, R. Wang, and I. Manchester. Reduced-order nonlinear observers via contraction analysis and convex optimization. *IEEE Transactions on Automatic Control*, 2021.

[240] H. Yin, B. Jayawardhana, and R. Reyes-Báez. Pinning synchronization of heterogeneous multi-agent nonlinear systems via contraction analysis. *IEEE Control Systems Letters*, 2021.

[241] H. Yin, P. Seiler, and M. Arcak. Stability analysis using quadratic constraints for systems with neural network controllers. *IEEE Transactions on Automatic Control*, 67(4):1980–1987, 2022.

[242] M. Zamani and P. Tabuada. Backstepping design for incremental stability. *IEEE Transactions on Automatic Control*, 56(9):2184–2189, 2011.

[243] F. Zhang, H. Trentelman, and J. Scherpen. Fully distributed robust synchronization of networked Lur'e systems with incremental nonlinearities. *Automatica*, 50(10):2515–2526, 2014.

[244] N. Zhang and N. Capel. Leoc: A principled method in integrating reinforcement learning and classical control theory. In *Learning for Dynamics and Control*. PMLR, 2021.

[245] P. Zhao, A. Lakshmanan, K. Ackerman, A. Gahlawat, M. Pavone, and N. Hovakimyan. Tube-certified trajectory tracking for nonlinear systems with robust control contraction metrics. *IEEE Robotics and Automation Letters*, 2022.

[246] K. Zhou and J. C. Doyle. *Essentials of robust control*. Prentice hall Upper Saddle River, NJ, 1998.

[247] Z. Zuo, J. Wang, and L. Huang. Robust stabilization for non-linear discrete-time systems. *International Journal of Control*, 77(4):384–388, 2004.

Own References

[S1] A. Cecilia, **S. Zoboli**, U. Serres, D. Astolfi, and V. Andrieu. LMI conditions for k-contraction: analysis and design. *Under review on IEEE Transactions on Automatic Control*, 2023.

[S2] M. Giaccagli, **S. Zoboli**, D. Astolfi, V. Andrieu, and G. Casadei. Synchronization in Networks of Nonlinear Systems: Contraction Analysis via Riemannian Metrics and Deep-Learning for Feedback Estimation. *Under review on IEEE Transactions on Automatic Control. Preprint: hal-03801100*, 2023.

[S3] A. Mattioni, **S. Zoboli**, B. Mavkov, D. Astolfi, V. Andrieu, E. Witrant, P. Frasca, and C. Prieur. Enhancing deep reinforcement learning with integral action to control tokamak safety factor. *Fusion Engineering and Design*, 2023.

[S4] **S. Zoboli**, V. Andrieu, D. Astolfi, G. Casadei, J. S. Dibangoye, and M. Nadri. Reinforcement learning policies with local LQR guarantees for nonlinear discrete-time systems. In *60th IEEE Conference on Decision and Control (CDC)*, pages 2258–2263. IEEE, 2021.

[S5] **S. Zoboli**, D. Astolfi, and V. Andrieu. Total stability of equilibria motivates integral action in discrete-time nonlinear systems. *Automatica*, 155, 2023.

[S6] **S. Zoboli**, D. Astolfi, V. Andrieu, G. Casadei, and L. Zaccarian. Incremental stabilization and multi-agent synchronization of discrete-time nonlinear systems. *Under review on IEEE Transactions on Automatic Control*, 2023.

[S7] **S. Zoboli**, A. Cecilia, U. Serres, D. Astolfi, and V. Andrieu. LMI conditions for k-contraction analysis: a step towards design. In *Accepted at 62nd IEEE Conference on Decision and Control (CDC)*, 2023.

[S8] **S. Zoboli**, S. Janny, and M. Giaccagli. Deep learning-based output tracking via regulation and contraction theory. In *22nd IFAC World Congress*, 2023.