UNIVERSITY OF CALIFORNIA

Los Angeles

Privacy in Control over the Cloud
and Learning to Control From Expert Demonstrations

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical and Computer Engineering

by

Alimzhan Sultangazin

2022

ABSTRACT OF THE DISSERTATION

Privacy in Control over the Cloud
and Learning to Control From Expert Demonstrations

by

Alimzhan Sultangazin
Doctor of Philosophy in Electrical and Computer Engineering
University of California, Los Angeles, 2022
Professor Paulo Tabuada, Chair

In this thesis, we consider two problems relevant to the control of complex closed-loop systems. In the first chapter, we focus on the implications that control over the cloud has for privacy of control systems and propose a method that protects privacy without sacrificing control performance. In the second chapter, we revisit the problem of learning a controller from a finite number of demonstrations, while guaranteeing stability.

The first chapter considers the following question: ”Given the need to offload control of a system to a third-party (i.e., a cloud), can we still guarantee the privacy of information about the said system and its control objective?” Cloud computing platforms are being increasingly used for closing feedback control loops [1–3], especially when computationally expensive algorithms, such as model-predictive control, are used to optimize performance. Outsourcing of control algorithms entails an exchange of data between the control system and the cloud, and, naturally, raises concerns about the privacy of the control system's data (e.g., state trajectory, control objective). Moreover, any attempt at enforcing privacy needs to add minimal computational overhead to avoid degrading control performance. We propose several transformation-based methods for enforcing data privacy. We also quantify the amount of provided privacy and discuss how much privacy is lost when the adversary has access to side knowledge. We address three different scenarios: a) the cloud has no knowledge about the system being controlled; b) the cloud knows what sensors and actuators the system employs but not the system dynamics; c) the cloud knows the system dynamics, its sensors, and actuators. In all of these three scenarios, the proposed methods allow for the control over the cloud without compromising private information (which information is considered private depends on the considered scenario).

The second chapter addresses the problem of learning control from expert demonstrations. Learning control from expert demonstrations is useful for control tasks, where providing examples of the desired behaviour is easier than defining such behaviour formally (e.g., driving a car comfortably). This problem has been addressed in the literature by using tools from statistical machine learning [4–6]. However, many of the methods proposed in the literature lack formal guarantees on stability and safety. Using tools from control theory and by first focusing on feedback linearizable systems, we show how to combine expert demonstrations into a stabilizing controller, provided that demonstrations are sufficiently long and there are at least $n+1$ of them, where n is the number of states of the system being controlled. When we have more than $n+1$ demonstrations, we discuss how to optimally choose the best $n+1$ demonstrations to construct the stabilizing controller. We then extend these results to a class of systems that can be embedded into a higher-dimensional system containing a chain of integrators. The feasibility of the proposed algorithm is demonstrated by applying it on a CrazyFlie 2.0 quadrotor.

The dissertation of Alimzhan Sultangazin is approved.

Tetsuya Iwasaki

Lieven Vandenberghe

Lin Yang

Paulo Tabuada, Committee Chair

University of California, Los Angeles

2022

Men osy jūmysty,

azattyq jolynda myñ ölıp, myñ tırılgen,

Qazaqstan halqyna arnaimyn.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like first to thank the National Science Foundation, the Army Research Laboratory, and the CONIX Research Center for supporting the work, which has resulted in this dissertation.

Over the five years of my PhD program, I was blessed to have been surrounded by the kindest and most wonderful people, who made this seemingly arduous experience into the one that I can look back to with a smile. My dissertation would be incomplete without me thanking them and reflecting on our time together.

I want to first thank my advisor, Prof. Paulo Tabuada, for his patient guidance through every stage of my PhD program – from the conception of every project to this very dissertation. Working alongside with him has taught me many things. But, most importantly, it has taught me that, while having solid scientific results is necessary, as much thought needs to be put into the organization and presentation of these results. His intellectual contributions and detailed comments, both scientific and stylistic, helped me shape this dissertation into what it is today. While in research Paulo often likes to play the role of the devil's advocate, in life he is truly his students' ally. Throughout my PhD, I always felt supported and heard by him. Thank you for everything, Paulo!

Next, I would like to thank my PhD dissertation committee members, Prof. Tetsuya Iwasaki, Prof. Lin Yang, and Prof. Lieven Vandenberghe, for agreeing to serve on my committee and for their invaluable feedback. I would also like to thank Prof. Christina Fragouli and Prof. Suhas Diggavi with whom I had the pleasure to collaborate and from whom I learnt a great deal about network analysis and probabilistic modelling. I would also be remiss if I did not thank everyone at the Electrical and Computer Engineering Office of Graduate Student Affairs for guiding me through the necessary procedures ever since my admission into the program. Thank you, Deeona, Ylena, and Julio. Thank you, Ryo Arreola, you will be sorely missed by all of us.

At this point, I want to thank everyone at the Cyber-Physical Systems Laboratory, which has, in many ways, become my home away from home. Tzanis Anevlavis, Marcus Lucas, Lucas Fraile, Omar Hussein, Yanwen Mao, Calvin John, Cory Ye, Jonathan Bunton, Matteo Marchi, Muratkhan Abdirash, Carlos Murguia, Yskandar Gas, and Luigi Pannocchi - I am proud to have worked with brilliant individuals like you all, but, more importantly, I am fortunate to have the great privilege of having you as my friends. Your friendship and support have inspired me to go on through the hard times and all the way through the pandemic. I want to extend special thanks to my co-authors, Lucas Fraile and Luigi Pannocchi, for their invaluable work and support.

My PhD experience would not have been what it was without all the friends I met along the way. I would like to thank Eden Haney, Josh Hannan, Alibek Danyalov, Javier Mercado, Ingrid Mattinger, Muhammed Veli, Artem Goncharov, and Anastasios Papathanasopoulos for the wonderful times we shared during my PhD program. I am very grateful to have met you. My special and heartfelt thanks go out to my friends and roommates, Assel Seitbekova and Ablaikhan Akhazhanov. Thank you for all the good times and your support. I would also like to thank Alexandra Elbakyan and everyone who provides open access to otherwise prohibitively expensive scientific material.

Finally, I would like to thank my loved ones for always being there for me when I needed them. This dissertation would not have been possible without the support of my parents, Anuarbek Sultangazin and Agniya Kadirova, and my partner, Azhar Dyussekenova. Aramyzda teñızder men kontinentter bolğanda da, senderdıñ jyly sözderıñ jäñe şeksız mahabattaryñ menı jumysymdy jalğasuğa şabyttandyrdy. Ol üşın senderge ülken raqmet.

VITA

2017	B.Eng (Electrical and Electronics Engineering) Nazarbayev University, Astana, Kazakhstan.
2019	M.S. (Electrical and Computer Engineering) University of California, Los Angeles, CA, USA.
2018-present	Graduate Student Researcher, Department of Electrical and Computer Engineering, University of California, Los Angeles, CA, USA.
2021-present	Ph.D Candidate (Electrical and Computer Engineering) University of California, Los Angeles, CA, USA.

PUBLICATIONS

Alimzhan Sultangazin, Luigi Pannocchi, Lucas Fraile, and Paulo Tabuada, "Learning to Control From Expert Demonstrations," (submitted to) *IEEE Transactions on Automatic Control*.

Alimzhan Sultangazin, Luigi Pannocchi, Lucas Fraile, and Paulo Tabuada, "Watch and Learn: Learning to control feedback linearizable systems from expert demonstrations," in *Proceedings of the 39th IEEE International Conference on Robotics and Automation (ICRA)*, 2022, to appear.

Alimzhan Sultangazin, Lucas Fraile and Paulo Tabuada, "Exploiting the experts: Learning to control unknown SISO feedback linearizable systems from expert demonstrations," in *Proceedings of the 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 5789-5794.

Alimzhan Sultangazin and Paulo Tabuada, ”Symmetries and Isomorphisms for Privacy in Control Over the Cloud,” *IEEE Transactions on Automatic Control*, vol. 66, no. 2, pp. 538-549, Feb. 2021.

Alimzhan Sultangazin and Paulo Tabuada, ”Symmetries and privacy in control over the cloud: uncertainty sets and side knowledge,” in *Proceedings of the 58th IEEE Conference on Decision and Control (CDC)*, 2019, pp. 7209-7214.

Alimzhan Sultangazin and Paulo Tabuada, ”Towards the use of Symmetries to Ensure Privacy in Control Over the Cloud,” in *Proceedings of the 57th IEEE Conference on Decision and Control (CDC)*, 2018, pp. 5008-5013.

Alimzhan Sultangazin, Suhas Diggavi and Paulo Tabuada, ”Protecting the Privacy of Networked Multi-Agent Systems Controlled over the Cloud,” in *Proceedings of the 27th International Conference on Computer Communication and Networks (ICCCN)*, 2018, pp. 1-7.

CHAPTER 1

Using isomorphisms for privacy in control over the cloud

1.1 Introduction

1.1.1 Motivation

The recent advances in reliability and speed of communication have led to an increased use of cloud-based services, which provide computation and data storage capabilities to clients. Control over the cloud [1–3] has numerous advantages, which include easier installation and maintenance [9], and the availability of global information from all of the cloud's clients when making control decisions. However, the main advantage of control over the cloud is that it allows control systems to outsource expensive computational tasks to the cloud, thus potentially improving the speed of computation and freeing the local computational capabilities for other tasks.

An illustrative example of the benefits of outsourcing computing can be observed in Model Predictive Control (MPC). MPC is a conceptually simple, yet powerful scheme that was adopted in industry for multivariable control [10]. MPC inherently involves solving complex constrained optimization problems *on-line* (i.e., within one sampling interval). The work in [1] presents an experimental study that shows feasibility of MPC over the cloud for robot control. Another work (see [2]) considered the practicality and benefits of cloud-based MPC for a large-scale solar plant. The availability of global information provided by control over the cloud can have many practical benefits, as shown in [3]. There, the authors propose a solution to the problem of traffic flow estimation via the cloud.

However, relying on a third-party to perform computation is not without its dangers. Despite the benefits of control over the cloud, a number of studies have shown that exposing existing systems to connectivity may lead to security vulnerabilities in a vast variety of applications [11–14], including control of process plants, traffic infrastructure, and smart meter systems. Cyber-security attacks vary based on the amount of resources the attacker possesses [15]. One of the most basic attacks that requires little resources is eavesdropping. It can often serve as a stepping stone in the implementation of more complex attacks [16]. In control over the cloud, eavesdropping involves the adversary listening in to the communication channel between sensors, controllers, and actuators to leak valuable information about the model, the controller, and trajectories [17]. The client is expected to disclose all of this sensitive information to the cloud if it intends to receive valid control inputs from it. For example, we would expect drivers to share their locations, final destinations and, perhaps, dynamics to successfully allow traffic control over the cloud.

Eavesdropping attacks are usually prevented with encryption - the plant and the cloud establish a shared key with which they encrypt transmitted messages and decrypt the received ones. However, if the adversary manages to undermine the security of the cloud (e.g., gain unauthorized access to its memory), this technique can no longer protect the system since the cloud accesses the decrypted data. As stated in [18], traditional IT security provides only a partial solution. Therefore, there is a pressing need for development of control-over-the-cloud methods that do not rely on decryption of the incoming data. Although much effort has been directed to this problem, a universally secure scheme for control over the cloud that could support any client functionality has not yet been created [19, 20]. When solving the problem of private control over the cloud, two other important concerns need to be accounted for: efficiency and safety. Privacy cannot come at the cost of degradation of control performance either due to delays in the feedback loop or inaccurate control inputs.

1.1.2 Related work

The body of work on privacy in control over the cloud can be categorized into methods based on homomorphic encryption, differential privacy, and algebraic transformations.

When using homomorphic encryption techniques, the cloud is able to perform the computations on encrypted data without the need to decrypt it [21]. Homomorphic encyption can be classified into fully homomorphic encryption (FHE), which allows arbitrary computations on encypted data, and partially homomorphic encryption (PHE), which only allows for a subset of operations (e.g., modular multiplication) on encrypted data. Using PHE for control over the cloud with encrypted controllers was proposed in [22, 23]. In an effort to reduce communication with the cloud, in [24] the authors suggest using FHE for controller encryption. However, longer execution times of FHE [21] make it less practical than PHE when using optimization for control over the cloud. While PHE methods are shown to be feasible and are able to provide privacy guarantees [9,17,20,25–27], the execution time, which grows disproportionally with an increase in key length [17, 20], remains a valid concern in these methods. A consequence of this is that using homomorphic encrypion may potentially lead to instability in the controlled system due to processing delays. To address this problem, some works (see [17]) have shown that encryption parameters can be chosen to ensure stability of the closed-loop performance, thus providing a natural trade-off between security and control performance. The practical feasibility of encrypted control systems has been validated in [28] by considering control of a DC motor in real time.

Inspired by studies in privacy of databases, the problem of privacy in control over the cloud has also been approached from the standpoint of differential privacy (see [29,30]). This technique ensures that the risk of losing privacy of a single user's data by means of data queries is low. The main idea of these methods is to perturb the response to a data query with appropriate noise [31]. However, to achieve more privacy, the user must sacrifice accuracy (i.e., add more noise), which, in the context of control, degrades the control performance.

The ideas behind algebraic transformation methods have initially stemed from works on privacy in optimization. The idea is to use algebraic transformations to produce a different, but equivalent optimization problem. In other words, although the cloud does not know the original optimization problem, it can provide the client with an optimal solution to an equivalent optimization problem from which the client is able to recover the optimal solution to the original problem. Although initially these methods found application exclusively in linear programs [32, 33], several efforts have been directed to providing a unified framework and generalizing them to convex optimization problems (see [34, 35]). The work in [34] also shows one of the first attempls to define and *quantify* privacy of transformation-based methods. Algebraic transformation methods found applications in control due to their efficiency and guaranteed optimality of the solution [35]. For example, in [36] the authors propose a hybrid transformation-based method to preserve privacy of an MPC controller in networked control systems. In [37], transformation-based methods are used to provide privacy in a specific problem AC Optimal Power Flow.

1.1.3 Contributions

This chapter focuses on the use of transformation-based methods to preserve privacy of the system dynamics, control objective and constraints, and system trajectories. The contributions of this chapter are fourfold:

1. we propose using isomorphisms and symmetries of control systems as a source of transformations so as to keep data private;

2. we quantify the privacy guaranteed by these methods via the dimension of the set that describes the uncertainty experienced by the adversary;

3. we quantify how much privacy is lost when the adversary is assumed to have access to side knowledge;

4. we show that the proposed method is computationally light as it only requires matrix multiplications.

The method proposed in this chapter was initially introduced in [38]. In [39], it was extended to networked control systems with several agents requesting control input from a single cloud. In [40], the dimension of the set describing the uncertainty experienced by the adversary was proposed as a measure of privacy for this method and was evaluated for the special case of free group actions. This chapter provides a unified presentation of the results in [38, 40] with simpler proofs and several new results, such as the bounds on privacy when the group action is not free and an exact quantification of privacy for prime systems. The content of this chapter has been published in [41].

While privacy quantification in optimization has been studied in [35], this work considers how much privacy is preserved in the more challenging context of control. Moreover, the measure of privacy proposed in this work has been chosen to be suitable for problems of optimization in control systems and, therefore, is different from any of those proposed in [35]. Although the application of transformation-based methods in control has been previously discussed in [36], the scheme proposed there only considers a special case, where the cloud optimizes the weighted sum of the norms of the input and state, and the state is taken to be the output of the system. Our algorithm can be applied to a wider class of problems as we allow for arbitrary quadratic costs, linear constraints and outputs different from the state.

The proposed results do not address the case where the adversary has some belief about the structure or the range of values of the system parameters. Addressing the adversary's beliefs is likely to be more natural in a probabilistic/information-theoretic setup that is outside of the scope of this chapter, where we only employ deterministic techniques.

1.2 Problem Definition

1.2.1 Plant dynamics and control objective

We consider discrete-time affine plants, denoted by Σ, and described by:

$$\Sigma : \quad \begin{aligned} \bar{x}_{k+1} &= \bar{A}\bar{x}_k + \bar{B}u_k + \bar{c} \\ \bar{y}_k &= \bar{C}\bar{x}_k + \bar{d}, \end{aligned} \tag{1.1}$$

where $\bar{A} \in \mathbb{R}^{n\times n}$, $\bar{B} \in \mathbb{R}^{n\times m}$, $\bar{C} \in \mathbb{R}^{p\times n}$, $\bar{c} \in \mathbb{R}^n$, and $\bar{d} \in \mathbb{R}^p$ describe the dynamics of the system, and $\bar{x}_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$ and $\bar{y}_k \in \mathbb{R}^p$ denote the state, input and output of the system at time k, respectively. We assume that system Σ is controllable and observable. We also assume, without loss of generality, that $\ker \bar{B} = \{0\}$ and $\operatorname{Im} \bar{C} = \mathbb{R}^p$, since we can always eliminate linearly dependent columns (resp. rows) from $\bar{B}$ (resp. $\bar{C}$).

To simplify notation, we lift every affine map $Wx+v$ to a linear map through the following construction:

$$Wx + v \mapsto \begin{bmatrix} W & v \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix}. \tag{1.2}$$

Applying (1.2) to (1.1):

$$\begin{aligned} x_{k+1} \triangleq \begin{bmatrix} \bar{x}_{k+1} \\ 1 \end{bmatrix} &= \begin{bmatrix} \bar{A} & \bar{c} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{x}_k \\ 1 \end{bmatrix} + \begin{bmatrix} \bar{B} \\ 0 \end{bmatrix} u_k \\ &\triangleq Ax_k + Bu_k \\ y_k \triangleq \begin{bmatrix} \bar{y}_k \\ 1 \end{bmatrix} &= \begin{bmatrix} \bar{C} & \bar{d} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \bar{x}_k \\ 1 \end{bmatrix} \triangleq Cx_k. \end{aligned} \tag{1.3}$$

In the remainder of the chapter we suppress the inner structure for simplicity and represent all the systems in the linear form (1.3). However, the reader is advised to remember that we are dealing with affine maps. This is also true for the affine maps we will use to define isomorphisms.

We refer to system (1.3) as the triple $\Sigma = (A, B, C)$. We call a triple $\{x_k, u_k, y_k\}_{k\in\mathbb{N}}$ a trajectory of Σ if it satisfies (1.1) for all $k \in \mathbb{N}$.

Additionally, we define a cost function $J : \mathbb{R}^n \times (\mathbb{R}^m)^{N+1} \to \mathbb{R}$ for $N \in \mathbb{N} \cup \{+\infty\}$ that allows to compare trajectories and, thus, to formulate different control objectives. In alignment with the linear framework, we consider quadratic cost functions given by:

$$J(x, u) = \sum_{i=0}^{N} \Delta\eta_i^T M \Delta\eta_i, \tag{1.4}$$

where $\Delta\eta_i = \begin{bmatrix} x_i - x_i^* & u_i - u_i^* \end{bmatrix}^T$, $x = \{x_0, ..., x_N\}$ and $u = \{u_0, ..., u_N\}$. The sequences $x^* = \{x_0^*, ..., x_N^*\}$ and $u^* = \{u_0^*, ..., u_N^*\}$ denote the reference trajectories to be tracked. We define $M \in \mathbb{R}^{(n+m+1)\times(n+m+1)}$ to be a positive-definite matrix. Due to the lift (1.2), this cost includes not only quadratic, but also linear terms.

In addition to a cost, we also consider control objectives that require certain constraints to be satisfied at all times. These constraints are defined as:

$$D\eta_i \leq 0, \quad \forall i \in \{0, 1, ..., N\}, \tag{1.5}$$

where $\eta_i = \begin{bmatrix} x_i & u_i \end{bmatrix}^T$ and $D \in \mathbb{R}^{h\times(n+m+1)}$. Note that, despite appearing to be linear constraints, the constraints above are in fact affine, in view of the construction (1.2).

1.2.2 Attack model and privacy objectives

The cloud is treated as a curious but honest adversary: the cloud adheres to the computations prescribed by an agreed-upon protocol, but may seek to extract and leak confidential information by keeping record of all computations and communicated messages.

The interaction between the plant and the cloud is performed in two steps. During the first step, called the handshaking, the plant provides the cloud with a suitably modified version of the plant model, cost, and constraints. In exchange, the cloud agrees to compute the input minimizing the provided cost, subject to the constraints and plant dynamics. During the second step, called plant execution, the plant repeatedly sends a suitably modified version of its measurements to the cloud. The cloud computes a new input based on the received measurements and sends it to the plant, where it is suitably modified before being applied to the plant.

In the previous paragraph we purposely used the vague expression "suitably modified". Making this expression more concrete requires that we first define the knowledge available to the plant. We consider the following three scenarios.

Problem 1.2.1 (Scenario 1)**.** *Assuming the cloud has no knowledge about the plant:*

1. *how to modify the plant* (A, B, C)*, cost* J*, and constraint matrix* D *before sending them during the handshaking step,*

2. *how to modify the measurements sent to the plant, and*

3. *how to modify the inputs received from the plant,*

so that the plant's trajectory minimizes cost J *in* (1.4)*, while preventing the cloud from learning the plant* (A, B, C)*, the cost* J*, the constraint matrix* D*, and the plant's trajectory* $\{x_k, u_k, y_k\}_{k\in\mathbb{N}}$*?*

Problem 1.2.2 (Scenario 2)**.** *Assuming the cloud has no knowledge about the plant except for knowing what are its sensors and actuators:*

1. *how to modify the plant* (A, B, C)*, cost* J*, and constraint matrix* D *before sending them during the handshaking step;*

2. *how to modify the measurements sent to the plant, and*

3. *how to modify the inputs received from the plant,*

so that the plant's trajectory minimizes cost J in (1.4)*, while preventing the cloud from learning the plant* (A, B, C)*, the cost J, the constraint matrix D, and the plant's trajectory* $\{x_k, u_k, y_k\}_{k\in\mathbb{N}}$*?*

Problem 1.2.3 (Scenario 3)**.** *Assuming the cloud has complete knowledge about the plant dynamics, including its sensors and actuators:*

1. *how to modify cost J, and constraint matrix D before sending them alongside the plant (A, B, C) during the handshaking step;*
2. *how to modify the measurements sent to the plant, and*
3. *how to modify the inputs received from the plant,*

so that the plant's trajectory minimizes cost J in (1.4)*, while preventing the cloud from learning the cost J, the constraint matrix D, and the plant's trajectory* $\{x_k, u_k, y_k\}_{k\in\mathbb{N}}$*?*

These problems are solved in Section 1.4 by utilizing isomorphisms and symmetries of control systems we define next in Section 1.3.

1.3 Isomorphisms and symmetries of control systems

In this section, we introduce the notions of isomorphism and symmetry of control systems along with several technical results used in Section 1.4 to provide a solution to the problems described in Section 1.2.

Let us denote by $\mathcal{S}_{n,m,p}$ the set of all controllable and observable linear control systems with state, input and output dimensions n, m, and p, respectively.

Definition 1.3.1. An isomorphism of control systems in $\mathcal{S}_{n,m,p}$ is a quadruple $\psi = (P, F, G, S)$ consisting of a change of state coordinates $P : \mathbb{R}^n \to \mathbb{R}^n$, state feedback $F : \mathbb{R}^n \to \mathbb{R}^m$, a change of coordinates in the input space $G : \mathbb{R}^m \to \mathbb{R}^m$, and a change of coordinates in the output space $S : \mathbb{R}^p \to \mathbb{R}^p$. Transformations P and S are affine invertible maps, F is an affine map and G is a linear invertible map.

Recall that, to simplify notation, we lift the affine maps to linear maps using the transformation (1.2).

Let us also denote the set of isomorphisms of $\mathcal{S}_{n,m,p}$ described in Definition 1.3.1 as $\mathcal{G}_{n,m,p}$. The set $\mathcal{G}_{n,m,p}$ forms a group under function composition as the group operation[1]. This allows us to define a group action of $\mathcal{G}_{n,m,p}$ on the set of linear control systems $\mathcal{S}_{n,m,p}$.

Definition 1.3.2. Each element $\psi \in \mathcal{G}_{n,m,p}$ acts on $\Sigma \in \mathcal{S}_{n,m,p}$ to produce $\psi_* \Sigma$ given by:

$$\begin{aligned} \psi_* \Sigma &= (P, F, G, S)_*(A, B, C) \\ &= (P(A - BG^{-1}F)P^{-1}, PBG^{-1}, SCP^{-1}) \\ &\triangleq (\tilde{A}, \tilde{B}, \tilde{C}) \triangleq \tilde{\Sigma}. \end{aligned} \tag{1.6}$$

The map ψ_* is called an isomorphism action. We also say that systems Σ and $\tilde{\Sigma}$ are equivalent.

An isomorphism maps the state x_k, input u_k, and output y_k of system Σ to the state $\tilde{x}_k$, input $\tilde{u}_k$, and output $\tilde{y}_k$ of system $\tilde{\Sigma}$ as follows:

$$\tilde{x}_k = Px_k \tag{1.7}$$

$$\tilde{u}_k = Fx_k + Gu_k \tag{1.8}$$

$$\tilde{y}_k = Sy_k. \tag{1.9}$$

Similarly, an isomorphism induces transformation on the control objectives — i.e., the cost and constraints. The effect of ψ on η_k can be represented by:

$$\tilde{\eta}_k = \begin{bmatrix} \tilde{x}_k \\ \tilde{u}_k \end{bmatrix} = \begin{bmatrix} P & 0 \\ F & G \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} \triangleq L\eta_k. \tag{1.10}$$

[1] A composition of two isomorphisms is given by $\psi_2 \circ \psi_1 = (P_2P_1, G_2F_1 + F_2P_1, G_2G_1, S_2S_1)$, the identity is $\psi_e = (I, 0, I, I)$ and the inverse is given by $\psi^{-1} = (P^{-1}, -G^{-1}FP, G^{-1}, S^{-1})$.

Therefore, the cost function J can be expressed as a function of the sequence of modified states $\tilde{x} = \{\tilde{x}_0, ..., \tilde{x}_N\}$ and the sequence of modified inputs $\tilde{u} = \{u_0, ..., \tilde{u}_N\}$ as follows:

$$\tilde{J}(\tilde{x}, \tilde{u}) = \psi_* J\,(x, u) = \sum_{i=0}^{N} \Delta\tilde{\eta}_i^T \tilde{M} \Delta\tilde{\eta}_i, \tag{1.11}$$

where $\tilde{M} = L^{-T} M L^{-1}$. Applying the isomorphism action to the constraints in (1.5) yields:

$$\tilde{D}\tilde{\eta}_i \leq 0, \quad \forall i \in \{0, 1, ..., N\}, \tag{1.12}$$

where $\tilde{D} = \psi_* D = DL^{-1}$.

The effect of an isomorphism on the system, trajectory, cost and constraints will be used in Section 1.4 to prevent the cloud from learning them.

For a given system Σ, there is a special subgroup of $\mathcal{G}_{n,m,p}$ called the symmetry group of Σ, which is defined by the following property.

Definition 1.3.3. Let $\Sigma \in \mathcal{S}_{n,m,p}$. An isomorphism $\psi \in \mathcal{G}_{n,m,p}$ is said to be a symmetry of Σ if $\psi_* \Sigma = \Sigma$. The subgroup of symmetries of Σ is denoted here as $\mathcal{K}_{n,m,p}(\Sigma)$.

The notion of isomorphism was crafted to preserve properties of control systems. Among these, trajectories have a special significance. A simple induction argument can be used to establish the following result.

Lemma 1.3.4. *Let $\Sigma \in \mathcal{S}_{n,m,p}$ and $\psi \in \mathcal{G}_{n,m,p}$. If $\tilde{\Sigma} = \psi_* \Sigma$ and $\{x_k, u_k, y_k\}_{k\in\mathbb{N}}$ is a trajectory of Σ, then $\{\tilde{x}_k, \tilde{u}_k, \tilde{y}_k\}_{k\in\mathbb{N}}$, as given by* (1.7) - (1.9)*, is a valid trajectory of $\tilde{\Sigma}$.*

This means that if the cloud receives $\tilde{\Sigma}$ during the handshaking step, then the received sequence of measurements $\tilde{y}$ and the produced sequence of control inputs $\tilde{u}$ in the subsequent execution step are compatible with the plant $\tilde{\Sigma}$. To elaborate, both the modified measurements $\tilde{y}$ and modified control inputs $\tilde{u}$ would be compatible with modified dynamics $\tilde{\Sigma}$.

Let us now define $\bar{\mathcal{S}}_{n,m,p}$ to be a set of quadruples $\Omega \triangleq \{\Sigma, J, D, \{x_k, y_k, u_k\}\}_{k\in\mathbb{N}}$ such that $\{x_k, y_k, u_k\}$ is a trajectory of a linear system $\Sigma \in \mathcal{S}_{n,m,p}$ minimizing cost function J under constraints D.

Lemma 1.3.5. *The set $\bar{\mathcal{S}}_{n,m,p}$ is a smooth manifold.*

Proof. We can see that $\bar{\mathcal{S}}_{n,m,p}$ is, in fact, the Cartesian product of $\mathcal{S}_{n,m,p}$ with the set of cost functions $\mathcal{M}^{++}(m+n+1, \mathbb{R})$, defined by positive-definite matrices, with the set of constraints $\mathcal{M}_d(h\times(m+n+1), \mathbb{R})$, defined by the set of full-rank matrices, where $d = \min\{h, m+n+1\}$. It is known that the product space is a smooth manifold if its constituents are smooth manifolds [42, p. 21]. It remains to show that these constinuents are indeed smooth manifolds.

Let us construct the map:

$$\begin{aligned} f_S :&\mathbb{R}^{n\times(n+1)} \times \mathbb{R}^{n\times m} \times \mathbb{R}^{p\times(n+1)} \to \mathbb{R}^2 \\ &(A, B, C) \mapsto (\det \mathcal{C}, \det \mathcal{O}), \end{aligned} \tag{1.13}$$

where $\mathcal{C}$ and $\mathcal{O}$ are the controllability and observability matrices of the dynamics (A, B, C). It can be seen that $\mathcal{S}_{n,m,p} = f_S^{-1}(\mathbb{R}^2 \setminus (0, 0))$. The function f_S is continuous since each of its elements is defined by a polynomial function of the elements of (A, B, C). Given that for continuous functions the preimage of every open set is an open set, we have that $\mathcal{S}_{n,m,p}$ is an open subset of the domain of f_S. Seeing that the domain of f_S is a smooth manifold, $\mathcal{S}_{n,m,p}$ is a smooth manifold of dimension $n(n + 1) + nm + p(n + 1)$.

The set of positive-definite matrices $\mathcal{M}^{++}(m+n+1, \mathbb{R})$ is shown to be a smooth embedded submanifold of $\mathbb{R}^{(m+n+1)\times(m+n+1)}$ of dimension $(m + n + 1)(m + n + 2)/2$ in [43].

The set of full-rank matrices $\mathcal{M}_d(h \times (m + n + 1), \mathbb{R})$ is a smooth manifold of dimension $h(m + n + 1)$ [42, p. 19]. □

Similarly to $\mathcal{S}_{n,m,p}$, we can define a group action of $\mathcal{G}_{n,m,p}$ on $\bar{\mathcal{S}}_{n,m,p}$ in view of the previous discussion.

Therefore, we can use the isomorphism action of $\mathcal{G}_{n,m,p}$ to define an equivalence relation on $\bar{\mathcal{S}}_{n,m,p}$.

Definition 1.3.6. Let $\Omega = (\Sigma, J, D, \{x_k, u_k, y_k\}_{k\in\mathbb{N}})$ and $\tilde{\Omega} = (\tilde{\Sigma}, \tilde{J}, \tilde{D}, \{\tilde{x}_k, \tilde{u}_k, \tilde{y}_k\}_{k\in\mathbb{N}})$ be elements of $\bar{\mathcal{S}}_{n,m,p}$. The equivalence relation $\sim_{\mathcal{G}}$ on $\bar{\mathcal{S}}_{n,m,p}$ denoted by:

$$\Omega \sim_{\mathcal{G}} \tilde{\Omega}, \tag{1.14}$$

is defined by the existence of $\psi \in \mathcal{G}_{n,m,p}$ such that:

$$\tilde{\Omega} = \psi_*\Omega; \tag{1.15}$$

i.e., $\tilde{\Sigma} = \psi_*\Sigma$, $\tilde{J} = \psi_*J$, $\tilde{D} = \psi_*D$, and $\{\tilde{x}_k, \tilde{u}_k, \tilde{y}_k\}_{k\in\mathbb{N}}$ is given in terms of $\{x_k, u_k, y_k\}_{k\in\mathbb{N}}$ as in (1.7) - (1.9).

The equivalence relation $\sim_{\mathcal{G}}$, in turn, defines equivalence classes in $\bar{\mathcal{S}}_{n,m,p}$. The equivalence class of $\Omega \in \bar{\mathcal{S}}_{n,m,p}$ defined by the action of $\mathcal{G}_{n,m,p}$ is the set:

$$\begin{aligned}[\Omega] &\triangleq \{\Omega' \in \bar{\mathcal{S}}_{n,m,p} | \exists\psi \in \mathcal{G}_{n,m,p} \text{ such that } \Omega' = \psi_*\Omega\} \\ &= \{\psi_*\Omega | \psi \in \mathcal{G}_{n,m,p}\}. \end{aligned} \tag{1.16}$$

This equivalence class is also called the orbit of Ω under action of $\mathcal{G}_{n,m,p}$.

To facilitate further results, let us show that $\mathcal{G}_{n,m,p}$ is a Lie group acting on $\bar{\mathcal{S}}_{n,m,p}$.

Lemma 1.3.7. *The group $\mathcal{G}_{n,m,p}$ is a Lie group of dimension $n(n+1)+m(n+1)+m^2+p(p+1)$ acting smoothly on $\bar{\mathcal{S}}_{n,m,p}$.*

Proof. It was previously established that $\mathcal{G}_{n,m,p}$ is a group. It is a Lie group because it is a Cartesian product of smooth manifolds (i.e., general linear groups and vector spaces of various dimensions) and its multiplication and inversion maps are smooth. Moreover, since the dimension of a product of smooth manifolds is equal to the sum of the factors' dimensions, the dimension of $\mathcal{G}_{n,m,p}$ is $n(n + 1) + m(n + 1) + m^2 + p(p + 1)$ [42, p. 21].

The group $\mathcal{G}_{n,m,p}$ acts smoothly on $\bar{\mathcal{S}}_{n,m,p}$ since its action involves matrix multiplication and matrix inversion: the former results in every element of the product being a polynomial function of the elements of the factors, while the latter is smooth by Cramer's rule [42]. □

The next result shows that when the cloud optimizes $\tilde{J}$ and the plant replaces each y_k with output $\tilde{y}_k$, the resulting sequence of inputs $\tilde{u}$ can be used to reconstruct a sequence of inputs u that optimizes J. Its proof amounts to using the change of variables (1.7)-(1.9).

Lemma 1.3.8. *Let $\Omega \in \bar{\mathcal{S}}_{n,m,p}$ and $\psi \in \mathcal{G}_{n,m,p}$. Suppose the cloud solves the optimization problem:*

$$\begin{aligned} &\underset{\tilde{u}}{min} && \tilde{J}(\tilde{x}, \tilde{u}) \\ &subject\ to && \hat{D}\hat{\eta}_i \leq 0, \quad \forall i \in \{0, ..., N\}, \end{aligned}$$

for the plant $\tilde{\Sigma} = \psi_* \Sigma$ and the sequence $\tilde{u}^*$ is a unique solution of this optimization problem. Then, the unique solution of the optimization problem:

$$\begin{aligned} &\underset{u}{min} && J(x, u) \\ &subject\ to && D\eta_i \leq 0, \quad \forall i \in \{0, ..., N\} \end{aligned}$$

for the plant Σ is the sequence u^* such that $u_i^* = G^{-1}(\tilde{u}_i^* - Fx_i)$ for all $i \in \{0, ..., N\}$.

1.4 Solving the control-over-the-cloud privacy problem

1.4.1 Enforcing privacy

The main reason for using isomorphisms is to preclude the cloud from distinguishing between isomorphic systems. We now formalize the notion of indistinguishability.

Definition 1.4.1. A protocol renders two quadruples Ω and $\tilde{\Omega}$ indistinguishable by the cloud if the exchanged messages, when using the protocol between the cloud and the plant Ω, and the exchanged messages, when using the protocol between the cloud and the plant $\tilde{\Omega}$, can be made the same.

The results from Section 1.3 allow us to construct a communication protocol between the plant and the cloud that, as will be further shown, solves Problems 1.2.1-1.2.3. We start by detailing this protocol.

Algorithm 1 Secure communication

```
Input: Plant: $\psi$, $\Sigma$, $J$, $D$, $\tilde{u}_k$;
    Cloud: $\tilde{y}_k$, $\tilde{\Sigma}$, $\tilde{J}$, $\tilde{D}$
Output: Plant: $\tilde{\Sigma}$, $\tilde{J}$, $\tilde{D}$, $\tilde{y}_k$;
    Cloud: $\tilde{u}_k$
  Phase 1: Handshaking:
  Plant: Encode $\Sigma$, $J$, $D$ into $\tilde{\Sigma} = \psi_* \Sigma$, $\tilde{J} = \psi_* J$ and $\tilde{D} = \psi_* D$;
  Plant: Send $\tilde{\Sigma}$, $\tilde{J}$, and $\tilde{D}$ to the cloud;
  Phase 2: Execution:
  Plant: Encode measurement $y_k$ into $\tilde{y}_k = S y_k$ and send $\tilde{y}_k$ to the cloud;
  Cloud: Use the received $\tilde{y}_k$ to estimate $\tilde{x}_k$ and compute $\tilde{u}_k$ minimizing $\tilde{J}$ subject to the constraints $\tilde{D}$ and the dynamics $\tilde{\Sigma}$;
  Cloud: Send $\tilde{u}_k$ to the plant;
  Plant: Use the isomorphism $\psi$ to decode $\tilde{u}_k$ and produce $u_k$ using (1.8);
  Plant: Apply $u_k$ to the actuators.
```

From Lemma 1.3.8, we see that Algorithm 1 provides the plant with the inputs u_k that satisfy the original control objective — i.e., the plant's trajectory minimizes cost J under affine constraints D.

Let us note how all the required computations in this algorithm are matrix multiplications, which means that both handshaking and execution can be performed in $O(k^3)$ time, where $k = \max\{n, m, p\}$. However, performing matrix multiplications of constant matrices (e.g., $G^{-1}F$) in advance would reduce the complexity of the execution to $O(k^2)$. Both of these complexities were calculated only for the client side (i.e., Plant) of the algorithm.

Let us now show that applying this protocol indeed makes any two systems in the same equivalence class indistinguishable from each other.

Theorem 1.4.2. *Algorithm 1 renders isomorphic systems* $\Omega = (\Sigma, J, D, \{x_k, u_k, y_k\}_{k\in\mathbb{N}})$ *and* $\tilde{\Omega} = (\tilde{\Sigma}, \tilde{J}, \tilde{D}, \{\tilde{x}_k, \tilde{u}_k, \tilde{y}_k\}_{k\in\mathbb{N}})$ *indistinguishable by the cloud.*

Proof. Since Ω and $\tilde{\Omega}$ are isomorphic, there exists an isomorphism ψ such that $\psi_*\Sigma = \tilde{\Sigma}$, $\psi_*J = \tilde{J}$, and $\psi_*D = \tilde{D}$. Indistinguishability of Ω and $\tilde{\Omega}$ will be shown by running two instances of Algorithm 1: one with Ω and ψ as inputs, the other - with $\tilde{\Omega}$ and the identity isomorphism ψ_e. Let us denote the communication algorithm described in Algorithm 1 applied to $\Omega \in \bar{\mathcal{S}}_{n,m,p}$ with the selected isomorphism $\psi \in \mathcal{G}_{n,m,p}$ by $\mathrm{Alg}(\Omega, \psi)$. During handshaking:

- when $\mathrm{Alg}(\Omega, \psi)$ is executed, the plant sends $\psi_*\Sigma$, ψ_*J, and ψ_*D;
- when $\mathrm{Alg}(\tilde{\Omega}, \psi_e)$ is executed (ψ_e is the identity of $\mathcal{G}_{n,m,p}$), the plant sends $\tilde{\Sigma}$, $\tilde{J}$, and matrix $\tilde{D}$ unprotected.

Thus, the communicated dynamics and optimization problems are the same. During execution:

- when $\mathrm{Alg}(\Omega, \psi)$ is executed, ψ takes trajectories $\{x_k, u_k, y_k\}_{k\in\mathbb{N}}$ of Σ to trajectories $\{\tilde{x}_k, \tilde{u}_k, \tilde{y}_k\}_{k\in\mathbb{N}}$ of $\psi_*\Sigma$;
- when $\mathrm{Alg}(\tilde{\Omega}, \psi_e)$ is executed, the trajectories are $\{\tilde{x}_k, \tilde{u}_k, \tilde{y}_k\}_{k\in\mathbb{N}}$.

Therefore, the cloud receives the same measurements from both plants. In response, since both plants communicated the same optimization problem, the cloud sends the same control inputs to both plant Ω and $\tilde{\Omega}$. □

The result described in Theorem 1.4.2 states that the cloud cannot differentiate between any two plants, costs, constraints or trajectories contained in the same equivalence class of the $\sim_{\mathcal{G}}$-equivalence relation, thereby protecting the privacy of the system. In the next section, we quantify the amount of privacy provided by Algorithm 1.

1.4.2 Quantifying privacy

Privacy is created by preventing the cloud from knowing which quadruple Ω in its equivalence class $[\Omega]$ it is interacting with. Clearly, the larger the equivalence class, the more privacy is ensured. Since each equivalence class has infinitely many elements, cardinality cannot be used as a measure of privacy. In this section, we show that each equivalence class is a smooth manifold and we quantify privacy using the dimension of this manifold.

1.4.2.1 Preliminaries: stabilizer subgroups and their dimensions

The stabilizer subgroup of $\mathcal{G}_{n,m,p}$ for any $\Omega \in \bar{S}_{n,m,p}$, denoted by $\mathcal{K}_{n,m,p}(\Omega)$, is defined by:

$$\mathcal{K}_{n,m,p}(\Omega) = \{\psi \in \mathcal{G}_{n,m,p} | \psi_* \Omega = \Omega\}. \tag{1.17}$$

The subgroup $\mathcal{K}_{n,m,p}(\Omega)$ must be a subset of the symmetry subgroup $\mathcal{K}_{n,m,p}(\Sigma)$ since it must preserve the dynamics.

In [44], Respondek gives a characterization of the symmetries of controllable pairs (A, B). Since when considering pairs (A, B) the output is not relevant, the isomorphisms of (A, B) degenerate into the form $\phi = (P, F, G)$, where the matrices P, F and G are defined to be the same as their counterparts in Definition 1.3.1. We denote the group of these isomorphisms by $\mathcal{G}_{n,m}$. The group action of $\mathcal{G}_{n,m}$ is given by:

$$\phi_*(A, B) = (P(A - BG^{-1}F)P^{-1}, PBG^{-1}). \tag{1.18}$$

Let us define the symmetry subgroup of controllable systems (A, B) as:

$$\mathcal{K}_{n,m}(A, B) = \{\phi \in \mathcal{G}_{n,m} | \phi_*(A, B) = (A, B)\}. \tag{1.19}$$

The next proposition uses the results from [45] and the notion of controllability indices (see [46] for a definition) to estimate the dimension of $\mathcal{K}_{n,m}(A,B)$:

Proposition 1.4.3. *Let (A,B) be a controllable pair. Then:*

$$m(n+1) - s \leq \dim\, \mathcal{K}_{n,m}(A,B) \leq n(m+1) - s,$$

where:

$$\begin{aligned} s &= \sum_{i=2}^{m} r_{i-1} r_i, \\ r_1 &= \text{rank } B, \\ r_i &= \text{rank } S_{i-1}(A,B) - \text{rank } S_{i-2}(A,B), \;\; i = 2, ..., m, \\ S_j(A,B) &= \begin{bmatrix} B & AB & ... & A^j B \end{bmatrix}, \;\; j = 1, ..., m-1. \end{aligned}$$

and $\{\kappa_i\}_{i=1}^m$ *are controllability indices of* (A,B).

Proof. The symmetry subgroup $\mathcal{K}_{n,m}(A,B)$ consists of solutions to the following system of equations:

$$\begin{cases} A = P(A - BG^{-1}F)P^{-1} \\ B = PBG^{-1}, \end{cases} \tag{1.20}$$

which is equivalent to:

$$\begin{cases} AP + BF = PA \\ BG = PB. \end{cases} \tag{1.21}$$

Recall that elements of the pair (A,B) and transformations (P,F,G) are, in fact, affine maps. If we express (1.21) using the inner structure of the maps, we get:

$$\begin{cases} \bar{A}\bar{P} + \bar{B}\bar{F} = \bar{P}\bar{A} \\ \bar{B}G = \bar{P}\bar{B} \\ (\bar{A} - I)\bar{p} + \bar{B}\bar{f} = \bar{P}\bar{c} - \bar{c}, \end{cases} \tag{1.22}$$

where $P = \begin{bmatrix} \bar{P} & \bar{p} \\ 0 & 1 \end{bmatrix}$ and $F = \begin{bmatrix} \bar{F} & \bar{f} \end{bmatrix}$. Finding elements of $\mathcal{K}_{n,m}(A, B)$ is equivalent to finding $(\bar{P}, \bar{p}, \bar{F}, \bar{f}, G)$. According to Theorem 2.2 in [45], the dimension of solution space S of $(\bar{P}, \bar{F}, G)$ satisfying the first and second equations in (1.22) is equal to:

$$\begin{aligned} \dim S &= m(n+m) - \sum_{i=1}^{m} r_{i-1} r_i \\ &= m(n+m) - r_0 r_1 - \sum_{i=2}^{m} r_{i-1} r_i \\ &= mn - \sum_{i=2}^{m} r_{i-1} r_i, \end{aligned} \tag{1.23}$$

because $r_0 = r_1 = m$, $\kappa_1 = m$ and (A, B) is a controllable pair. Fixing $(\bar{P}, \bar{F}, G)$, one can find the dimension of the solution space of the third equation in (1.22). It can be observed that the dimension of the solution space is equal to dim ker $\begin{bmatrix} \bar{A} - I & \bar{B} \end{bmatrix}$. Since rank $\bar{B} = m$, it follows that:

$$m \leq \text{dim ker} \begin{bmatrix} \bar{A} - I & \bar{B} \end{bmatrix} \leq n. \tag{1.24}$$

The result then follows from (1.23) and (1.24). □

This result can be used to estimate the dimension of $\mathcal{K}_{n,m,p}(\Sigma)$. If $\Sigma = (A, B, C)$, then, from Proposition 1.4.3, we know the dimension of $\mathcal{K}_{n,m}(A, B)$ and that any $\phi \in \mathcal{K}_{n,m}(A, B)$ satisfies $\phi_*(A, B) = (A, B)$. Given $\phi = (P, F, G) \in \mathcal{K}_{n,m}(A, B)$, finding a corresponding $\psi = (P, F, G, S) \in \mathcal{K}_{n,m,p}(\Sigma)$ requires finding S such that $C = SCP^{-1}$. Since we assume C has linearly independent rows, for a given P, this equation has at most one solution. A solution exists if and only if Im $C^T \subset$ Im $P^{-T} C^T$ [47]. Let $\mathcal{Q}(A, B, C)$ be the subset of $\mathcal{K}_{n,m}(A, B)$ defined by the elements (P, F, G) for which a unique solution to $C = SCP^{-1}$ exists. It can be seen that there is a one-to-one correspondence between $\mathcal{Q}(A, B, C)$ and $\mathcal{K}_{n,m,p}(\Sigma)$. Since $\mathcal{Q}(A, B, C) \subset \mathcal{K}_{n,m}(A, B)$, this gives an upper bound on the dimension of the symmetry subgroup:

$$\dim \mathcal{K}_{n,m,p}(\Sigma) \leq \dim \mathcal{K}_{n,m}(A, B). \tag{1.25}$$

Lemma 1.4.4. *For any* $\Omega = (\Sigma, J, D, \{x_k, u_k, y_k\}_{k\in\mathbb{N}}) \in \bar{\mathcal{S}}_{n,m,p}$,

$$dim\ \mathcal{K}_{n,m,p}(\Omega) \leq dim\ \mathcal{K}_{n,m,p}(\Sigma) \leq dim\ \mathcal{K}_{n,m}(A,B),$$

where $dim\ \mathcal{K}_{n,m,p}(A,B)$ *is given by Proposition 1.4.3.*

Let us consider a special case, in which the dimension of $\mathcal{K}_{n,m,p}(\Sigma)$ can be computed exactly.

Definition 1.4.5. A system $\Sigma \in \mathcal{S}_{n,m,p}$ is said to be a prime system if it is $\sim_{\mathcal{G}}$-equivalent to the system of the form:

$$\Sigma : \quad \begin{cases} x_{k+1}^{(i,1)} = x_k^{(i,2)}, \\ \vdots \\ x_{k+1}^{(i,\kappa_i)} = u_k^{(i)}, \\ y_k^{(i)} = x_k^{(i,1)}, \ \ 1 \leq i \leq m, \end{cases} \tag{1.26}$$

where $x_k = \left[x_k^{(1,1)}, ..., x_k^{(1,\kappa_1)}, ..., x_k^{(m,1)}, ..., x_k^{(m,\kappa_m)}\right]^T \in \mathbb{R}^n$ and $\{\kappa_i\}_{i=1}^m$ are controllability indices of (A,B).

For prime systems we have the following characterization of the dimension of $\mathcal{K}_{n,m,p}(\Sigma)$.

Lemma 1.4.6. *Let* $\Sigma \in \mathcal{S}_{n,m,p}$ *be a prime system. Then,*

$$\sum_{i=1}^{m} r_{\kappa_i} + m \leq dim\ \mathcal{K}_{n,m,p}(\Sigma) \leq \sum_{i=1}^{m} r_{\kappa_i} + n, \tag{1.27}$$

where

$$r_1 = rank\ B,$$

$$r_i = rank\ S_{i-1}(A,B) - rank\ S_{i-2}(A,B), \ \ i = 2, ..., m,$$

$$S_j(A,B) = \begin{bmatrix} B & AB & ... & A^jB \end{bmatrix}, \ \ j = 1, ..., m-1,$$

and $\{\kappa_i\}_{i=1}^m$ *are controllability indices of* (A,B).

Proof. Without loss of generality, let us consider a prime system of the form (1.26). From Proposition 2 in [44], we can see that if a system is prime, a symmetry $\psi = (P, F, G, S)$ is uniquely defined by a transformation on its outputs (i.e., by transformation S).

We want to show that, in order to define a symmetry, transformation S needs to be constructed in such a way that each transformed output $\tilde{y}_k^{(i)}$ is an affine function of outputs $y_k^{(j)}$ with relative degrees greater or equal than that of $y_k^{(i)}$. To simplify notation, we prove this claim for the example with controllability indices $\kappa_1 = \kappa_2 = 2$, $\kappa_3 = 1$, although the employed arguments apply to any prime system:

$$\begin{aligned}
x_{k+1}^{(1,1)} &= x_k^{(1,2)} \qquad & x_{k+1}^{(2,1)} &= x_k^{(2,2)} \qquad & x_{k+1}^{(3,1)} &= u_k^{(3)} \\
x_{k+1}^{(1,2)} &= u_k^{(1)} \qquad & x_{k+1}^{(2,2)} &= u_k^{(2)} \\
y_k^{(1)} &= x_k^{(1,1)} \qquad & y_k^{(2)} &= x_k^{(2,1)} \qquad & y_k^{(3)} &= x_k^{(3,1)}.
\end{aligned} \tag{1.28}$$

We will show, by contradiction, that if S produces a transformed output based on outputs of a smaller relative degree, then S cannot be part of a symmetry. In other words, there exist no matrices P, F, and G such that the quadruple (P, F, G, S) satisfies the equations:

$$A = P(A - BG^{-1}F)P^{-1} \tag{1.29}$$

$$B = PBG^{-1} \tag{1.30}$$

$$C = SCP^{-1}. \tag{1.31}$$

Assume that (1.29)-(1.31) are satisfied and that S contains non-zero elements S_{ij} if $\kappa_i > \kappa_j$ (i.e., the transformed output uses outputs of a smaller relative degree). From (1.31), we have that:

$$SCA^qB = CPA^qB, \quad \forall\ 0 \le q < \kappa_1. \tag{1.32}$$

By using (1.29) and (1.30), the following relation can be shown:

$$PA = AP + PBG^{-1}F = AP + BF. \tag{1.33}$$

Recursively substituting (1.33) into (1.32) results in:

$$\begin{aligned}SCA^qB &= C(PA)A^{q-1}B = C(AP+BF)A^{q-1}B\\ &= CBFA^{q-1}B + CAPA^{q-1}B\\ &= CBFA^{q-1}B + CA(PA)A^{q-2}B\\ &= \dots\\ &= \sum_{l=0}^{q-1} CA^lBFA^{q-l-1}B + CA^qPB.\end{aligned}$$

Equation (1.30) implies that $PB = BG$ and, thus, leads to:

$$SCA^qB = \sum_{l=0}^{q-1} CA^lBFA^{q-l-1}B + CA^qBG. \tag{1.34}$$

Note that CA^lB is a diagonal matrix such that:

$$[CA^lB]_{ii} = \begin{cases} 1, & \text{if } \kappa_i = l+1 \\ 0, & \text{otherwise.} \end{cases} \tag{1.35}$$

In other words, this diagonal matrix marks the indices corresponding to the outputs of equal relative degree. In addition, the expression $FA^{q-l-1}B$ is an $m \times m$ matrix composed out of elements of F (recall that A and B are in the form (1.26)).

The left-hand side of (1.34) selects the columns of S corresponding to the outputs of relative degree $\kappa_i = q+1$. For the example in (1.28), taking $q = 0$ gives:

$$SCB = \begin{bmatrix} 0 & 0 & S_{13} \\ 0 & 0 & S_{23} \\ 0 & 0 & S_{33} \end{bmatrix}. \tag{1.36}$$

The right-hand side of (1.34) fills the rows corresponding to the outputs of relative degree smaller or equal than $\kappa_i = q+1$ with values from G. In case of example in (1.28), the right-hand side, given $q = 0$, is:

$$CBG = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \times & \times & \times \end{bmatrix}. \tag{1.37}$$

Thus, the equality in (1.34), which was derived using the definition of symmetry, forces S_{ij} to zero if $\kappa_i > \kappa_j$. In the example in (1.28), this leads to $S_{13} = S_{23} = 0$. This contradicts the assumption that S produces a transformed output based on outputs of a smaller relative degree.

This idea can be generalized to any prime system and, therefore, each transformed output $\tilde{y}_k^{(i)}$ can only be an affine function of outputs $y_k^{(j)}$ with relative degrees greater or equal than that of $y_k^{(i)}$.

The number of outputs $y_k^{(j)}$ with a relative degree greater or equal to that of $y_k^{(i)}$ (i.e., greater or equal than k_i) is equal to r_{k_i} [45]. Therefore, each modified output $y_k^{(i)}$ is an affine function with r_{k_i} arguments. The constant terms of transformations P, F, and S, denoted by $\bar{p}$, $\bar{f}$, and $\bar{s}$, respectively, need to satisfy the following equalities:

$$\begin{cases} (\bar{A} - I)\bar{p} + \bar{B}\bar{f} = \bar{P}\bar{c} - \bar{c} \\ \bar{s} = \bar{C}\bar{p} + \bar{d} - \bar{S}\bar{d}, \end{cases}$$

where $P = \begin{bmatrix} \bar{P} & \bar{p} \\ 0 & 1 \end{bmatrix}$, $F = \begin{bmatrix} \bar{F} & \bar{f} \end{bmatrix}$, and $S = \begin{bmatrix} \bar{S} & \bar{s} \\ 0 & 1 \end{bmatrix}$. Similarly to the proof of Proposition 1.4.3, the dimension of the solution space of this system is given by the dimension of the kernel of the linear map defining the left-hand side of the system of equations as:

$$m \leq \dim \ker \begin{bmatrix} \bar{A} - I & \bar{B} & 0 \\ 0 & 0 & I \end{bmatrix} \leq n, \tag{1.38}$$

thereby leading to the result of this lemma.

□

1.4.2.2 Main results

Consider the scenario from Problem 1.2.1, in which the cloud does not know anything about the system. In this scenario, the plant encodes Ω using an isomorphism $\psi = (P, F, G, S)$ that can be regarded as a private key used to encode and decode the information exchanged with the cloud. This isomorphism ψ is chosen from $\mathcal{G}_{n,m,p}$, the group of all isomorphisms.

Proposition 1.4.7. *Let $\Omega \in \bar{\mathcal{S}}_{n,m,p}$. Then, under the scenario described in Problem 1.2.1, the cloud cannot distinguish between Ω and any other system in the uncertainty set $[\Omega]_{\mathcal{G}}$ (i.e., the equivalence class of Ω defined by the action of $\mathcal{G}_{n,m,p}$) of dimension:*

$$dim\ \mathcal{G}_{n,m,p} - dim\ \mathcal{K}_{n,m,p}(\Omega), \tag{1.39}$$

if Algorithm 1 is used.

This implies that the dimension of $[\Omega]_{\mathcal{G}}$ is greater or equal than:

$$n^2 + m(m+1) + p(p+1) + \sum_{i=2}^{m} r_{i-1}r_i, \tag{1.40}$$

where r_i is given in Lemma 1.4.3.

For $\Omega \in \bar{\mathcal{S}}_{n,m,p}$ such that its corresponding $\Sigma \in \mathcal{S}_{n,m,p}$ is prime, this implies that the dimension of $[\Omega]_{\mathcal{G}}$ is greater or equal to:

$$n^2 + m(n+1) + m^2 + p(p+1) - \sum_{i=1}^{m} r_{\kappa_i}, \tag{1.41}$$

where r_{κ_i} is given in Lemma 1.4.6.

Proof. From Theorem 1.4.2, we know that Algorithm 1 renders isomorphic systems indistinguishable by the cloud. Therefore, the uncertainty set is the set of systems isomorphic to $[\Omega]_{\mathcal{G}}$ - namely, the equivalence class of Ω defined by the action of $\mathcal{G}_{n,m,p}$.

Let us define a map:

$$\theta_\Omega : \mathcal{G}_{n,m,p} \to \bar{\mathcal{S}}_{n,m,p}$$

$$\psi \mapsto \psi_* \Omega.$$

Here, θ_Ω is smooth because, as shown in Lemma 1.3.7, $\mathcal{G}_{n,m,p}$ acts smoothly on $\bar{\mathcal{S}}_{n,m,p}$. The stabilizer set can be defined by:

$$\mathcal{K}_{n,m,p}(\Omega) = (\theta_\Omega)^{-1}(\Omega) = \{\psi | \psi_* \Omega = \Omega\}.$$

Since θ_Ω and its inverse are smooth and, therefore, continuous, the subgroup $\mathcal{K}_{n,m,p}(\Omega)$ is closed.

By Theorem 21.17 in [42], the quotient space $\mathcal{G}_{n,m,p}/\mathcal{K}_{n,m,p}(\Omega)$ is a smooth manifold of dimension dim $\mathcal{G}_{n,m,p}-$dim $\mathcal{K}_{n,m,p}(\Omega)$ such that the quotient map $\pi : \mathcal{G}_{n,m,p} \to \mathcal{G}_{n,m,p}/\mathcal{K}_{n,m,p}(\Omega)$ is a smooth submersion.

Now, let us define a map:

$$\begin{aligned} \Theta_\Omega : \mathcal{G}_{n,m,p}/\mathcal{K}_{n,m,p}(\Omega) &\to \bar{\mathcal{S}}_{n,m,p} \\ \psi\mathcal{K}_{n,m,p}(\Omega) &\mapsto \psi_*\Omega, \end{aligned}$$

where $\psi\mathcal{K}_{n,m,p}(\Omega)$ is a left coset of $\mathcal{K}_{n,m,p}(\Omega)$. It can be shown that Θ_Ω is well-defined.

By Theorem 4.29 in [42], Θ_Ω is smooth because $\theta_\Omega = \Theta_\Omega \circ \pi$ is smooth and π is a smooth submersion.

It can be shown that the map Θ_Ω is equivariant (see [42, p. 164]) and, therefore, by the equivariant rank theorem [42, p. 165], we have that Θ_Ω has a constant rank.

Let us show that Θ_Ω is injective. If $\Theta_\Omega(\psi_1\mathcal{K}_{n,m,p}(\Omega)) = \Theta_\Omega(\psi_2\mathcal{K}_{n,m,p}(\Omega))$, then $(\psi_1)_*\Omega = (\psi_2)_*\Omega$. This implies that $(\psi_1)^{-1}\psi_2 \in \mathcal{K}_{n,m,p}(\Omega)$ and, therefore, $\psi_1\mathcal{K}_{n,m,p}(\Omega) = \psi_2\mathcal{K}_{n,m,p}(\Omega)$. Therefore, Θ_Ω is a smooth immersion.

By Proposition 5.18 in [42], the image of Θ_Ω (i.e., the equivalence class $[\Omega]_\mathcal{G}$) is an immersed submanifold such that $\Theta_\Omega : \mathcal{G}_{n,m,p}/\mathcal{K}_{n,m,p}(\Omega) \to [\Omega]_\mathcal{G}$ is a diffeomorphism and, therefore, the dimension of $[\Omega]_\mathcal{G}$ is equal to the dimension of $\mathcal{G}_{n,m,p}/\mathcal{K}_{n,m,p}(\Omega)$.

A more concrete quantification of privacy can be given for various special cases. Using the results of Proposition 1.4.3 and Lemma 1.4.4, we have that, for any $\Omega \in \bar{\mathcal{S}}_{n,m,p}$, the uncertainty sets under the scenario described in Problem 1.2.1 are smooth manifolds of dimension greater or equal to the value in (1.40)

The dimension of the uncertainty sets for prime systems can be shown to be greater or equal to the value in (1.41) using Lemma 1.4.6. □

We can determine the knowledge the cloud can extract about the plant by considering what properties remain invariant under isomorphisms. Since controllability, observability, and the relative degree remain invariant, the cloud will not learn anything else beyond knowing that the plant is controllable, observable, and has a certain relative degree.

Example 1.4.8. To illustrate how different the systems produced by the proposed encoding scheme can be, consider a system with the following dynamics:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

We arbitrarily choose two sets of isomorphisms $\psi_1, \psi_2 \in \mathcal{G}_{n,m,p}$ such that the elements of their constituent matrices are between 0 and 1 (i.e., we pick isomorphisms from a bounded set of $\mathcal{G}_{n,m,p}$). We will not be explicitly writing these isomorphisms here due to space limitations. Applying these isomorphisms to the system above, we arrive at completely different systems $\tilde{\Sigma}_1 = \psi_{1*}\Sigma$ and $\tilde{\Sigma}_2 = \psi_{2*}\Sigma$:

$$\tilde{A}_1 = \begin{bmatrix} 35 & 9.4 & -40 \\ -3 & 0.1 & 2.9 \\ 28 & 8.3 & -33 \end{bmatrix}, \qquad \tilde{A}_2 = \begin{bmatrix} -4.06 & 5.35 & 1.48 \\ 4.87 & -4.0 & -2.33 \\ 0.68 & 2.40 & -0.88 \end{bmatrix},$$

$$\tilde{B}_1 = \begin{bmatrix} 16 & -7.7 \\ -2.2 & 1.5 \\ 13 & -6.1 \end{bmatrix}, \qquad \tilde{B}_2 = \begin{bmatrix} 0.16 & 0.95 \\ 1.03 & -1.27 \\ 0.70 & -0.31 \end{bmatrix},$$

$$\tilde{C}_1 = \begin{bmatrix} 2.4 & 0.02 & -1.8 \\ 1.5 & 0.01 & -1.1 \end{bmatrix}, \qquad \tilde{C}_2 = \begin{bmatrix} -0.33 & -1.68 & 1.56 \\ 3.39 & -4.58 & -0.97 \end{bmatrix}.$$

Proposition 1.4.7 can be used to quantify privacy of other scenarios presented in Section 1.2.

Consider the scenario in Problem 1.2.2, where the cloud does not know the dynamics but knows which sensors and actuators will be used. An arbitrary isomorphism can no longer be used for encoding since it could lead to inputs and outputs that are inconsistent with existing sensors and actuators. This inconsistency would signal the cloud that the plant is being dishonest about its measurements and provide the cloud with an opportunity to exploit this fact to gather additional knowledge. Therefore, we need to restrict the group of isomorphisms used for encoding. These isomorphisms are given by any composition of $\psi_1 = (P, 0, I, I)$ for any $P \in GL(n, \mathbb{R})$ and $\psi_2 \in \mathcal{K}_{n,m,p}(\Sigma)$. It can be shown that this set of isomorphisms forms a subgroup that we denote by $\mathcal{H}_{n,m,p}(\Sigma) \subset \mathcal{G}_{n,m,p}$.

Corollary 1.4.9. *Let $\Omega \in \bar{\mathcal{S}}_{n,m,p}$. Then, under the scenario described in Problem 1.2.2, the cloud cannot distinguish between Ω and any other system in the uncertainty set $[\Omega]_{\mathcal{H}}$ (i.e., the equivalence class of Ω defined by the action of $\mathcal{H}_{n,m,p}$) of dimension:*

$$dim\ \mathcal{H}_{n,m,p}(\Sigma) - dim\ \mathcal{K}_{n,m,p}(\Omega), \tag{1.42}$$

if Algorithm 1 is used. This implies that the dimension of $[\Omega]_{\mathcal{H}}$ is greater or equal to $n(n+1)$.

Proof. From Theorem 1.4.2, we know that Algorithm 1 renders isomorphic systems indistinguishable by the cloud. However, the uncertainty set is no longer the equivalence class under the entire group of isomorphisms $\mathcal{G}_{n,m,p}$, but the equivalence class under a smaller group $\mathcal{H}_{n,m,p}(\Sigma)$ denoted by $[\Omega]_{\mathcal{H}}$.

It can be shown that $\mathcal{H}_{n,m,p}(\Sigma)$ is a Lie subgroup of $\mathcal{G}_{n,m,p}$. This subgroup $\mathcal{H}_{n,m,p}(\Sigma)$ can be thought of as a product manifold of $\mathcal{K}_{n,m,p}(\Sigma)$ and a space of invertible affine maps. Since the dimension of a product manifold is a sum of its factors' dimensions, we have:

$$\dim\ \mathcal{H}_{n,m,p}(\Sigma) = \dim\ \mathcal{K}_{n,m,p}(\Sigma) + n(n+1).$$

The result follows by applying Proposition 1.4.7 to $\mathcal{H}_{n,m,p}(\Sigma)$. Using the result from Lemma 1.4.4, we can see that the dimension of the uncertainty set for any $\Omega \in \bar{\mathcal{S}}_{n,m,p}$ is greater or equal to $n(n+1)$. □

Since in this scenario the plant can no longer change the input, the cloud will learn the transfer function, but not the particular realization of the plant. The cloud would still be unable to learn the trajectory of the state.

Finally, in the scenario described in Problem 1.2.3, where the cloud possesses the complete knowledge of dynamics, only the isomorphisms from the symmetry subgroup $\psi \in \mathcal{K}_{n,m,p}(\Sigma)$ can be used. To provide privacy guarantees for this scenario, let us assume that we have $n+1$ linearly independent constraints on the state x_k expressed by the constraint matrix D. This is a reasonable assumption because systems often have an operational envelope bounding the states. Therefore, any $\psi \in \mathcal{K}_{n,m,p}(\Omega)$ must satisfy:

$$\begin{aligned} DL^{-1} = D \iff DL &= D \\ \iff \begin{bmatrix} D_{11} & 0 \\ D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} P & 0 \\ F & G \end{bmatrix} &= \begin{bmatrix} D_{11} & 0 \\ D_{21} & D_{22} \end{bmatrix} \\ \implies D_{11}P = D_{11}. \end{aligned}$$

Given that $D_{11} \in \mathbb{R}^{h_1 \times (n+1)}$ is injective, the last equality is satisfied if and only if $P = I$. Since P uniquely defines F, G and S, we also have that the only isomorphism that keeps (A, B, C, D_{11}) invariant is $\psi = \psi_e = (I, 0, I, I)$. Therefore, the only element of $\mathcal{K}_{n,m,p}(\Omega)$ is $\phi_e = (I, 0, I, I)$ and dim $\mathcal{K}_{n,m,p}(\Omega) = 0$.

Corollary 1.4.10. *Let $\Omega \in \bar{\mathcal{S}}_{n,m,p}$. Then, under the scenario described in Problem 1.2.3, the cloud cannot distinguish between Ω and any other system in the uncertainty set $[\Omega]_{\mathcal{K}}$ (i.e., the equivalence class of Ω defined by the action of $\mathcal{K}_{n,m,p}(\Sigma)$) of dimension:*

$$dim\ \mathcal{K}_{n,m,p}(\Sigma) - dim\ \mathcal{K}_{n,m,p}(\Omega), \tag{1.43}$$

if Algorithm 1 is used.

When the constraint matrix D contains $n+1$ linearly independent constraints on the state, the dimension of the uncertainty set is equal to dim $\mathcal{K}_{n,m,p}(\Sigma)$, which is less or equal to:

$$n(m+1) - \sum_{i=2}^{m} r_{i-1} r_i,$$

where r_i is given in Lemma 1.4.3.

Moreover, for any $\Omega \in \bar{\mathcal{S}}_{n,m,p}$ such that its corresponding $\Sigma \in \mathcal{S}_{n,m,p}$ is prime, the dimension of $[\Omega]_{\mathcal{K}}$ is greater or equal to

$$\sum_{i=1}^{m} r_{k_i} + m$$

.

Proof. The proof of this statement is similar to that of Corollary 1.4.9. The dimensions of equivalence classes for prime and general systems were evaluated using results of Proposition 1.4.3 and Lemma 1.4.6. □

In this scenario, by applying Algorithm 1, the plant would be able to conceal the state trajectory from the cloud.

To illustrate the main results of this section, consider the following example.

Example 1.4.11. Consider a drone with linearized dynamics given in [48] and a bounded operational envelope (i.e., constraints on the extreme values of its state). From the linear model in [48] we observe that $n = 12$, $m = 4$, $p = 4$ and $r_1 = 4$, $r_2 = 4$, $r_3 = 2$, $r_4 = 2$. Suppose we decide to offload the control of this drone to the cloud. Let us evaluate the privacy guarantees Algorithm 1 can provide in each of the scenarios described in Section 1.2.

In the first scenario, when the cloud has no prior knowledge about the drone, we can choose any $\psi \in \mathcal{G}_{n,m,p}$. Therefore, using Propositon 1.4.7, we estimate the dimension of the uncertainty set to be greater than 212.

In the second scenario, when the cloud knows what sensors and actuators the drone has, we must choose an isomorphism $\psi \in \mathcal{H}_{n,m,p}(\Sigma)$ to keep inputs and outputs consistent.

A practical example of this could be if the cloud was owned by a company that provides computations specifically for drones. In this case, we use Corollary 1.4.9 and estimate the dimension of the uncertainty set to be greater than 156.

Finally, when the cloud has complete knowledge about the plant, we are forced to choose a symmetry $\psi \in \mathcal{K}_{n,m,p}(\Sigma)$ to keep the dynamics unchanged. This scenario could, for example, occur if the cloud belongs to the drone's manufacturer. Using Corollary 1.4.10, we estimate the dimension of the uncertainty set to be less or equal than 32. Unfortunately, we generally cannot provide a guarantee for the lower bound in this scenario. The dimension of the uncertainty set, however, can be found exactly by determining $\mathcal{K}_{n,m,p}(\Sigma)$ for a given Σ.

1.5 Side knowledge

The privacy guarantees derived in Section 1.4 are compromised when the adversary has partial information about the encoding isomorphism. In our problem formulation, we assume that the cloud may have learned those through some external channels or through some prior knowledge about the system.

Recall that by Lemma 1.3.7, $\mathcal{G}_{n,m,p}$ is a Lie group of dimension $n(n+1) + m(n+1) + m^2 + p(p+1)$. In this section, we assume that the constraint matrix D has $n+1$ linearly independent constraints on the state and, therefore, as shown in the previous section, $\mathcal{K}_{n,m,p}(\Omega) = \{\psi_e\}$, where ψ_e is the identity element of $\mathcal{G}_{n,m,p}$.

Suppose the cloud has partial knowledge about the encoding isomorphism. We shall represent the partial knowledge available to the cloud as a projection from $\mathcal{G}_{n,m,p}$ onto a k-dimensional vector space. Let us define $\rho : \mathcal{G}_{n,m,p} \to \mathbb{R}^k$ to be a surjective map of constant rank k, providing side knowledge about the encoding isomorphism. Then, we can say that the cloud knows some vector $l \in \mathbb{R}^k$, where:

$$l = \rho(P, F, G, S). \tag{1.44}$$

Note that this map is not known to us, and the results that follow do not require the knowledge of this map.

Side knowledge does not change the result of Theorem 1.4.2, however the privacy guaranteed by the scheme changes. It is obvious that the size of the uncertainty set defined by isomorphisms that satisfy (1.44) is no greater and, in general, smaller than if no side knowledge is available. Moreover, the uncertainty set is no longer neither an orbit nor an equivalence class because the preimage of ρ does not necessarily have a group structure.

Let us show that the object defined by (1.44) on $\mathcal{G}_{n,m,p}$ is still a manifold.

Lemma 1.5.1. *Let $\mathcal{G}_{n,m,p}$ be the group of all isomorphisms, $\rho : \mathcal{G}_{n,m,p} \to \mathbb{R}^k$ be a surjective map of constant rank k and assume the cloud knows that $l = \rho(P, F, G, S)$. Then, $\rho^{-1}(l)$, representing the possible encoding isomorphisms used by the client, is a properly embedded submanifold of $\mathcal{G}_{n,m,p}$. Its dimension is dim $\mathcal{G}_{n,m,p} - k$.*

Proof. By the global rank theorem [42, p. 83], since ρ is a surjective map of constant rank k, it is a smooth submersion. From the submersion level set theorem [42, p. 105], since both $\mathcal{G}_{n,m,p}$ and $\mathbb{R}^k$ are smooth manifolds and ρ is a smooth submersion, we have that $\rho^{-1}(l)$ is a properly embedded submanifold of dimension $\dim \mathcal{G}_{n,m,p} - \dim \mathbb{R}^k = n(n+1) + m(n+1) + m^2 + p(p+1) - k$. □

Let us now consider the map Θ_Ω defined earlier in Proposition 1.4.7. Since $\mathcal{K}_{n,m,p}(\Omega) = \psi_e$, we have that $\mathcal{G}_{n,m,p}/\mathcal{K}_{n,m,p}(\Omega)$ is equivalent to $\mathcal{G}_{n,m,p}$. Therefore, the map Θ_Ω is equivalent to the orbit map θ_Ω. It was shown in Proposition 1.4.7 that Θ_Ω is injective. The image of $\Theta_\Omega(\rho^{-1}(l))$ constitutes the uncertainty set, between the elements of which the cloud is not be able to distinguish. Therefore, the main result of this section requires finding the dimension of $\Theta_\Omega(\rho^{-1}(l))$.

Proposition 1.5.2. *Assume $\Omega \in \bar{\mathcal{S}}_{n,m,p}$ is such that the constraint matrix D has $n+1$ linearly independent constraints on the state. Suppose that Algorithm 1 is used and the cloud* *has the following side knowledge about the selected isomorphism ψ:*

$$\rho(P, F, G, S) = l \in \mathbb{R}^k,$$

where $\rho : \mathcal{G}_{n,m,p} \to \mathbb{R}^k$ is a surjective map of constant rank k. Then, under the scenario described in Problem 1.2.1, the cloud cannot distinguish between Ω and any other system in the uncertainty set $\mathcal{U} = \Theta_\Omega(\rho^{-1}(l))$ of dimension:

$$dim\ \mathcal{G}_{n,m,p} - k = n(n+1) + m(n+1) + m^2 + p(p+1) - k. \tag{1.45}$$

Proof. By Theorem 1.4.2, Algorithm 1 renders isomorphic systems indistinguishable by the cloud. However, the cloud knows that we use an isomorphism $\psi \in \rho^{-1}(l)$ and, therefore, the uncertainty set is no longer the equivalence class under the entire group of isomorphisms $\mathcal{G}_{n,m,p}$, but the subset of this equivalence class $\mathcal{U} = \Theta_\Omega(\rho^{-1}(l))$.

By the property of the orbit map [42, p. 166], for each Ω, the orbit map Θ_Ω is smooth and has constant rank. Since Θ_Ω is also injective, we have, by the Global Rank Theorem, that it is a smooth immersion [42, p. 83]. As it was shown in Lemma 1.5.1, the set $\rho^{-1}(l)$ is an embedded submanifold of $\mathcal{G}_{n,m,p}$ and, therefore, the inclusion map $i : \rho^{-1}(l) \to \mathcal{G}_{n,m,p}$ is a smooth embedding.

The map $\Theta_\Omega \circ i$ is a smooth immersion because it is a composition of smooth immersions. Since images of smooth immersions are smooth immersed submanifolds (by Proposition 5.18 from [42]), the uncertainty set $\mathcal{U} = \Theta_\Omega(\rho^{-1}(l))$ is a smooth immersed submanifold of $\bar{\mathcal{S}}_{n,m,p}$ diffeomorphic to $\rho^{-1}(l)$ and, hence, has the same dimension (refer to Lemma 1.5.1).

Using Lemma 1.3.7, the dimension of the uncertainty set is evaluated to be:

$$n(n+1) + m(n+1) + m^2 + p(p+1) - k.$$

□

Remark: although Proposition 1.5.2 was proved under the assumption that D has $n+1$ linearly independent constraints on the state, this assumption can be dropped if we assume the intersection of $\rho^{-1}(l)$ and the left cosets of $\mathcal{K}_{n,m,p}(\Omega)$ in $\mathcal{G}$ is well-behaved.

This result shows that the proposed scheme degrades gracefully with side knowledge — i.e., side knowledge allows the cloud to reduce the dimension of the uncertainty set only by the amount of side knowledge and not more. Moreover, this result can be generalized for other scenarios considered in Section 1.4.2.2 using similar proofs.

Corollary 1.5.3. *Assume $\Omega \in \bar{\mathcal{S}}_{n,m,p}$ is such that the constraint matrix D has $n+1$ linearly independent constraints on the state. Suppose that Algorithm 1 is used and the cloud has the following side knowledge $l \in \mathbb{R}^k$ about the selected isomorphism ψ:*

$$l = \rho(P, F, G, S),$$

where $\rho : \mathcal{G}_{n,m,p} \to \mathbb{R}^k$ is a surjective map of constant rank k. Then, under the scenario described in Problem 1.2.2, the cloud cannot distinguish between Ω and any other system in the uncertainty set $\mathcal{U} = \Theta_\Omega(\rho^{-1}(l))$ of dimension:

$$dim\ \mathcal{H}_{n,m,p}(\Sigma) - k. \tag{1.46}$$

Under the scenario described in Problem 1.2.3, the dimension of the uncertainty set is:

$$dim\ \mathcal{K}_{n,m,p}(\Sigma) - k. \tag{1.47}$$

1.6 Conclusions and future work

In this chapter, we proposed a transformation-based method to preserve privacy in control over the cloud. In addition to its low computational overhead, we have formally shown that this method precludes the adversary from inferring the private data by eavesdropping on the messages exchanged between the plant and the cloud. We quantified the guaranteed privacy via the dimension of the set that describes the uncertainty experienced by the adversary.

The problem of computing the dimension of the stabilizer set $\mathcal{K}_{n,m,p}(\Omega)$ remains open, and its solution requires a detailed analysis of system-theoretic properties. As future work, it would also be interesting to investigate other measures of privacy that may lead to a deeper insight into the proposed method. Moreover, similar techniques can be proposed for protecting privacy of non-linear control systems when controlling over the cloud. This will, however, likely involve expanding the isomorphism set beyond the set of affine transformations and require a different metric for privacy.

CHAPTER 2

Learning to control from expert demonstrations

2.1 Introduction

2.1.1 Motivation

The usefulness of learning from demonstrations has been well-argued in the literature (see [4–6]). In the context of control, imagine that we need to design a controller for an autonomous car that prioritizes comfort of its passengers. It is not obvious how to capture the idea of comfortable driving in a mathematical expression. It is fairly straightforward, however, to collect demonstrations of comfortable driving from human drivers. There are many other control tasks where providing examples of the desired behaviour is easier than defining such behaviour formally (e.g., teaching a robot to manipulate objects). The growing research interest in learning from demonstrations (LfD) for robot control [6] reflects the need for a well-defined controller design methodology for such tasks. In this work, we propose a methodology that uses expert demonstrations to construct a stabilizing controller.

There are many examples in the literature, where various LfD methodologies have been applied to robots [6]. The most popular application of LfD so far is in robotic manipulators. More specifically, LfD is used to teach manipulators skills to perform tasks in manufacturing [49], health-care [50, 51], and human-robot interaction [52, 53]. In addition, LfD has been applied with significant success to ground vehicles [54, 55], aerial vehicles [56, 57], bipedal robots [58, 59], and quadrupedal robots [60, 61]. These examples illustrate that, for these platforms, there exist control tasks for which LfD techniques are preferable to traditional control approaches.

2.1.2 Related work

In this section, we describe the previous work in learning from demonstrations to indicate where our approach lies within the existing landscape. This is in no way a comprehensive account of the literature on learning from demonstrations, but rather an overview of approaches related to ours (please refer to [6] or [62] for a description of the literature on LfD).

Policy-learning LfD methods, to which this work belongs, assume that there exists a mapping from state (or observations) to control input that dictates the expert's behaviour. This mapping is referred to as the expert's policy. The goal of these methods is to find (or approximate) the expert's policy given expert demonstrations. In many machine-learning-based LfD methods, policy learning is viewed as a supervised-learning problem where states and control inputs are treated as features and labels, respectively. We refer to these methods as behavioural cloning methods. Pioneered in the 80s by works like [63], this class of methods is still popular today. Behavioural cloning methods are typically agnostic to the nature of the expert — demonstrations can be provided by a human (see [54, 64]), an offline optimal controller (see [65,66]), or a controller with access to privileged state information (see [56,67]). They do, however, require a large number of demonstrations to work well in practice and, if trained solely on data from unmodified expert demonstrations, generate unstable policies that cannot recover from drifts or disturbances [54]. The latter problem can be fixed using online meta-algorithms like DAgger [68] which ensure that training data includes observations of recoveries from perturbations. Using such algorithms, however, comes at the expense of enlarging the training dataset. Moreover, the works on behavioural cloning typically provide few formal stability guarantees and, instead, illustrate performance with experiments.

Currently, there is a concerted effort to develop policy-learning LfD methods that improve on existing techniques using tools from control theory. In that context, the work that is closest to ours is described in [69], where the authors use convex optimization to construct a linear policy that is both close to expert demonstrations and stabilizes a linear system. They guarantee that the resulting controller is optimal with respect to some quadratic cost by adding an additional set of constraints (originally proposed in [70]) to the optimization problem. This work has been extended in [71] to enforce other properties, such as stability, optimality, and $\mathcal{H}_\infty$-robustness. Our methodology is different from those in [69] and [71] because we do not assume the expert to be a linear time-invariant controller.

2.1.3 Contributions

In this chapter, we propose a methodology for constructing a controller for a known nonlinear system from a finite number of expert demonstrations of desired behaviour, provided their number exceeds the number of states and the demonstrations are sufficiently long. Our approach consists of two steps:

- use feedback linearization to transform the nonlinear system into a chain of integrators;
- use affine combinations of demonstrations in the transformed coordinates to construct a control law stabilizing the original system.

The expert demonstrations are assumed to be of finite-length, whereas the resulting controller is expected to control the system indefinitely, making this a non-trivial problem to address. In this chapter, we formally prove the learned controller asymptotically stabilizes the system. Furthermore, in case there are more demonstrations than states, we determine which subset of demonstrations needs to be chosen to minimize the error between the trajectory of the learned controller and the trajectory of the expert controller. To demonstrate the feasibility of this methodology, we apply it to the problem of quadrotor control. Unlike [69], our methodology produces a controller that is time-varying and not linear in the original coordinates. This reflects our belief that, in many cases, the expert demonstration is produced by a nonlinear controller. We also extend the proposed methodology beyond the class of feedback linearizable systems by using the embedding technique described in [72] and demonstrate its feasibility on the classical example of the ball-and-beam system.

A preliminary version of this methodology was introduced in [73]. In [74], it was combined together with the data-driven control results from [75] to learn to control unknown SISO systems from demonstrations. This chapter provides a unified presentation of the results from [73], as well as several new results, such as the discussion on the optimality of the controller approximation error and the extension of the results beyond the class of feedback linearizable systems. The content of this chapter has been published in [76].

2.2 Problem Statement and Preliminaries

2.2.1 Notations and basic definitions

The notation used in this chapter is fairly standard. The integers are denoted by $\mathbb{Z}$, the natural numbers, including zero, by $\mathbb{N}_0$, the real numbers by $\mathbb{R}$, the positive real numbers by $\mathbb{R}^+$, and the non-negative real numbers by $\mathbb{R}^+_0$. We denote by $\|\cdot\|$ (or by $\|\cdot\|_2$) the standard Euclidean norm or the induced matrix 2-norm; and by $\|\cdot\|_F$ the matrix Frobenius norm. A set of vectors $\{v_1, \ldots, v_k\}$ in $\mathbb{R}^n$ is affinely independent if the set $\{v_2 - v_1, \ldots, v_k - v_1\}$ is linearly independent.

A function $\alpha : \mathbb{R}^+_0 \to \mathbb{R}^+_0$ is of class $\mathcal{K}$ if α is continuous, strictly increasing, and $\alpha(0) = 0$. If α is also unbounded, it is of class $\mathcal{K}_\infty$. A function $\beta : \mathbb{R}^+_0 \times \mathbb{R}^+_0 \to \mathbb{R}^+_0$ is of class $\mathcal{KL}$ if, for fixed $t \geq 0$, $\beta(\cdot, t)$ is of class $\mathcal{K}$ and $\beta(r, \cdot)$ decreases to 0 as $t \to \infty$ for each fixed $r \geq 0$.

The Lie derivative of a function $h : \mathbb{R}^n \to \mathbb{R}$ along a vector field $f : \mathbb{R}^n \to \mathbb{R}^n$, given by $\frac{\partial h}{\partial x} f$, is denoted by $L_f h$. We use the notation $L_f^k h$ for the iterated Lie derivative, i.e., $L_f^k h = L_f(L_f^{k-1} h)$, with $L_f^0 h = h$. Given open sets $U \subseteq \mathbb{R}^n$ and $V \subseteq \mathbb{R}^n$, a smooth map $\Phi : U \to V$ is called a diffeomorphism from U to V if it is a bijection and its inverse $\Phi^{-1} : V \to U$ is smooth.

Consider the continuous-time system:

$$\dot{x} = f(t, x), \tag{2.1}$$

where $x \in \mathbb{R}^n$ is the state and $f : \mathbb{R}_0^+ \times \mathbb{R}^n \to \mathbb{R}^n$ is a smooth function. The origin of (2.1) is uniformly asymptotically stable if there exist $\beta \in \mathcal{KL}$ and $c > 0$ such that, for all $\|x(t_0)\| < c$, the following is satisfied [77]:

$$\|x(t)\| \leq \beta(\|x(t_0)\|, t - t_0), \quad \forall t \geq t_0 \geq 0. \tag{2.2}$$

Consider the continuous-time control system:

$$\dot{x} = f(t, x, u), \tag{2.3}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the input, and $f : \mathbb{R}_0^+ \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is a smooth function. The system (2.3) is said to be input-to-state stable (ISS) if there exist $\beta \in \mathcal{KL}$ and $\gamma \in \mathcal{K}$ such that for any $x(t_0) \in \mathbb{R}^n$ and any bounded input $u : [t_0, \infty) \to \mathbb{R}^m$, the following is satisfied:

$$\|x(t)\| \leq \beta(\|x(t_0)\|, t - t_0) + \gamma\left(\sup_{t_0 \leq \tau \leq t} \|u(\tau)\|\right). \tag{2.4}$$

Let $\mathcal{X} = \{x_1, \ldots, x_k\}$ be a set of points in $\mathbb{R}^n$. A point $x = \sum_{i=1}^k \theta_i x_i$ with $\sum_{i=1}^k \theta_i = 1$ is called an affine combination of points in $\mathcal{X}$. If, in addition, $\theta_i \geq 0$ for all $i \in \{1, \ldots, k\}$, then x is a convex combination of points in $\mathcal{X}$.

2.2.2 Problem Statement

Consider a known continuous-time control-affine system:

$$\Sigma : \quad \dot{x} = f(x) + g(x)u, \tag{2.5}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ are the state and the input, respectively; and $f : \mathbb{R}^n \to \mathbb{R}^n$, $g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are smooth functions. Assume that the origin is an equilibrium point of

(2.5). We call a pair $(x,u) : \mathbb{R}_0^+ \to \mathbb{R}^n \times \mathbb{R}^m$ a solution of the system (2.5) if, for all $t \in \mathbb{R}_0^+$, the equation (2.5) is satisfied. Furthermore, we refer to the functions x and u as a trajectory and a control input of the system (2.5).

We say that a controller $k : \mathbb{R}^n \to \mathbb{R}^m$ is asymptotically stabilizing for the system (2.5) if the origin is uniformly asymptotically stable for the system (2.5) with $u = k(x)$. Suppose there exists an unknown asymptotically stabilizing controller k, which we call the expert controller. We assume that k is smooth. Our goal is to learn a controller $\widehat{k} : \mathbb{R}_0^+ \times \mathbb{R}^n \to \mathbb{R}^m$ such that having $u = \widehat{k}(t,x)$ asymptotically stabilizes the origin of the system (2.5). Towards this goal, we use a set of M finite-length expert solutions $\mathcal{D} = \{(x^i, u^i)\}_{i=1}^M$ of (2.5), where: for each i, the trajectory $x^i : [0,T] \to \mathbb{R}^n$ and the control input $u^i : [0,T] \to \mathbb{R}^m$ are smooth and satisfy $u^i(t) = k(x^i(t))$ for all $t \in \mathbb{R}_0^+$; $T \in \mathbb{R}_0^+$ is the length of a solution; and $M \geq n+1$. We also ascertain that the "trivial" expert solution, wherein $x(t) = 0$ and $u(t) = 0$ for all $t \in [0,T]$, is included in $\mathcal{D}$.

Remark 2.2.1. In practice, we can record the values of continuous solutions provided by the expert only at certain sampling instants. In this work, however, we choose to work in continuous-time to simplify the theoretical analysis. We can do this without sacrificing practical applicability because it is well-known that continuous-time controller designs can be implemented via emulation and still guarantee stability [78].

We make the assumption that the system (2.5) is feedback linearizable on an open set $U \subseteq \mathbb{R}^n$ containing the origin and the expert demonstrations $x^i(t)$ belong to U for all $t \in [0,T]$. To avoid the cumbersome notation that comes with feedback linearization of multiple-input systems, we assume that $m = 1$, that is, the system (2.5) only has a single input. Readers familiar with feedback linearization can verify that all the results extend to multiple-input case, mutatis mutandis (refer to [79, Ch. 4-5] for a complete introduction to feedback linearization). In the single-input case, the system (2.5) is feedback linearizable on the open set $U \subseteq \mathbb{R}^n$ if there is an output function $h : \mathbb{R}^n \to \mathbb{R}$ that has relative degree n, i.e., for all $x \in U$, $L_g L_f^i h(x) = 0$ for $i = 0, \ldots, n-2$ and $L_g L_f^{n-1} h(x) \neq 0$. Moreover, the map:

$$z = \Phi(x) = \begin{bmatrix} h(x) & L_f h(x) & \cdots & L_f^{n-1} h(x) \end{bmatrix}^T, \tag{2.6}$$

is a diffeomorphism from U to its image $\Phi(U)$, i.e., the inverse $\Phi^{-1} : \Phi(U) \to U$ exists and is also smooth. We further assume, without loss of generality, that $h(0) = 0$.

2.3 Learning a stabilizing controller from $n+1$ expert demonstrations

Here, we describe the methodology for constructing an asymptotically stabilizing controller when $M = n + 1$. We consider the case when $M \geq n + 1$ in Section 2.4.

2.3.1 Feedback linearization

Recall that using the feedback linearizability assumption, we can rewrite the system dynamics (2.5) in the coordinates given by (2.6) resulting in:

$$\begin{aligned} \dot{z}_1 &= z_2, \\ &\vdots \\ \dot{z}_{n-1} &= z_n, \\ \dot{z}_n &= a(z) + b(z)u, \end{aligned} \tag{2.7}$$

where $a = \left(L_f^n h\right) \circ \Phi^{-1}$ and $b = \left(L_g L_f^{n-1} h\right) \circ \Phi^{-1}$. The feedback law:

$$u = b(z)^{-1}(-a(z) + v), \tag{2.8}$$

further transforms the system (2.5) into the system given by:

$$\dot{z} = Az + Bv, \tag{2.9}$$

where (A, B) is a Brunovsky pair.

Remark 2.3.1. The expert controller $\kappa : \mathbb{R}^n \to \mathbb{R}$ in the (z, v)-coordinates is given by $\kappa(z) = a(z) + b(z)k(\Phi^{-1}(z))$. The smoothness of k implies that the function κ is also smooth.

2.3.2 Expert demonstrations

Recall that the set of demonstrations $\mathcal{D}$ consists of solutions of the system (2.5). Using (2.6) and (2.8), we can represent the demonstrations $\mathcal{D}$ in (z, v)-coordinates. We denote the resulting set by $\mathcal{D}_{(z,v)} = \{(z^i, v^i)\}_{i=1}^M$, where functions $z^i : [0, T] \to \mathbb{R}^n$ and $v^i : [0, T] \to \mathbb{R}$ are given by:

$$z^i(t) \triangleq \Phi(x^i(t)) \tag{2.10}$$

$$v^i(t) \triangleq L_f^n h(x^i(t)) + L_g L_f^{n-1} h(x^i(t)) u^i(t), \tag{2.11}$$

for all $i \in \{1, \cdots, M\}$ and for all $t \in [0, T]$. We define the set of demonstrations $\mathcal{D}_{(z,v)}$ evaluated at time t as:

$$\mathcal{D}_{(z,v)}(t) = \{(z^i(t), v^i(t))\}_{i=1}^M.$$

It can be easily verified that the demonstrations in $\mathcal{D}_{(z,v)}$ satisfy the dynamics (2.9) and $v^i(t) = \kappa(z^i(t))$.

2.3.3 Constructing the learned controller

We denote by $\widehat{\kappa}(t, z)$ the controller learned from the expert demonstrations. We begin by partitioning time into intervals of length T and indexing these intervals with $p \in \mathbb{N}_0$. Let us construct the following matrices for $t \in [0, T]$:

$$Z(t) \triangleq \left[z^2(t) - z^1(t) \,\middle|\, \cdots \,\middle|\, z^{n+1}(t) - z^1(t) \right] \tag{2.12}$$

$$V(t) \triangleq \left[v^2(t) - v^1(t) \,\middle|\, \cdots \,\middle|\, v^{n+1}(t) - v^1(t) \right]. \tag{2.13}$$

Our first attempt at constructing the learned controller, which we improve upon later, is to use the piecewise-continuous controller $v(t) = \widehat{\kappa}(t, z(pT))$ for all $t \in [pT, (p + 1)T]$, where:

$$\widehat{\kappa}(t, z(pT)) = V(t - pT)\zeta(p), \tag{2.14}$$

with $\zeta(p) = Z^{-1}(0)z(pT)$, and $Z(t), V(t)$ defined in (2.12) and (2.13), respectively.

The next lemma formally shows that an affine combination of trajectories of (2.9) is a valid trajectory for (2.9).

Lemma 2.3.2. *Suppose we are given a set of finite-length solutions* $\{(z^i, v^i)\}_{i=1}^{n+1}$ *of the system* (2.9)*, where each* (z^i, v^i) *is defined for* $0 \le t \le T$, $T \in \mathbb{R}^+$*. Assume that* $\{z^i(0)\}_{i=1}^{n+1}$ *is an affinely independent set. Then, under the control law* $v(t) = V(t - t_0)\zeta$ *with* $\zeta = Z^{-1}(0)z(t_0)$*, the solution of the system* (2.9) *is* $z(t) = Z(t - t_0)\zeta$*, for* $t_0 \le t \le T + t_0$*, where* $Z(t)$ *and* $V(t)$ *are defined in* (2.12) *and* (2.13)*, respectively.*

Proof. This lemma can be verified by substitution. □

Remark 2.3.3. Affine independence of the set $\{z^i(0)\}_{i=1}^{n+1}$ is a generic property, i.e., this is true for almost all expert demonstrations. In practice, if this set is not affinely independent, a user can eliminate the affinely dependent demonstrations and request the expert to provide additional demonstrations.

We note, however, that the control law (2.14) samples the state z with a sampling time T and essentially operates in open loop in between these samples. To allow for closed-loop control, we propose the improved controller that has, for all $t \in [pT, (p+1)T]$, the following form:

$$\begin{aligned} v(t) = \widehat{\kappa}(t, z(t)) = V(t - pT)\zeta(p, t), \\ \zeta(p, t) = Z^{-1}(t - pT)z(t). \end{aligned} \tag{2.15}$$

In the absence of uncertainties and disturbances, by Lemma 2.3.2, the coefficients ζ satisfy:

$$\zeta(p, t) = Z^{-1}(t - pT)z(t) = Z^{-1}(0)z(pT), \tag{2.16}$$

i.e., the controller (2.15) applies the input equal to that applied by the controller (2.14).

2.3.4 Stability of the learned controller

Assuming (2.16) holds, the system (2.9) in closed loop with (2.15) has the following form:

$$\dot{z} = Az + BV(t - pT)Z^{-1}(0)z(pT), \tag{2.17}$$

for all $t \in [pT, (p+1)T]$. Integrating the dynamics, we show that the sequence $\{z(pT)\}_{p\in\mathbb{N}_0}$ satisfies:

$$z((p+1)T) = \Psi(T)z(pT), \tag{2.18}$$

where:

$$\Psi(T) \triangleq e^{AT} + \int_0^T e^{A(T-\tau)}BV(\tau)Z^{-1}(0)\mathrm{d}\tau. \tag{2.19}$$

By adopting a term from Floquet's theory, we refer to $\Psi(T)$ in (2.19) as the closed-loop monodromy matrix [80].

This section's main result provides sufficient conditions for asymptotic stability of system (2.5) in closed loop with (2.8)-(2.15).

Theorem 2.3.4. *Consider the system* (2.5) *and assume it is feedback linearizable on an open set $U \subseteq \mathbb{R}^n$ containing the origin. Let $T \in \mathbb{R}^+$ and suppose we are given a finite set of demonstrations $\mathcal{D} = \{(x^i, u^i)\}_{i=1}^{n+1}$ generated by the system* (2.5)*, in closed loop with a smooth asymptotically stabilizing controller $k : \mathbb{R}^n \to \mathbb{R}$, and satisfying $x^i(t) \in U$ for all $t \in [0, T]$. Assume that $\{\Phi(x^i(t))\}_{i=1}^{n+1}$ is affinely independent for all $t \in [0, T]$. Then, there is a $\tilde{T} \in \mathbb{R}^+$ such that for all $T \geq \tilde{T}$, the origin of system* (2.5) *in closed-loop with controller* (2.8)-(2.15) *is uniformly asymptotically stable.*

Proof. The asymptotic stability of (2.5) and (2.9) are equivalent on U and $\Phi(U)$ [81], and, therefore, the set $\mathcal{D}_{(z,v)}$ given by (2.10) and (2.11) also consists of asymptotically stable solutions, i.e., there exists $\beta \in \mathcal{KL}$ such that for all $i \in \{1, ..., n+1\}$:

$$\|z^i(t)\| \leq \beta(\|z^i(0)\|, t), \quad \forall t \in \mathbb{R}_0^+. \tag{2.20}$$

Consider the closed-loop system (2.17). By Lemma 2.3.2:

$$z((p+1)T) = Z(T)Z^{-1}(0)z(pT), \quad \forall T \in \mathbb{R}_0^+.$$

Combining this with (2.18) implies that:

$$\Psi(T) = Z(T)Z^{-1}(0). \tag{2.21}$$

We claim that, for any constants $a, b, c > 0$, there exists $t \in \mathbb{R}^+$ such that $\beta(r,t) < c$ for all $r \in [a,b]$. This claim will be shown using an argument similar to that of the proof of Lemma 16 in [82]. Using Lemma 4.3 from [83], there exist class $\mathcal{K}_\infty$ functions σ_1, σ_2 such that $\beta(r,t) \leq \sigma_1(\sigma_2(r)e^{-t})$ for all $r, t \in \mathbb{R}_0^+$. Let $0 < \varepsilon < c$. Define, for all $r \in \mathbb{R}^+$, $t(r)$ to be the solution of $\sigma_1(\sigma_2(r)e^{-t}) = c - \varepsilon$ and obtain:

$$t(r) = -\log \frac{\sigma_1^{-1}(c-\varepsilon)}{\sigma_2(r)}.$$

Since $t(r)$ is a continuous function and $[a,b]$ is compact, the extreme value theorem implies that $t^* = \max_{r\in[a,b]} t(r)$ is well-defined. For all $r \in [a,b]$, it is true that:

$$\beta(r,t^*) \leq \sigma_1(\sigma_2(r)e^{-t^*}) \leq c - \varepsilon < c.$$

Using the previous claim with $a = \min_{i\in\{1,\ldots,n+1\}} \|z^i(0)\|$, $b = \max_{i\in\{1,\ldots,n+1\}} \|z^i(0)\|$ and $c = 1/\left(2\sqrt{n}\,\|Z^{-1}(0)\|\right)$, we conclude the existence of $\tilde{T} \in \mathbb{R}^+$ such that, for all $T \geq \tilde{T}$, the following inequality holds:

$$\beta(\|z^i(0)\|, T) < \frac{1}{2\sqrt{n}\|Z^{-1}(0)\|},$$

for all $i \in \{1,\ldots,n+1\}$. Therefore, by (2.20), for all $i \in \{1,\ldots,n+1\}$ and for all $T \geq \tilde{T}$, we have:

$$\|z^i(T)\| < \frac{1}{2\sqrt{n}\|Z^{-1}(0)\|}. \tag{2.22}$$

Using (2.21) and (2.22), for all $T \geq \tilde{T}$, we have:

$$\begin{aligned} \|\Psi(T)\| &\leq \|Z(T)\| \left\|Z^{-1}(0)\right\| \leq \|Z(T)\|_F \left\|Z^{-1}(0)\right\| \\ &= \left(\sum_{i=2}^{n+1} \left\|z^i(T) - z^1(T)\right\|^2 \right)^{\frac{1}{2}} \left\|Z^{-1}(0)\right\| \\ &< \frac{\sqrt{n}}{\sqrt{n}\left\|Z^{-1}(0)\right\|} \cdot \left\|Z^{-1}(0)\right\| < 1. \end{aligned} \tag{2.23}$$

According to stability conditions for linear discrete-time systems (see Theorem 10.9 in [46]), the equation (2.23) implies that, for all $T > \tilde{T}$, the system (2.18) is uniformly exponentially stable. From [80], we know that uniform exponential stability of the sampled-data system (2.18) implies uniform exponential stability of the system (2.9)-(2.15) because the matrices $\Psi(t)$ are bounded for $t \in [0, T]$. Uniform asymptotic stability of the origin for the system (2.9)-(2.15) in the (z, v)-coordinates implies uniform asymptotic stability of the origin for the feedback equivalent system (2.5)-(2.8)-(2.15) in (x, u)-coordinates [81]. □

Remark 2.3.5. Theorem 2.3.4 shows the existence of $\tilde{T} \in \mathbb{R}^+$ such that $\|\Psi(T)\| < 1$ for all $T \geq \tilde{T}$. In practice, a user can determine $T \in \mathbb{R}^+$ satisfying this condition by directly computing $\|\Psi(t)\| = \|Z(t)Z^{-1}(0)\|$ for various $t \in \mathbb{R}_0^+$.

Remark 2.3.6. The fact that we assume feedback linearizability on some open set $U \subseteq \mathbb{R}^n$ presents the user with the opportunity to use either local or global feedback linearization results, depending on what their application allows for. We recommend [79] as a good starting point to find conditions for both local (see Theorem 4.2.3 in [79]) and global (see Theorem 9.1.1 in [79]) feedback linearizability.

Remark 2.3.7. In Theorem 2.3.4, we provide a guarantee the learned controller $\widehat{k}$ stabilizes the system at the origin. This result can also be useful when the objective of the learned controller is to track a trajectory. The key idea is to recast the problem of trajectory tracking into that of stabilizing the error dynamics (see Section 4.5 in [79]). We consider this generality of the learned controller to be a strength of this approach. We will experimentally illustrate this in Section 2.6.1.

Remark 2.3.8. Although we assume in this work an exact knowledge of the state, in most applications, the state is estimated via an observer. Depending on the design of the observer, the stability results of our methodology may also vary. To give an example, using Lemma III.8 from [74], we can show that, with a well-designed sampled-data observer providing state estimates of both the expert demonstrations and the current state, we can still retain asymptotic stability. In general, however, a persistent error between the state estimate and the current state can weaken the guarantee of asymptotic stability guarantee of the closed-loop system to that of practical stability.

2.4 Learning from more than $n+1$ expert demonstrations

Here, we extend the previous results to the case where more than $M > n+1$. For every interval of length T, we show how to select a subset of $n+1$ demonstrations that results in the best approximation of the expert controller.

2.4.1 Preliminaries

We begin by reviewing several key concepts from multivariate linear interpolation. Let $\mathcal{X} = \{x_1, \dots, x_k\}$ be a finite set of points in $\mathbb{R}^n$. The convex hull of a set $\mathcal{X}$, denoted $\operatorname{conv} \mathcal{X}$, is the set of all convex combinations of points in $\mathcal{X}$ [84]. For any $\mathcal{I} \subset \{1, \dots, k\}$, we define the subset $\mathcal{X}_{\mathcal{I}} = \{x_i \in \mathcal{X} \mid i \in \mathcal{I}\}$. A Cartesian product of two sets $\mathcal{X} \times \mathcal{Y}$ has a natural left projection map $\pi_1 : \mathcal{X} \times \mathcal{Y} \to \mathcal{X}$ (resp., right projection map $\pi_2 : \mathcal{X} \times \mathcal{Y} \to \mathcal{Y}$) given by $\pi_1(x, y) = x$ (resp., $\pi_2(x, y) = y$). An n-simplex S is the convex hull of a set $\mathcal{X}' = \{x'_1, \dots, x'_{n+1}\}$ of $n+1$ affinely independent points. A triangulation of points in $\mathcal{X}$, denoted $\mathcal{T}(\mathcal{X})$, is a collection of n-simplices such that their vertices are points in $\mathcal{X}$, their interiors are disjoint, and their union is $\operatorname{conv} \mathcal{X}$. We denote the n-simplex in $\mathcal{T}(\mathcal{X})$ containing $x \in \operatorname{conv} \mathcal{X}$ by $S_{\mathcal{T}}(x)$ and define a vertex index set associated with x in $\mathcal{T}(\mathcal{X})$, denoted $\mathcal{I}_{\mathcal{T}}(x)$, as to satisfy $S_{\mathcal{T}}(x) = \operatorname{conv} \mathcal{X}_{\mathcal{I}_{\mathcal{T}}(x)}$. The Delaunay triangulation of $\mathcal{X}$, denoted

$\mathcal{DT}(\mathcal{X})$, is a triangulation with the property that the circum-hypersphere of every n-simplex in the triangulation contains no point from $\mathcal{X}$ in its interior. It is unique if no $n+1$ points are on the same hyperplane and no $n+2$ points are on the same hypersphere [85].

Let $\psi : \mathbb{R}^n \to \mathbb{R}^m$ be an unknown function. Given a finite set of points $\mathcal{X} = \{x_1, \ldots, x_k\} \subset \mathbb{R}^n$ and a set of function values $\mathcal{Y} = \{y_1, \ldots, y_k\} \triangleq \{\psi(x_1), \ldots, \psi(x_k)\}$, an interpolant $\widehat{\psi}^{\mathcal{X},\mathcal{Y}} : \operatorname{conv} \mathcal{X} \to \mathbb{R}^m$ is an approximation of ψ that satisfies $\widehat{\psi}(x) = \psi(x)$ for all $x \in \mathcal{X}$. We define an interpolant $\widehat{\psi}_{\mathcal{T}}^{\mathcal{X},\mathcal{Y}} : \operatorname{conv} \mathcal{X} \to \mathbb{R}^m$, called a piecewise-linear interpolant based on $\mathcal{T}(\mathcal{X})$, as:

$$\widehat{\psi}_{\mathcal{T}}^{\mathcal{X},\mathcal{Y}}(x) = \sum_{i \in \mathcal{I}_{\mathcal{T}}(x)} \theta_i y_i,$$

where $\theta_i \geq 0$ satisfy:

$$x = \sum_{i \in \mathcal{I}_{\mathcal{T}}(x)} \theta_i x_i, \qquad \sum_{i \in \mathcal{I}_{\mathcal{T}}(x)} \theta_i = 1.$$

2.4.2 Constructing the learned controller

Let us describe the construction of the controller $v = \widehat{\kappa}(t, z)$ for $M \geq n+1$. Define $\mathcal{Z}(t) = \pi_1\left(\mathcal{D}_{(z,v)}(t)\right)$ and $\mathcal{V}(t) = \pi_2\left(\mathcal{D}_{(z,v)}(t)\right)$. We partition time into intervals of length T, indexed by $p \in \mathbb{N}_0$. For each $[pT, (p+1)T]$, we propose using the piecewise-continuous control law $v(t) = \widehat{\kappa}(t, z(t))$, where $\widehat{\kappa}(\tau, \xi)$ is defined as follows:

(i) For $\xi \in \operatorname{conv} \mathcal{Z}(\tau - pT)$, the value of $\widehat{\kappa}(\tau, \xi)$ is given by the value at ξ of a piecewise-linear interpolant $\widehat{\psi}_{\mathcal{T}}^{\mathcal{Z}(\tau-pT),\mathcal{V}(\tau-pT)}$. Since a piecewise-linear interpolant is determined by an associated triangulation $\mathcal{T}(\mathcal{Z}(\tau - pT))$ [85], this implies that there is a family of possible learned controllers we can construct from $\mathcal{D}_{(z,v)}$. Moreover, the value of the interpolant depends only on the values of $\mathcal{Z}_{\mathcal{I}_{\mathcal{T}}(\xi)}(\tau - pT)$ and $\mathcal{V}_{\mathcal{I}_{\mathcal{T}}(\xi)}(\tau - pT)$, where $\mathcal{I}_{\mathcal{T}}(\xi)$ is a vertex set associated with ξ in $\mathcal{T}(\mathcal{Z}(\tau - pT))$.

(ii) For $\xi \notin \operatorname{conv} \mathcal{Z}(\tau - pT)$, let ξ^* be the Euclidean projection of ξ onto $\operatorname{conv} \mathcal{Z}(\tau - pT)$. Define the index set $\mathcal{I}_{\mathcal{T}}(\xi) = \mathcal{I}_{\mathcal{T}}(\xi^*)$ and express ξ as an affine combination

$\xi = \sum_{i \in \mathcal{I}_{\mathcal{T}(\xi)}} \theta_i z^i(0)$. Then, the value of $\widehat{\kappa}(\tau, \xi)$ is given by $\widehat{\kappa}(\tau, \xi) = \sum_{i \in \mathcal{I}_{T}(\xi)} \theta_i v^i(\tau - pT)$.

In both cases, the controller can be concisely expressed if, given a vertex index set $\mathcal{I} = \{i_1, \ldots, i_{n+1}\}$ for $\mathcal{Z}(t)$ and $\mathcal{V}(t)$, we construct the following matrices:

$$Z_{\mathcal{I}}(t) \triangleq \left[z^{i_2}(t) - z^{i_1}(t) \,\middle|\, \cdots \,\middle|\, z^{i_{n+1}}(t) - z^{i_1}(t) \right] \tag{2.24}$$

$$V_{\mathcal{I}}(t) \triangleq \left[v^{i_2}(t) - v^{i_1}(t) \,\middle|\, \cdots \,\middle|\, v^{i_{n+1}}(t) - v^{i_1}(t) \right], \tag{2.25}$$

for $t \in [0, T]$. Then, using (2.24) and (2.25), the proposed control law, for all $t \in [pT, (p+1)T]$, is given by:

$$\begin{aligned} v(t) = \widehat{\kappa}_{\mathcal{T}}(t, z(t)) = V_{\mathcal{I}_{\mathcal{T}}(z(t))}(t - pT)\zeta(p, t) \\ \zeta(p, t) = Z^{-1}_{\mathcal{I}_{\mathcal{T}}(z(t))}(t - pT)z(t). \end{aligned} \tag{2.26}$$

Note that, in the absence of uncertainties and disturbances, by Lemma 2.3.2, the coefficients satisfy:

$$\begin{aligned} \zeta(p, t) &= Z^{-1}_{\mathcal{I}_{\mathcal{T}}(z(t))}(t - pT)z(t) \\ &= Z^{-1}_{\mathcal{I}_{\mathcal{T}}(z(pT))}(0)z(pT). \end{aligned} \tag{2.27}$$

Therefore, for all $t \in [pT, (p+1)T]$, the controller (2.26) applies the input equal to that applied by the following controller:

$$\begin{aligned} v(t) = \widehat{\kappa}_{\mathcal{T}}(t, z(pT)) = V_{\mathcal{I}_{\mathcal{T}}(z(pT))}(t - pT)\zeta(p) \\ \zeta(p) = Z^{-1}_{\mathcal{I}_{\mathcal{T}}(z(pT))}(0)z(pT). \end{aligned} \tag{2.28}$$

Incidentally, this corresponds to the value of the piecewise-linear interpolant $\widehat{\psi}^{\mathcal{T}}_{\mathcal{Z}(0), \mathcal{V}(t-pT)}$ at $z(pT)$.

2.4.3 Stability of the learned controller

Let us define the collection of index sets $\mathcal{P} = \{\mathcal{I}_1, \ldots, \mathcal{I}_P\}$, where each $\mathcal{I}_j$ selects vertices of an n-simplex in $\mathcal{T}(\mathcal{Z}(0))$ and $P = |\mathcal{T}(\mathcal{Z}(0))|$. Note that $\mathcal{P}$ is a finite set because there are only finitely many n-simplices in $\mathcal{T}(\mathcal{Z}(0))$. Suppose the index set associated with $z(pT)$ in $\mathcal{T}(\mathcal{Z}(0))$ is $\mathcal{I}_{\mathcal{T}}(z(pT)) = \mathcal{I}_{j(p)}$ for some $j(p) \in \{1, \ldots, P\}$. Assuming (2.27) holds, the system (2.9) in closed loop with (2.26) is given by:

$$\dot{z} = Az + BV_{\mathcal{I}_{j(p)}}(t - pT)Z^{-1}_{\mathcal{I}_{j(p)}}(0)z(pT), \tag{2.29}$$

for all $t \in [pT, (p+1)T]$. Integrating the dynamics shows that the sequence $\{z(pT)\}_{p\in\mathbb{N}_0}$ satisfies:

$$z((p+1)T) = \Psi_{j(p)}(T)z(pT), \tag{2.30}$$

where

$$\Psi_{j(p)}(T) \triangleq e^{AT} + \int_0^T e^{A(T-\tau)}BV_{\mathcal{I}_{j(p)}}(\tau)Z^{-1}_{\mathcal{I}_{j(p)}}(0)\mathrm{d}\tau.$$

Note that now, instead of a single monodromy matrix, we have a set of monodromy matrices $\{\Psi_j(T)\}_{j=1}^P$.

The following result is an extension of Theorem 2.3.4 for $M \geq n+1$ demonstrations.

Theorem 2.4.1. *Consider the system* (2.5) *and assume it is feedback linearizable on an open set $U \subseteq \mathbb{R}^n$ containing the origin. Let $T \in \mathbb{R}^+$ and suppose we are given a finite set of demonstrations $\mathcal{D} = \{(x^i, u^i)\}_{i=1}^M$ generated by the system* (2.5)*, in closed loop with a smooth asymptotically stabilizing controller $k : \mathbb{R}^n \to \mathbb{R}$, and satisfying $x^i(t) \in U$ for all $t \in [0, T]$. Assume that $\{\Phi(x^i(t))\}_{i=1}^M$ is affinely independent for all $t \in [0, T]$. Then, there exists a $\tilde{T} \in \mathbb{R}^+$ such that for all $T \geq \tilde{T}$, the origin of system* (2.5) *in closed-loop with controller* (2.8)-(2.26) *is uniformly asymptotically stable.*

Proof. The proof of Theorem 2.3.4 implies the existence of $\tilde{T}_j \in \mathbb{R}$ such that $\|\Psi_j(t)\| < 1$ for all $t \geq \tilde{T}_j$. We choose $\tilde{T} = \max_{j\in\{1,\ldots,P\}} T_j$. The system (2.9) in closed loop with controller (2.26) can be represented as a switched system (2.30), where $j(p) \in \{1, \ldots, P\}$ is a switching sequence. By Theorem 3 in [86], the fact that $\|\Psi_j(T)\| < 1$ for all $T \geq \tilde{T}$ and $j \in \{1, \ldots, P\}$ implies that, for any switching signal $j(p)$, the system (2.30) is uniformly exponentially stable. Since the matrices $\Psi_j(t)$ are bounded for $t \in [0,T]$, the system (2.9) in closed loop with controller (2.26) is uniformly exponentially stable. Uniform asymptotic stability of the origin for the system (2.9)-(2.26) in the (z,v)-coordinates implies uniform asymptotic stability of the origin for the feedback equivalent system (2.5)-(2.8)-(2.26) in (x,u)-coordinates [81]. □

2.4.4 Optimality of the learned controller

Recall that the piecewise-linear interpolant defining the controller $\widehat{\kappa}_{\mathcal{T}}$ depends on the choice of the triangulation $\mathcal{T}(\mathcal{Z}(t-pT))$. Assuming (2.27) holds, this choice reduces to the choice of the triangulation $\mathcal{T}(\mathcal{Z}(0))$, which dictates the index set of demonstrations $\mathcal{I}_{\mathcal{T}}(z(pT))$ used to construct the solution for each interval $[pT,(p+1)T]$. Without loss of generality, in what follows we discuss the solutions on the interval $[0,T]$ only — a solution on $[pT,(p+1)T]$ can be represented as a solution on $[0,T]$ with the initial condition equal to $z(pT)$.

Typically, there are several triangulations one can define given a set of sample points $\mathcal{Z}(0)$. We want our choice of triangulation to result in closed-loop trajectories that approximate expert trajectories well for any initial state $z_0 \in \operatorname{conv} \mathcal{Z}(0)$ distinct from $\mathcal{Z}(0)$. More precisely, we want to find a triangulation $\mathcal{T}(\mathcal{Z}(0))$ that best approximates the function $\phi : [0,T] \times \operatorname{conv} \mathcal{Z}(0) \to \mathbb{R}^n$, which defines solutions of (2.9) under the expert controller κ, by the function $\widehat{\phi}_{\mathcal{T}} : [0,T] \times \operatorname{conv} \mathcal{Z}(0) \to \mathbb{R}^n$, which defines the solutions of (2.9) under the learned controller $\widehat{\kappa}_{\mathcal{T}}$. That is, we want solution to:

$$\min_{\mathcal{T}(\mathcal{Z}(0))} \sup_{\phi \in \mathcal{F}} \max_{t \in [0,T]} \left\| \phi(t,z_0) - \widehat{\phi}_{\mathcal{T}}(t,z_0) \right\|, \tag{2.31}$$

where $\mathcal{F}$ is the class of functions to which the expert solutions belong. We can view (2.31) as a game where we pick $\mathcal{T}(\mathcal{Z}(0))$, and the adversary, upon seeing our choice of $\mathcal{T}(\mathcal{Z}(0))$, picks ϕ to maximize the cost.

Let us leverage the properties $\phi(t,z_0)$ has by virtue of describing solutions of (2.9) under the expert controller κ to determine the class $\mathcal{F}$. We will use the notation $\phi_t : \mathbb{R}^n \to \mathbb{R}^n$ for

$\phi_t(z_0) = \phi(t, z_0)$. By Theorem 4.1 in [87, Ch. V], since κ is a smooth function, the Hessians of the coordinate functions of the solution $\frac{\partial^2 \phi_i}{\partial z_0^2}(t, z_0)$ are continuous with respect to t and z_0. By the extreme value theorem, compactness of $\operatorname{conv} \mathcal{Z}(0)$ implies that, for every i, there exists $H \in \mathbb{R}_0^+$ such that $\left\| \frac{\partial^2 \phi_i}{\partial z_0^2}(t, z_0) \right\| \leq H_i$ for all $t \in [0, T]$ and $z_0 \in \operatorname{conv} \mathcal{Z}(0)$. Thus, the norms of the Hessians of the coordinate functions can be bounded by $H = \max\{H_1, \ldots, H_n\}$. We denote the class of functions whose coordinate functions have the Hessian norm smaller or equal to H by $\mathcal{F}(H)$. For a fixed $t \in [0, T]$, $\phi_t \in \mathcal{F}(H)$ and, therefore, the function ϕ belongs to $\mathcal{F}(H)^{[0,T]}$, the set of all functions from $[0, T]$ to $\mathcal{F}(H)$.

Definition 2.4.2. For any $z_0 \in \operatorname{conv} \mathcal{Z}(0)$ and any learned controller $\kappa_{\mathcal{T}}$, the worst-case trajectory approximation error on the interval $[0, T]$ is given by:

$$\sup_{\phi \in \mathcal{F}(H)^{[0,T]}} \max_{t \in [0,T]} \left\| \phi(t, z_0) - \widehat{\phi}_{\mathcal{T}}(t, z_0) \right\|,$$

where $\phi : [0, T] \times \mathbb{R}^n \to \mathbb{R}^n$ is the trajectory of the system (2.9) with the initial condition z_0 under the expert controller κ, $\widehat{\phi}_{\mathcal{T}} : [0, T] \times \mathbb{R}^n \to \mathbb{R}^n$ is the trajectory of the system (2.9) with the same initial condition z_0 under the learned controller $\widehat{\kappa}_{\mathcal{T}}$, and $\mathcal{F}(H)^{[0,T]}$ is the set of all functions from $[0, T]$ to $\mathcal{F}(H)$. The smallest worst-case trajectory approximation error on the interval $[0, T]$ is given by:

$$\min_{\mathcal{T}(\mathcal{Z}(0))} \sup_{\phi \in \mathcal{F}(H)^{[0,T]}} \max_{t \in [0,T]} \left\| \phi(t, z_0) - \widehat{\phi}_{\mathcal{T}}(t, z_0) \right\|. \tag{2.32}$$

The following lemma by Omohundro [88] shows that the Delaunay triangulation leads to the best worst-case piecewise-linear interpolation for functions in $\mathcal{F}(H)$. For an efficient implementation of piecewise-linear interpolation based on the Delaunay triangulation, we refer the reader to [85].

Lemma 2.4.3 ([88]). *Let $\psi : \mathbb{R}^n \to \mathbb{R}^m$ satisfy the bounded Hessian norm property, i.e., $\psi \in \mathcal{F}(H)$, for some $H \in \mathbb{R}_0^+$. Given a set of points $\mathcal{X} = \{x_1, \ldots, x_k\} \subset \mathbb{R}^n$ and a set of function values $\mathcal{Y} = \{y_1, \ldots, y_k\} \subset \mathbb{R}^m$, the piecewise-linear interpolant with the smallest* *maximum approximation error is based on the Delaunay triangulation $\mathcal{DT}(\mathcal{X})$, i.e., for any point $x \in \operatorname{conv} \mathcal{X}$, the following is true:*

$$\left\| \psi(x) - \widehat{\psi}^{\mathcal{X},\mathcal{Y}}_{\mathcal{DT}}(x) \right\| = \min_{\mathcal{T}(\mathcal{X})} \max_{\psi \in \mathcal{F}(H)} \left\| \psi(x) - \widehat{\psi}^{\mathcal{X},\mathcal{Y}}_{\mathcal{T}}(x) \right\|.$$

The following proposition uses Lemma 2.4.3 to show that choosing the Delaunay triangulation defines the learned controller that results in closed-loop trajectories that best approximate the corresponding expert trajectories.

Proposition 2.4.4. *Consider the system* (2.5) *and assume it is feedback linearizable on an open set $U \subseteq \mathbb{R}^n$ containing the origin. Let $T \in \mathbb{R}^+$ and suppose we are given a finite set of demonstrations $\mathcal{D} = \{(x^i, u^i)\}_{i=1}^M$ generated by the system* (2.5)*, in closed loop with a smooth asymptotically stabilizing controller $k : \mathbb{R}^n \to \mathbb{R}$, and satisfying $x^i(t) \in U$ for all $t \in [0, T]$. Assume that $\{\Phi(x^i(t))\}_{i=1}^M$ is affinely independent for all $t \in [0, T]$. For any $z_0 \in \operatorname{conv} \mathcal{Z}(0)$, the controller $\widehat{\kappa}_{\mathcal{DT}}$ based on the Delaunay triangulation $\mathcal{DT}(\mathcal{Z}(0))$ defined as in* (2.26) *results in closed-loop trajectories in z-coordinates that have the smallest worst-case trajectory approximation error on the interval $[0, T]$ as defined in Definition 2.4.2.*

Proof. Recall that by Lemma 2.3.2 the trajectory of the system (2.9) under the learned controller (2.26) is given by:

$$\widehat{\phi}_{\mathcal{T}}(t, z_0) = Z_{\mathcal{I}_{\mathcal{T}}(z_0)}(t)\zeta,$$

where $\zeta \in \mathbb{R}^n$ is a vector of affine coefficients that we choose ζ at the beginning of the interval $[0, T]$ and keep constant.

For a fixed $t \in [0, T]$, we can interpret $\widehat{\phi}_{\mathcal{T}}(t, \cdot)$ as a piecewise-linear interpolant of ϕ_t mapping initial conditions to the state reached at time t based on sample points $\mathcal{Z}(0)$ and sample values $\mathcal{Z}(t)$. Therefore, since, for any $t \in [0, T]$, the function $\phi_t \in \mathcal{F}(H)$, by Lemma

2.4.3, the function $\widehat{\phi}_{\mathcal{DT}}(t,\cdot)$ is the best worst-case approximation of the function ϕ_t, i.e.:

$$\begin{aligned}\sup_{\phi_t \in \mathcal{F}(H)} \left\| \phi_t(z_0) - \widehat{\phi}_{\mathcal{DT}}(t, z_0) \right\| \leq \\ \sup_{\phi \in \mathcal{F}(H)} \left\| \phi_t(z_0) - \widehat{\phi}_{\mathcal{T}}(t, z_0) \right\|,\end{aligned} \tag{2.33}$$

for any triangulation $\mathcal{T}(Z(0))$ and any $z_0 \in \operatorname{conv} \mathcal{Z}(0)$. Noting that (2.33) holds for all $t \in [0, T]$, we have:

$$\max_{t\in[0,T]} \sup_{\phi \in \mathcal{F}(H)^{[0,T]}} \left\| \phi(t, z_0) - \widehat{\phi}_{\mathcal{DT}}(t, z_0) \right\| \leq \max_{t\in[0,T]} \sup_{\phi \in \mathcal{F}(H)^{[0,T]}} \left\| \phi(t, z_0) - \widehat{\phi}_{\mathcal{T}}(t, z_0) \right\|,$$

that can be written as:

$$\sup_{\phi \in \mathcal{F}(H)^{[0,T]}} \max_{t\in[0,T]} \left\| \phi(t, z_0) - \widehat{\phi}_{\mathcal{DT}}(t, z_0) \right\| \leq \sup_{\phi \in \mathcal{F}(H)^{[0,T]}} \max_{t\in[0,T]} \left\| \phi(t, z_0) - \widehat{\phi}_{\mathcal{T}}(t, z_0) \right\|.$$

□

Remark 2.4.5. While we justify the construction of the controller for $z(pT) \in \operatorname{conv} \mathcal{Z}(0)$ with optimality in terms of approximation error, we cannot provide a similar justification for $z(pT) \notin \operatorname{conv} \mathcal{Z}(0)$. Therefore, we suggest collecting the expert demonstrations in such a way that the normal region of operation belongs to the convex hull of the demonstrations.

Remark 2.4.6. Note that the metric we use to formulate the error in (2.32) is expressed in z-coordinates instead of the original x-coordinates. While we cannot generally have a guarantee that the best worst-case approximation in the z-coordinates translates to that in the x-coordinates, the metric used in (2.32) and the Euclidean norm metric in the x-coordinates are strongly equivalent on $\operatorname{conv} \mathcal{Z}(t - pT)$ due to the Lipschitz continuity of Φ and its inverse.

Remark 2.4.7. Similarly to Proposition 2.4.4, one can use Lemma 2.4.3 to show that the learned controller $\widehat{\kappa}_{\mathcal{DT}}$ based on the Delaunay approximation $\mathcal{DT}(\mathcal{Z}(0))$ is the best worst-case approximation of the expert controller κ, i.e., for any $z \in \operatorname{conv} \mathcal{Z}(0)$, the Delaunay triangulation is the solution of:

$$\min_{\mathcal{T}(\mathcal{Z}(0))} \max_{\kappa \in \mathcal{F}(H')} \left\| \kappa(z) - \widehat{\kappa}_{\mathcal{T}}(z) \right\|, \tag{2.34}$$

where $H' \in \mathbb{R}$ is a bound on the Hessian norms of coordinate functions of κ.

2.5 Learning a stabilizing controller for non-feedback linearizable systems

In Sections 2.3 and 2.4, we propose a methodology for learning control from expert demonstrations assuming the system is feedback linearizable. Here, we extend our methodology to systems outside of the class of feedback linearizable systems using an embedding technique described in [72].

2.5.1 Embedding technique

First, we describe the embedding technique from [72]. This technique immerses a nonlinear system of dimension n into an extended system that contains a chain of n integrators via dynamic feedback. Although in [72] only single-input single-output systems were considered, it can be shown that a similar technique applies to multiple-input multiple-output systems. For clarity of exposition, however, we will consider a system (2.5) with $m = 1$ and the results extend to multiple-input multiple-output case, mutatis mutandis.

Given constants $w_j \in \mathbb{R}$, $j = 1, \dots, n-1$ and an output map $h : \mathbb{R}^n \to \mathbb{R}$, we define $\Phi : \mathbb{R}^{2n-1} \to \mathbb{R}^{2n-1}$ by:

$$\begin{bmatrix} z \\ \xi \end{bmatrix} = \Phi(x, \xi) = \begin{bmatrix} \Phi_z(x,\xi) \\ \xi \end{bmatrix} = \begin{bmatrix} h(x) + \xi_1 \\ L_f h(x) + \xi_2 \\ L_f^2 h(x) + \xi_3 \\ \vdots \\ L_f^{n-1} h(x) - \sum_{j=1}^{n-1} w_j \xi_j \\ \xi \end{bmatrix}, \tag{2.35}$$

where $\xi \in \mathbb{R}^{n-1}$. We also define the auxiliary dynamics:

$$\begin{aligned}
\dot{\xi}_1 &= \xi_2 - L_g h(x) u, \\
\dot{\xi}_2 &= \xi_3 - L_g L_f h(x) u, \\
&\vdots \\
\dot{\xi}_{n-1} &= -\sum_{i=1}^{n-1} w_i \xi_i - L_g L_f^{n-2} h(x) u,
\end{aligned} \tag{2.36}$$

and the feedback law:

$$u = \frac{1}{r(x)} \left(s(x, \xi) + v \right) \tag{2.37}$$

where:

$$r(x) = L_g L_f^{n-1} h(x) + \sum_{j=1}^{n-1} w_j L_g L_f^{j-1} h(x), \tag{2.38}$$

and:

$$s(x, \xi) = -L_f^n h(x) + \sum_{i=1}^{n-1} w_i w_{n-1} \xi_i - \sum_{j=1}^{n-2} w_j \xi_{j+1}. \tag{2.39}$$

We say that the system (2.5) is feedback linearizable through an embedding on the open set $U \subseteq \mathbb{R}^n$ if there exist constants w_j, $j = 1, \ldots, n-1$ and the output map h such that Φ is a diffeomorphism from U to $\Phi(U)$ and $r(x) \neq 0$ for all $x \in U$.

If the system (2.5) is feedback linearizable through an embedding, we can rewrite the dynamics of (2.5) and (2.36) in the (z, ξ)-coordinates given by (2.35) resulting in the system that consists of the subsystem describing evolution of z given by:

$$\begin{aligned}
\dot{z}_1 &= z_2, \\
&\vdots \\
\dot{z}_{n-1} &= z_n, \\
\dot{z}_n &= -s(x, \xi) + r(x) u,
\end{aligned} \tag{2.40}$$

and the subsystem describing the evolution of ξ given by:

$$\begin{aligned}\dot{\xi}_1 &= (\xi_2 - L_g h(x)u)_{(x,\xi)=\Phi^{-1}(z,\xi)}\\ &\vdots\\ \dot{\xi}_{n-1} &= \left(-\sum_{i=1}^{n-1} w_i\xi_i - L_g L_f^{n-2} h(x)u\right)_{(x,\xi)=\Phi^{-1}(z,\xi)}.\end{aligned} \tag{2.41}$$

Furthermore, when the system (2.5) is feedback linearizable through an embedding, the feedback law given by (2.37) is well-defined and transforms the z-subsystem (2.40) into the chain of integrators given by (2.9) and the ξ-subsystem (2.41) into:

$$\begin{aligned}\dot{\xi}_1 &= \left(\xi_2 - \frac{L_g h(x)}{r(x)}(s(x,\xi)+v)\right)_{(x,\xi)=\Phi^{-1}(z,\xi)}\\ &\vdots\\ \dot{\xi}_{n-1} &= \left(-\sum_{i=1}^{n-1} w_i\xi_i - \frac{L_g L_f^{n-2} h(x)}{r(x)}(s(x,\xi)+v)\right)_{(x,\xi)=\Phi^{-1}(z,\xi)}.\end{aligned} \tag{2.42}$$

2.5.2 Expert demonstrations

Similarly to the case of fully feedback linearizable systems, we assume that the system (2.5) is feedback linearizable through an embedding on an open set $U \subseteq \mathbb{R}^n$ containing the origin and the demonstrations $x^i(t)$ belong to U for all $t \in [0,T]$. We first transform the demonstrations $\mathcal{D}$ into (z,ξ,v)-coordinates. For each demonstration (x^i,u^i), we use (2.35), (2.36), and (2.37) to transform (x^i,u^i) into (z^i,ξ^i,v^i). More specifically:

- we choose an arbitrary $\xi_0 \in \mathbb{R}^{n-1}$ to initialize $\xi^i(0) = \xi_0$, and solve the equation in (2.36) using demonstrations (x^i,u^i) as the input to determine $\xi^i(t)$ for $t \in [0,T]$;
- for all $t \in [0,T]$, using $\xi^i(t)$, we determine $z^i(t)$ from (2.35) and $v^i(t)$ from (2.37) as:

$$z^i(t) = \Phi(x^i(t), \xi^i(t)) \tag{2.43}$$

$$v^i(t) = r(x^i(t))u^i(t) - s(x^i(t),\xi^i(t)). \tag{2.44}$$

We denote the resulting set of demonstrations by:

$$\mathcal{D}_{(z,\xi,v)} = \{(z^1,\xi^1,v^1),\dots,(z^n,\xi^n,v^n)\}, \tag{2.45}$$

where $z^i : [0,T] \to \mathbb{R}^n$, $\xi^i : [0,T] \to \mathbb{R}^{n-1}$, and $v^i : [0,T] \to \mathbb{R}$.

2.5.3 Constructing the learned controller for the extended class of systems

We now show that, for $M = n+1$, the controller $v = \widehat{\kappa}(t,z)$ from (2.15) stabilizes the system (2.5) by stabilizing the chain of integrators (2.9) in the transformed coordinates (z,ξ). Please note that we focus on the case $M = n+1$ for ease of exposition, and the proposed extension is also compatible with the case $M \geq n+1$ described in Section 2.4.

The following statement provides sufficient conditions for stability of (2.5) under the control law (2.15).

Theorem 2.5.1. *Consider the system* (2.5) *and assume it is feedback linearizable through an embedding on an open set $U \subseteq \mathbb{R}^n$ containing the origin. Let $T \in \mathbb{R}^+$ and suppose we are given a finite set of demonstrations $\mathcal{D} = \{(x^i,u^i)\}_{i=1}^{n+1}$ of the system* (2.5) *generated by the system* (2.5)*, in closed loop with a smooth asymptotically stabilizing controller $k : \mathbb{R}^n \to \mathbb{R}$, and satisfying $x^i(t) \in U$ for all $t \in [0,T]$. Further, suppose the following two conditions hold:*

(A_1) the matrix:

$$A_\xi = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \\ -w_1 & -w_2 & \cdots & -w_{n-1} \end{bmatrix} \tag{2.46}$$

is Hurwitz;

(A_2) the ξ-subsystem in (2.42) *is input-to-state stable (ISS) with respect to z and v.*

Assume that the set $\{\Phi_z(x^1(t)), \ldots, \Phi_z(x^n(t))\}$ *is affinely independent for all* $t \in [0, T]$*. Then, there exists a* $\tilde{T} \in \mathbb{R}^+$ *such that for all* $T \geq \tilde{T}$*, the origin of system* (2.5) *in closed-loop with the auxiliary dynamics* (2.36) *and controller* (2.37)-(2.15) *is uniformly asymptotically stable.*

Proof. Let us use condition (A_1) to show that the expert solutions $\{(x^i, u^i)\}_{i=1}^{n+1}$ are uniformly asymptotically stable. Since $k(x)$ is asymptotically stabilizing, the origin of the system $\dot{x} = f(x) + g(x)k(x)$ is uniformly asymptotically stable. Because the origin is the equilibrium point of $\dot{x} = f(x) + g(x)k(x)$, we have that $k(0) = 0$. The ξ-subsystem in (2.36) with $u = k(x)$ can be interpreted as a control system with the input x. Condition (A_1) together with the fact that $k(0) = 0$ implies that the ξ-subsystem in (2.36) with $u = k(x)$ is uniformly exponentially stable when $x \equiv 0$. Uniform exponential stability of the unforced ξ-subsystem implies that the ξ-subsystem in (2.36) with $u = k(x)$ is ISS with respect to x (see Lemma 4.6 in [77]). By Lemma 4.7 in [77], input-to-state stability of the ξ-subsystem with $u = k(x)$ with respect to x as input and uniform asymptotic stability of $\dot{x} = f(x) + g(x)k(x)$ implies that there is a class $\mathcal{KL}$ function $\beta : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}_0^+$ such that for all $i \in \{1, \ldots, n\}$:

$$\left\| \begin{bmatrix} x^i(t) \\ \xi^i(t) \end{bmatrix} \right\| \leq \beta \left(\left\| \begin{bmatrix} x^i(0) \\ \xi^i(0) \end{bmatrix} \right\|, t \right). \tag{2.47}$$

Because the system (2.5) is feedback linearizable through an embedding, we have that Φ given by (2.35) is a diffeomorphism. Therefore, according to [81], the inequality (2.47) implies that for all $i \in \{1, \ldots, n\}$:

$$\left\| \begin{bmatrix} z^i(t) \\ \xi^i(t) \end{bmatrix} \right\| \leq \beta_1 \left(\left\| \begin{bmatrix} z^i(0) \\ \xi^i(0) \end{bmatrix} \right\|, t \right), \tag{2.48}$$

where $\beta_1 : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}_0^+$ is a class $\mathcal{KL}$ function.

We can use (2.48) to show that there is a $\tilde{T} \in \mathbb{R}_0^+$ such that:

$$\beta_1 \left(\left\| \begin{bmatrix} z^i(0) \\ \xi^i(0) \end{bmatrix} \right\|, T \right) < \frac{1}{2\sqrt{n}\|Z^{-1}(0)\|},$$

for all $T \geq \tilde{T}$. Using the argument from the proof of Theorem 2.3.4 allows us to conclude uniform exponential stability of the origin of the z-subsystem given by (2.9), provided $T > \tilde{T}$.

The uniform exponential stability of the z-subsystem implies that there is a function $\beta_z \in \mathcal{KL}$ such that:

$$\|z(t)\| \leq \beta_z(\|z(0)\|, t). \tag{2.49}$$

The matrix product $V(t)Z^{-1}(t)$ is continuous with respect to t and defined on $[0, T]$ that is compact. By the extreme value theorem, this product has a bounded norm. This fact, together with the inequality (2.49), implies that the control input $v\widehat{\kappa}(t, z)$ given by (2.15) satisfies:

$$\|v(t)\| \leq \beta_v(\|z(0)\|, pT), \tag{2.50}$$

where β_v is also a class $\mathcal{KL}$ function.

By condition (A_2), the ξ-subsystem in (2.42) is ISS with respect to z and v. Lemma 4.7 in [77] shows that the ISS property, along with the bounds (2.49) and (2.50), allows us to conclude that the origin of the system (2.9)-(2.42) in closed-loop with the controller (2.15) is uniformly asymptotically stable. Uniform asymptotic stability of the origin of the system (2.9)-(2.42)-(2.15) in the (z, ξ)-coordinates implies uniform asymptotic stability of the origin of the feedback equivalent system (2.5)-(2.36)-(2.37)-(2.15) [81]. □

Remark 2.5.2. Similarly to Theorems 2.3.4 and 2.4.1, in Theorem 2.5.1, we assume feedback linearizability through an embedding on some open set $U \subseteq \mathbb{R}^n$ without explicitly specifying under what conditions this occurs. This is done to give the user an opportunity to use either local or global results, depending on what their application allows for. To show local feedback linearizability through an embedding, we suggest using the conditions from Proposition 4 in [72], namely that there exist constants w_j, $j = 1, \ldots, n-1$ and an output map h such that:

(B_1) the matrix

$$\mathcal{O}(x) = \begin{bmatrix} dh(x)^T & dL_f h(x)^T & \dots & dL_f^{n-1} h(x)^T \end{bmatrix}^T,$$

has rank n at the origin, implying that Φ given by (2.35) is a diffeomorphism from some neighborhood U_1 of the origin to $\Phi(U_1)$;

(B_2) $r(0) \neq 0$, which implies that $r(x) \neq 0$ for some neighborhood U_2 of the origin.

These conditions imply that the system (2.5) is feedback linearizable through an embedding on an open set $U = U_1 \cap U_2$. Please note that the condition (B_1) is also the sufficient condition for local observability at the origin. The condition (B_2) is violated if and only if $L_g L_f^{j-1} h(0) = 0$ for all $j = 1, \dots, n-1$, which is equivalent to $\mathcal{O}(0) \cdot g(0) = 0$. Given that $\mathcal{O}(0)$ is full-rank, this condition is, in turn, a consequence of (B_1), provided $g(0) \neq 0$.

Remark 2.5.3. Note that the class of feedback linearizable systems through an embedding strictly contains feedback linearizable systems. In Section 2.6.2, we provide an example of a system belonging to this class which is not feedback linearizable.

Remark 2.5.4. In general, verifying condition (A_2) can be a challenging task. Therefore, the authors of [72] suggest substituting the ISS condition (A_2) with the more verifiable condition that the matrix $A_\xi + A_w$ is Hurwitz, where:

$$A_w = \nabla_\xi \begin{bmatrix} L_g h(x) \frac{s(x,\xi)}{r(x)} \Big|_{(x,\xi)=\Phi^{-1}(z,\xi)} \\ L_g L_f h(x) \frac{s(x,\xi)}{r(x)} \Big|_{(x,\xi)=\Phi^{-1}(z,\xi)} \\ \vdots \\ L_g L_f^{n-2} h(x) \frac{s(x,\xi)}{r(x)} \Big|_{(x,\xi)=\Phi^{-1}(z,\xi)} \end{bmatrix}_{(z,\xi)=(0,0)}. \tag{2.51}$$

This is because the matrix $A_\xi + A_w$ is a linear approximation of the unforced ξ-subsystem in (2.42) around the origin and its stability implies local ISS of the system (2.42).

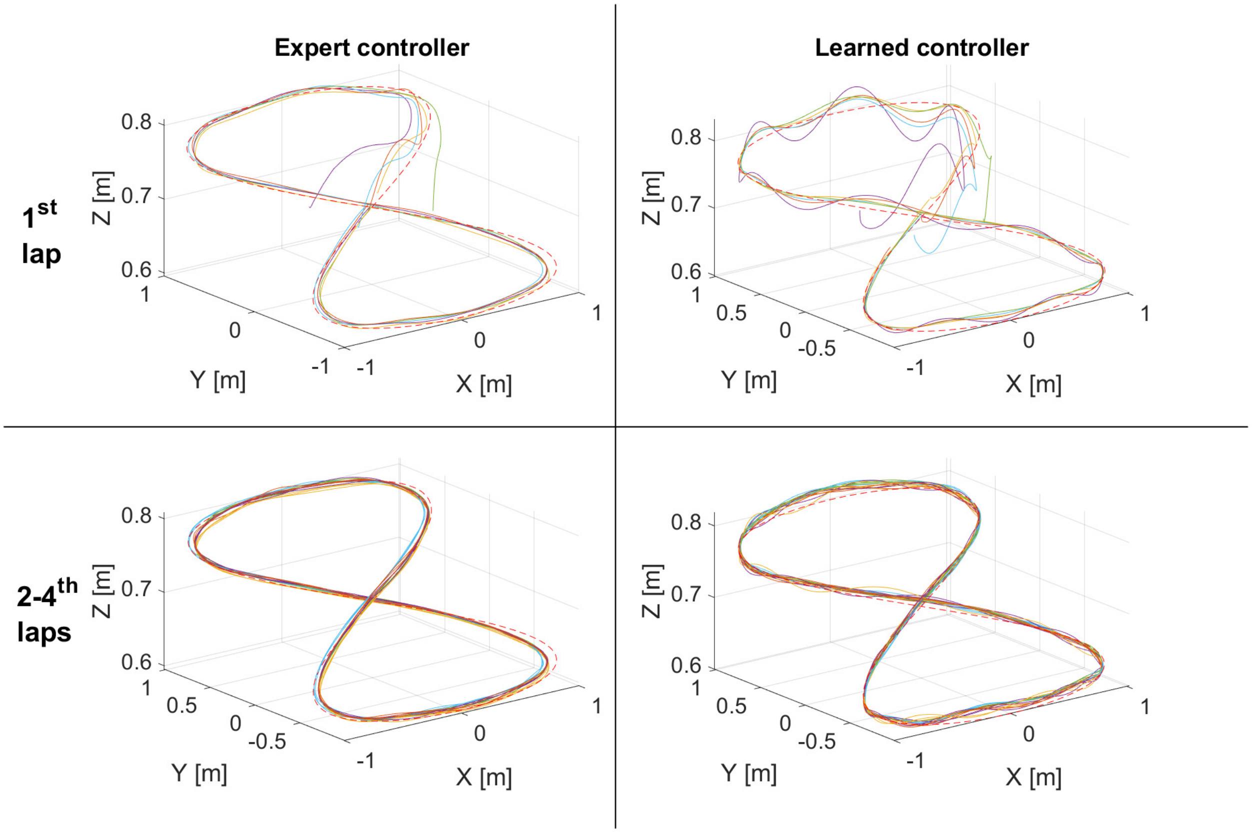

Figure 2.1: Trajectory tracking of the nonlinear controller from [7] (left column) and the learned controller from (2.15) (right column) under five different initial conditions. Each experiment is plotted with a different color. The first lap trajectories (top row) are plotted separately from those in the subsequent laps (bottom row).

2.6 Experiments and simulations

2.6.1 Quadrotor control experiment

We illustrate the performance of our methodology using the example of quadrotor dynamics:

$$\ddot{p} = \frac{1}{m}\left(\tau R e_3 - [\omega]_\times J\omega\right), \tag{2.52}$$

$$\dot{R} = R[\omega]_\times, \tag{2.53}$$

$$\dot{\omega} = J^{-1}(\eta - [\omega]_\times J\omega), \tag{2.54}$$

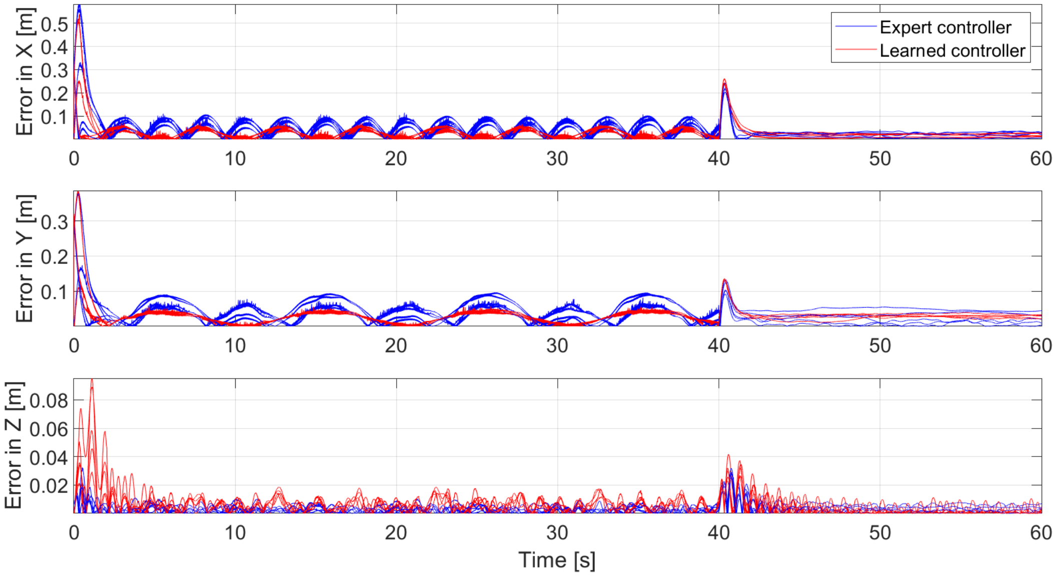

Figure 2.2: Comparison of tracking errors in X, Y and Z coordinates of learned controller from (2.15) (red) and nonlinear controller from [7] (blue) for all five experiments.

where: $p \in \mathbb{R}^3$, $R \in SO(3)$, $\omega \in \mathbb{R}^3$ are the position, orientation, and angular velocity of the quadrotor, respectively; $\tau \in \mathbb{R}$ and $\eta \in \mathbb{R}^3$ are thrust and torque inputs, respectively; $m \in \mathbb{R}$, and $J \in \mathbb{R}^{3\times3}$ are the mass and the inertia matrix; $[\,\cdot\,]_\times$ denotes the matrix form of the vector cross product, and $e_3 = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$ is a unit vector.

We split the dynamics (2.52)-(2.54) into two subsystems: one described by (2.52)-(2.53) with the state $x = (p, \dot{p}, R, \tau)$ and the virtual inputs $u = (\dot{\tau}, \omega)$, and the other described by (2.54) with the state $x' = \omega$ and the virtual inputs $u' = \eta$. Typically, quadrotors have high-frequency internal controllers that track the desired angular velocity based on state feedback and, therefore, it is reasonable to assume that we can directly control the angular velocity [89].

It is known that the dynamics (2.52)-(2.54) are differentially flat with respect to position and yaw angle [90]. In what follows, we focus on controlling the position p, whereas the yaw angle is controlled to remain constant. Differential flatness allows us to transform the dynamics (2.52)-(2.53) into linear dynamics $\dot{z}_1 = z_2,\ \dot{z}_2 = z_3,\ \dot{z}_3 = v$ via a coordinate transformation:

$$z = \begin{bmatrix} z_1 & z_2 & z_3 \end{bmatrix}^T \triangleq \begin{bmatrix} p & \dot{p} & \ddot{p} \end{bmatrix}^T,$$

and the feedback law:

$$v = \frac{1}{m}(\dot{\tau} R e_3 - \tau R \omega_1 e_2 + \tau R \omega_2 e_1) \triangleq b(z)u.$$

We apply the controller design[1] from [7] to the dynamics (2.52)-(2.54) in simulation[2] and use the resulting solutions as the expert demonstrations. The controller parameters are chosen as follows: $K_P = \mathrm{diag}(7.0, 7.0, 16.5)$, $K_I = \mathrm{diag}(0, 0, 15.5)$, $K_D = \mathrm{diag}(5.0, 5.0, 3.4)$, $K_{rp} = 6.0$, $K_y = 2.0$. The expert is commanded to stabilize the quadrotor at the origin, starting from various positions, velocities and accelerations. Given the dimension of the state z equals 9, we record 10 expert solutions $\{(z^i, v^i)\}_{i=1}^{10}$ from simulations[3], including the pair corresponding to the trivial solution $(z^1, v^1) \equiv (0, 0)$. Please note that the pairs (z^i, v^i) in this context are merely evolutions of position, velocity, acceleration, and jerk. The recorded data is studied to ensure that the sufficient conditions of Theorem 2.3.4 are satisfied, i.e., the matrix $Z(t)$ in (2.12) is always invertible and $\|Z(T)Z^{-1}(0)\| < 1$, and a fragment of length $T = 2$ s is used to construct a stabilizing controller (2.15).

Next, we compare the learned controller (2.15) and the expert controller from [7] by using them to control a BitCraze CrazyFlie 2.0 quadrotor. In these experiments, the control inputs (τ, ω) are supplied by a computer via a USB radio at the average rate of 300 Hz. The internal PD controller of the CrazyFlie tracks (τ, ω) by controlling angular speeds of

[1] The only difference of the expert controller used in this work and that used in [7] is that here the low-level controller is a linear PD controller.

[2] We collected expert demonstrations in simulation to ensure there is no estimation error affecting the controller construction. In future work, we aim to construct the controller from expert solutions given by an observer.

[3] The initial conditions used are the unit vectors of $\mathbb{R}^9$.

individual rotors. For state estimation, we use a Kalman filter that gets the position and attitude measurements from an OptiTrack motion capture system.

The experimental benchmark[4] we choose to compare the controllers is to track the reference depicted on Figure 2.1, which consists of two parts: a figure of eight given by:

$$p_R(t) = \left(\sin(4\pi f t), \sin(2\pi f t), 0.1 \sin(2\pi f t) + 0.7\right),$$

where $f = 0.1\,\mathrm{Hz}$, from $t = 0\,\mathrm{s}$ to $t = 40\,\mathrm{s}$; and a setpoint at the origin after $t \geq 40\,\mathrm{s}$. Note the reference trajectory is quite different from the collected expert demonstrations. We use the learned controller $\widehat{\kappa}$ from (2.15) to control the tracking error with $v(t) = \widehat{\kappa}(t, z(t) - z_R(t))$, where $z_R = (p_R, \dot{p}_R, \ddot{p}_R)$, together with the feedback law:

$$u(t) = (\dddot{p}_R(t) + v(t))/b(z(t)).$$

For both the expert and the learned controllers, we perform five experiments — each from a different initial position[5].

In Figure 2.1, we depict the quadrotor trajectories for both the nonlinear controller in [7] and the learned controller (2.15) tracking the aforementioned trajectory. We plot the position trajectories in the first lap separately from those in the subsequent laps to decouple the transient behaviour of a controller from the steady-state behaviour. In Figure 2.2 we compare the tracking errors of the learned controller with those of the nonlinear controller from [7] for all five experiments. The initial conditions used during the experiments are purposefully chosen to be far from the initial conditions used during the simulation. The learned controller appears to track the trajectory well — the error is of the order of centimeters. It can be seen qualitatively, however, from Figure 2.1 that, in comparison to the expert controller, the learned controller takes a longer time to settle — this is especially noticeable in the experiments where the initial position of the quadrotor does not match that

[4] Code used in the experiments can be found at `https://github.com/cyphylab/cyphy_testbed/tree/LFD`.

[5] The initial positions used are $(0, 0, 0.7)$, $(0.3, 0.3, 0.7)$, $(0.3, -0.3, 0.7)$, $(-0.3, 0.3, 0.7)$, $(-0.3, -0.3, 0.7)$.

of the reference. From Figure 2.2, we observe that the errors of the learned controller and the expert controller are comparable, with the errors of the learned controller being slightly smaller in X and Y coordinates, whereas being slightly larger in Z coordinates. For $t \geq 40$ s, the error in position does not tend to zero for neither of the controllers, which appears to contradict the theoretical results. We attribute this to the several milliseconds of delay with which the control input is sent to the quadrotor[6].

2.6.2 Ball and beam control simulation

Consider the ball and beam model described by [91]:

$$\begin{aligned} \ddot{r} &= \bar{b}(r\omega^2 - \bar{g}\sin(\phi)) \\ \dot{\phi} &= \omega \\ \dot{\omega} &= u, \end{aligned} \tag{2.55}$$

with $r \in \mathbb{R}$ and $v \in \mathbb{R}$ denoting the position and the velocity of the ball on the beam, respectively, while $\phi \in \mathbb{R}$ and $\omega \in \mathbb{R}$ denote the angle and the angular velocity of the beam with respect to the horizontal line, respectively. The constant $\bar{g}$ is the gravity constant, and the constant $\bar{b} = m/(J_b/R^2 + m)$, where m is mass, J_b is the moment of inertia, and R is the radius of the ball. The state of the system (2.55) is given by $x = (r, \dot{r}, \phi, \omega)$. The values of the parameters in this simulation are chosen to be $\bar{b} = 0.7143$ and $\bar{g} = 9.81$.

We choose the stabilizing controller[7] proposed in [8] as the expert controller for the system (2.55). Since dimension of the state is 4, we record simulations of 5 expert solutions $\{(x^i, u^i)\}_{i=1}^5$ starting from various initial conditions[8], including the trivial solution $(x^1, u^1) \equiv (0, 0)$.

[6] Even in simulation, an introduction of such a delay into the control loop has resulted in the trajectory stabilizing at a non-zero steady-state error.

[7] This controller is interesting to study because it is nonlinear, contains nested saturations, and utilizes backstepping.

[8] The initial conditions used are $(1, 0, 0, 0)$, $(0, 1, 0, 0)$, $(0, 0, \pi/8, 0)$, $(0, 0, 0, 10)$.

It can be shown that system (2.55) is not feedback linearizable [91] and, therefore, techniques described in Sections 2.3 and 2.4 cannot be used. Instead, we use the technique described in Section 2.5 to approximate the expert controller. To immerse the system (2.55), we use the map Φ given by (2.35) with:

$$\Phi_z(x,\xi) = \begin{bmatrix} x_1 + \xi_1 \\ x_2 + \xi_2 \\ -\bar{b}\bar{g}\sin x_3 + \bar{b}x_1 x_4^2 + \xi_3 \\ \bar{b}x_2 x_4^2 - \bar{g}x_4\cos x_3 - \sum_{i=1}^{3} w_i\xi_i \end{bmatrix}, \tag{2.56}$$

and the dynamic control law:

$$\begin{cases} u = \frac{1}{r(x)}\left(s(x,\xi) + v\right) \\ \dot{\xi}_i = \xi_{i+1}, \quad i = 1,2 \\ \dot{\xi}_3 = -w_1\xi_1 - w_2\xi_2 - w_3\xi_3 - 2x_1x_4u, \end{cases} \tag{2.57}$$

where:

$$\begin{aligned} r(x) &= 2\bar{b}x_2x_4 - \bar{b}\bar{g}\cos x_3 + 2w_3bx_1x_4 \\ s(x,\xi) &= -\bar{b}^2x_4^2\left(-\bar{g}\sin x_3 + x_1x_4^2\right) - \bar{b}\bar{g}x_4^2\sin x_3 + w_1\xi_2 \\ &\quad + w_2\xi_3 - w_1w_3\xi_1 - w_2w_3\xi_2 - w_3^2\xi_3. \end{aligned}$$

Please note that this map and dynamic control law are only well-defined on an open set around the origin.

We choose the parameters $w = (1,3,3)$. First, we solve the differential equation in (2.57) for the initial state $\xi(0) = 0$ using each of the previously collected expert solutions $\{(x^i,u^i)\}_{i=1}^5$ as inputs. Next, we use the transformations (2.43) and (2.44) to transform the expert solutions into the form $\{(z^i,v^i)\}_{i=1}^5$. Then, these solutions are inspected to ensure that the conditions of Theorem 2.5.1 are satisfied, i.e., the matrix $Z(t)$ is always invertible and $\|Z(T)Z^{-1}(0)\| < 1$, and a fragment of length $T = 8$s is used to construct a stabilizing controller (2.15).

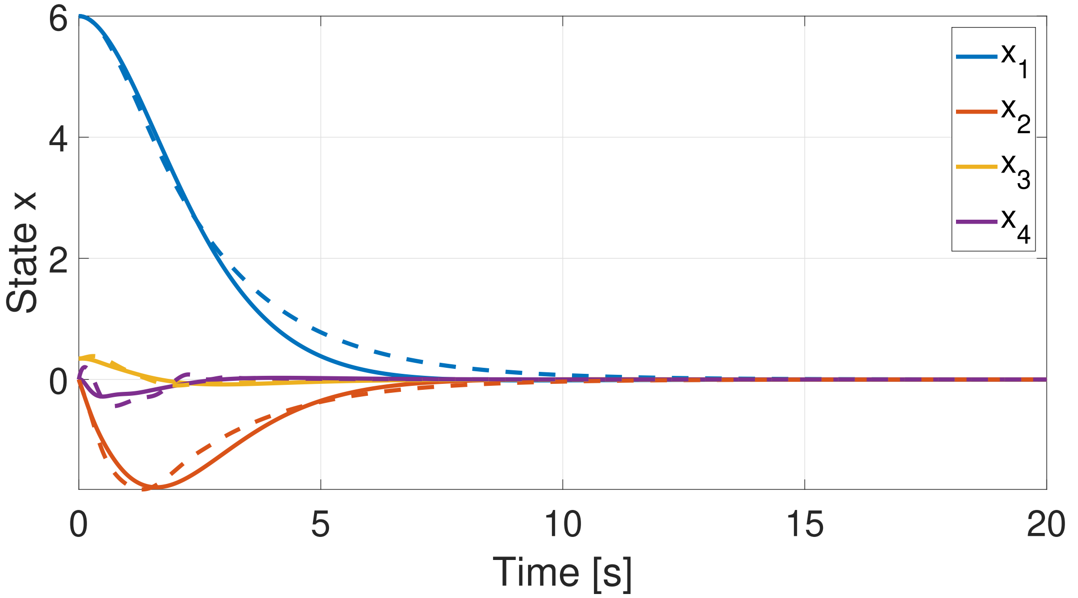

Figure 2.3: Comparison of stabilization between the learned controller from (2.15) (solid lines) and nonlinear controller from [8] (dashed lines).

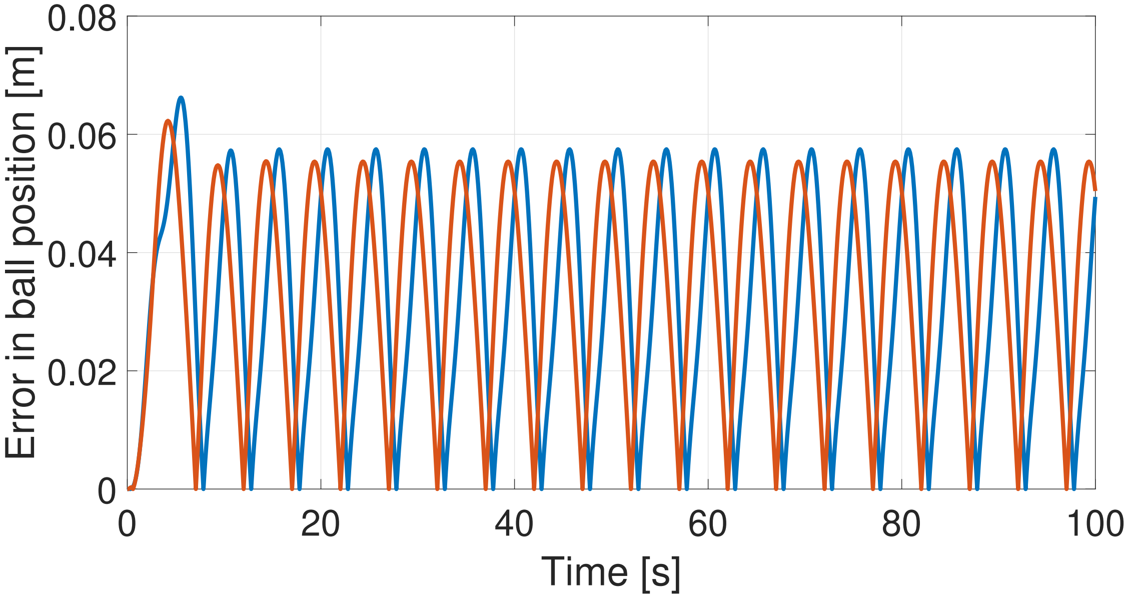

Figure 2.4: Comparison of tracking errors between the learned controller from (2.15) (blue) and nonlinear controller from [8] (red).

We compare the learned controller (2.15) and the expert controller from [8] based on their performance on two tasks: stabilization and output tracking. We define the position of the ball to be the output, i.e., $y = x_1$. When using the expert controller from [8] for output tracking, we use the output regulation method described in [92]. On Figure 2.3, we compare how both controllers stabilize the system to the origin from the initial state $x(0) = (6, 0, 0.345, 0)$. The initial conditions used when deploying the learned controller are purposefully chosen to be far from the initial conditions used during the expert's deployment. The closed-loop solutions of the learned controller and the expert controller appear to be very similar, although the learned controller stabilizes to the origin slightly slower than the expert. On Figure 2.4, we compare how both controllers follow a reference trajectory given by $y_R(t) = 6 \cdot \cos \frac{2\pi}{10} t$ from the initial state $x(0) = (6, 0, 0.345, 0)$ by plotting their tracking errors. We again observe that the performance of the learned controller and that of the expert controller are similar, with the learned controller having a slightly larger error.

2.7 Conclusions and future work

In this chapter, we proposed a methodology for constructing a controller for a known nonlinear system from a finite set of expert demonstrations of desired behaviour. Unlike many works in the literature on LfD, we provide formal guarantees of asymptotic stability of the closed-loop system. Furthermore, we discuss what choice of demonstrations results in the best worst-case approximation of the expert controller, given there are more than $n + 1$ demonstrations. Finally, we verify the methodology by applying it to control the quadrotor dynamics, which is an example of a feedback linearizable system, and the ball-and-beam dynamics, which is an example of a system that is feedback linearizable through an embedding.

The work presented in this chapter immediately leads to the following question: "How to learn a stabilizing controller from expert demonstrations when the system dynamics are unknown?" This question was addressed for single-input single-output feedback linearizable systems in [74], but remains an open question for a wider class of systems. The fundamental property that was a theoretical basis of this work was that an affine combination of solutions of a linear system is also its solution. It would be interesting to study whether there exists a similar property for some class of nonlinear systems, i.e., one can express any solution of a nonlinear system based on a finite set of its solutions. This would allow one to bypass feedback linearization step and, if this class of systems is distinct from the class feedback linearizable systems, apply a similar methodology to a wider range of systems.

REFERENCES

[1] A. Vick, J. Guhl, and J. Kruger, “Model predictive control as a service - Concept and architecture for use in cloud-based robot control,” in *Proceedings of the 21st International Conference on Methods and Models in Automation and Robotics (MMAR)*, Aug. 2016, pp. 607–612.

[2] T. Hegazy and M. Hefeeda, “Industrial Automation as a Cloud Service,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 10, pp. 2750–2763, Oct. 2015.

[3] B. Hoh, T. Iwuchukwu, Q. Jacobson, D. Work, A. M. Bayen, R. Herring, J. C. Herrera, M. Gruteser, M. Annavaram, and J. Ban, “Enhancing privacy and accuracy in probe vehicle-based traffic monitoring via virtual trip lines,” *IEEE Transactions on Mobile Computing*, vol. 11, no. 5, pp. 849–864, May 2012.

[4] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, “A Survey of Robot Learning from Demonstration,” *Robot. Auton. Syst.*, vol. 57, no. 5, p. 469–483, May 2009.

[5] A. G. Billard, S. Calinon, and R. Dillmann, *Learning from Humans.* Cham, Switzerland: Springer International Publishing, 2016, pp. 1995–2014.

[6] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, “Recent Advances in Robot Learning from Demonstration,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 297–330, 2020.

[7] M. Faessler, F. Fontana, C. Forster, and D. Scaramuzza, “Automatic re-initialization and failure recovery for aggressive flight with a monocular vision-based quadrotor,” in *Proceedings of the 32nd IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1722–1729.

[8] C. Barbu, R. Sepulchre, W. Lin, and P. Kokotovic, “Global asymptotic stabilization of the ball-and-beam system,” in *Proceedings of the 36th IEEE Conference on Decision and Control*, vol. 3, 1997, pp. 2351–2355 vol.3.

[9] Y. Lin, F. Farokhi, I. Shames, and D. Nezic, “Secure Control of Nonlinear Systems Using Semi-Homomorphic Encryption,” in *Proceedings of the 57th IEEE Conference on Decision and Control*, 2018, pp. 5002–5007.

[10] D. Q. Mayne, “Model predictive control: Recent developments and future promise,” *Automatica*, vol. 50, pp. 2967–2986, 2014.

[11] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, “Comprehensive Experimental Analyses of Automotive Attack Surfaces,” in *Proceedings of the 20th USENIX Conference on Security*, ser. SEC’11, 2011, pp. 6–6.

[12] D. Gollmann, P. Gurikov, A. Isakov, M. Krotofil, J. Larsen, and A. Winnicki, "Cyber-Physical Systems Security: Experimental Analysis of a Vinyl Acetate Monomer Plant," in *Proceedings of the 1st ACM Workshop on Cyber-Physical System Security*, 2015, pp. 1–12.

[13] F. M. Cleveland, "Cyber security issues for Advanced Metering Infrasttructure (AMI)," in *Proceedings of the 6th IEEE Power and Energy Society General Meeting*, July 2008, pp. 1–5.

[14] B. Ghena, W. Beyer, A. Hillaker, J. Pevarnek, and J. A. Halderman, "Green Lights Forever: Analyzing the Security of Traffic Infrastructure," in *Proceedings of the 8th USENIX Conference on Offensive Technologies*, 2014, pp. 7–7.

[15] A. Teixeira, K. C. Sou, H. Sandberg, and K. H. Johansson, "Secure control systems: A quantitative risk management approach," *IEEE Control Systems*, vol. 35, no. 1, pp. 24–45, 2015.

[16] Y. Mo and B. Sinopoli, "Secure control against replay attacks," in *Proceedings of the 47th Annual Allerton Conference on Communication, Control, and Computing*, Oct 2009, pp. 911–918.

[17] F. Farokhi, I. Shames, and N. Batterham, "Secure and private control using semi-homomorphic encryption," *Control Engineering Practice*, vol. 67, pp. 13–20, Oct 2017.

[18] H. Sandberg, S. Amin, and K. H. Johansson, "Cyberphysical security in networked control systems: An introduction to the issue," *IEEE Control Systems*, vol. 35, no. 1, pp. 20–23, Feb 2015.

[19] M. Van Dijk and A. Juels, "On the impossibility of cryptography alone for privacy-preserving cloud computing," in *Proceedings of the 5th USENIX conference on Hot topics in security*, July 2010, pp. 1–8.

[20] A. B. Alexandru, M. Morari, and G. J. Pappas, "Cloud-Based MPC with Encrypted Data," in *Proceedings of the 36th Conference on Decision and Control (CDC)*, 2018, pp. 5014–5019.

[21] F. Armknecht, C. Boyd, C. Carr, K. Gjosteen, A. Jaeschke, C. A. Reuter, and M. Strand, "A Guide to Fully Homomorphic Encryption," *IACR Cryptology ePrint Archive*, vol. 2015, p. 1192, 2015.

[22] K. Kogiso and T. Fujita, "Cyber-security enhancement of networked control systems using homomorphic encryption," in *Proceedings of the 54th IEEE Conference on Decision and Control (CDC)*, 2015, pp. 6836–6843.

[23] T. Fujita, K. Kogiso, K. Sawada, and S. Shin, "Security enhancements of networked control systems using rsa public-key cryptosystem," in *Proceedings of the 10th Asian Control Conference (ASCC)*, May 2015, pp. 1–6.

[24] J. Kim, C. Lee, H. Shim, J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Encrypting controller using fully homomorphic encryption for security of cyber-physical systems," *IFAC-PapersOnLine*, vol. 49, no. 22, pp. 175 – 180, 2016, 6th IFAC Workshop on Distributed Estimation and Control in Networked Systems.

[25] F. Farokhi, I. Shames, and N. Batterham, "Secure and private cloud-based control using semi-homomorphic encryption," *IFAC-PapersOnLine*, vol. 49, no. 22, pp. 163–168, 2016, Proceedings of the 6th IFAC Workshop on Distributed Estimation and Control in Networked Systems.

[26] Y. Shoukry, K. Gatsis, A. Alanwar, G. J. Pappas, S. A. Seshia, M. Srivastava, and P. Tabuada, "Privacy-aware quadratic optimization using partially homomorphic encryption," in *Proceedings of the 55th IEEE Conference on Decision and Control (CDC)*, Dec 2016, pp. 5053–5058.

[27] A. B. Alexandru, K. Gatsis, Y. Shoukry, S. A. Seshia, P. Tabuada, and G. J. Pappas, "Cloud-based quadratic optimization with partially homomorphic encryption," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2357–2364, 2021.

[28] K. Kogiso, R. Baba, and M. Kusaka, "Development and Examination of Encrypted Control Systems," in *Proceedings of the 21st IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, July 2018, pp. 1338–1343.

[29] J. Cortes, G. E. Dullerud, S. Han, J. L. Ny, S. Mitra, and G. J. Pappas, "Differential privacy in control and network systems," in *Proceedings of the 55th IEEE Conference on Decision and Control*, Dec 2016, pp. 4252–4272.

[30] F. Koufogiannis and G. J. Pappas, "Differential privacy for dynamical sensitive data," in *Proceedings of the IEEE 56th IEEE Conference on Decision and Control (CDC)*, Dec 2017, pp. 1118–1125.

[31] C. Dwork, "Differential privacy," in *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006)*, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds. Berlin, Heidelberg: Springer, 2006, pp. 1–12.

[32] O. L. Mangasarian, "Privacy-preserving linear programming," *Opt. Letters*, vol. 5, no. 1, pp. 165–172, Feb 2011.

[33] C. Wang, K. Ren, and J. Wang, "Secure and practical outsourcing of linear programming in cloud computing," in *Proceedings of the 30th IEEE INFOCOM*, April 2011, pp. 820–828.

[34] P. C. Weeraddana, G. Athanasiou, C. Fischione, and J. S. Baras, "Per-se Privacy Preserving Solution Methods Based on Optimization," in *Proceedings of the 52nd IEEE Conference on Decision and Control (CDC)*, Dec 2013, pp. 206–211.

[35] P. Weeraddana and C. Fischione, "On the Privacy of Optimization," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 9502 – 9508, 2017, Proceedings of the 20th IFAC World Congress.

[36] Z. Xu and Q. Zhu, "Secure and Resilient Control Design for Cloud Enabled Networked Control Systems," in *Proceedings of the 1st ACM Workshop on Cyber-Physical Systems-Security and/or Privacy*, 2015, pp. 31–42.

[37] D. Wu, B. C. Lesieutre, P. Ramanathan, and B. Kakunoori, "Preserving privacy of AC optimal power flow models in multi-party electric grids," *IEEE Transactions on Smart Grid*, vol. 7, no. 4, pp. 2050–2060, July 2016.

[38] A. Sultangazin and P. Tabuada, "Towards the use of symmetries to ensure privacy in control over the cloud," in *Proceedings of the 57th IEEE Conference on Decision and Control (CDC)*, Dec 2018, pp. 5008–5013.

[39] A. Sultangazin, S. Diggavi, and P. Tabuada, "Protecting the privacy of networked multi-agent systems controlled over the cloud," in *Proceedings of the 27th International Conference on Computer Communication and Networks (ICCCN)*, July 2018, pp. 1–7.

[40] A. Sultangazin and P. Tabuada, "Symmetries and privacy in control over the cloud: uncertainty sets and side knowledge," in *Proceedings of the 58th IEEE Conference on Decision and Control (CDC)*, 2019, pp. 7209–7214.

[41] ——, "Symmetries and isomorphisms for privacy in control over the cloud," *IEEE Transactions on Automatic Control*, vol. 66, no. 2, pp. 538–549, Feb. 2021.

[42] J. M. Lee, *Introduction to Smooth Manifolds*, ser. Graduate Texts in Mathematics. New York, USA: Springer-Verlag, 2003.

[43] B. Vandereycken, P. A. Absil, and S. Vandewalle, "Embedded geometry of the set of symmetric positive semidefinite matrices of fixed rank," in *Proceedings of the 15th IEEE/SP Workshop on Statistical Signal Processing*, Aug 2009, pp. 389–392.

[44] W. Respondek, "Symmetries and Minimal Flat Outputs of Nonlinear Control Systems," in *New Trends in Nonlinear Dynamics and Control and their Applications*, W. Kang, C. Borges, and M. Xiao, Eds. Berlin, Heidelberg: Springer, 2003, pp. 65–86.

[45] M. A. Beitia, J. M. Gracia, and I. de Hoyos, "A linear matrix equation: a criterion for block similarity," *Linear and Multilinear Algebra*, vol. 31, pp. 93–118, 1992.

[46] Panos J. Antsaklis and Anthony N. Michel, *Linear Systems*, 1st ed. Boston, USA: Birkhäuser, 2006.

[47] A. J. Laub, *Matrix Analysis For Scientists And Engineers*. Philadelphia, USA: Society for Industrial and Applied Mathematics, 2004.

[48] F. Sabatino, "Quadrotor control: modeling, nonlinear control design, and simulation," Master's thesis, Department of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden, Jun. 2015.

[49] D. Vogt, S. Stepputtis, S. Grehl, B. Jung, and H. Ben Amor, "A system for learning continuous human-robot interactions from human-human demonstrations," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 2882–2889.

[50] J. van den Berg, S. Miller, D. Duckworth, H. Hu, A. Wan, X. Fu, K. Goldberg, and P. Abbeel, "Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations," in *Proceedings of the 34th IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 2074–2081.

[51] C. Lauretti, F. Cordella, E. Guglielmelli, and L. Zollo, "Learning by Demonstration for Planning Activities of Daily Living in Rehabilitation and Assistive Robotics," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1375–1382, 2017.

[52] G. J. Maeda, G. Neumann, M. Ewerton, R. Lioutikov, O. Kroemer, and J. Peters, "Probabilistic movement primitives for coordination of multiple human–robot collaborative tasks," *Autonomous Robots*, vol. 41, p. 593–612, 2017.

[53] H. C. Ravichandar, D. Trombetta, and A. P. Dani, "Human Intention-Driven Learning Control for Trajectory Synchronization in Human-Robot Collaborative Tasks," *IFAC-PapersOnLine*, vol. 51, no. 34, pp. 1 – 7, 2019, proceedings of the 2nd IFAC Conference on Cyber-Physical and Human Systems CPHS 2018.

[54] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-End Driving Via Conditional Imitation Learning," in *2018 Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 4693–4700.

[55] Y. Pan, C.-A. Cheng, K. Saigol, K. Lee, X. Yan, E. A. Theodorou, and B. Boots, "Imitation learning for agile autonomous driving," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 286–302, 2020.

[56] E. Kaufmann, A. Loquercio, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza, "Deep Drone Acrobatics," in *2020 Proceedings of the Robotics: Science and Systems*, 2020.

[57] P. Abbeel, A. Coates, and A. Y. Ng, "Autonomous Helicopter Aerobatics through Apprenticeship Learning," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1608–1639, 2010.

[58] A. Farchy, S. Barrett, P. MacAlpine, and P. Stone, "Humanoid Robots Learning to Walk Faster: From the Real World to Simulation and Back," in *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*, 2013, p. 39–46.

[59] Çetin Meriçli and M. Veloso, "Biped walk learning through playback and corrective demonstration," in *Proceedings of the 24th Conference on Artificial Intelligence*, 2010, pp. 1594–1599.

[60] J. Kolter, P. Abbeel, and A. Ng, "Hierarchical Apprenticeship Learning with Application to Quadruped Locomotion," in *2008 Proceedings of the Advances in Neural Information Processing Systems*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds., vol. 20, 2008, pp. 769–776.

[61] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato, "Learning from demonstration and adaptation of biped locomotion," *Robotics and Autonomous Systems*, vol. 47, no. 2, pp. 79 – 91, 2004.

[62] O. Kroemer, S. Niekum, and G. D. Konidaris, "A Review of Robot Learning for Manipulation: Challenges, Representations, and Algorithms," *Journal of Machine Learning Research*, vol. 22, pp. 1–82, 2021.

[63] D. A. Pomerleau, "ALVINN: An Autonomous Land Vehicle in a Neural Network," in *Proceedings of the Advances in Neural Information Processing Systems*, D. Touretzky, Ed., vol. 1. Morgan-Kaufmann, 1988, p. 305–313.

[64] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to End Learning for Self-Driving Cars," *arXiv e-prints*, Apr. 2016.

[65] S. Levine and V. Koltun, "Learning Complex Neural Network Policies with Trajectory Optimization," in *Proceedings of the 31st International Conference on International Conference on Machine Learning*, 2014, p. II–829–II–837.

[66] S. Chen, K. Saulnier, N. Atanasov, D. D. Lee, V. Kumar, G. J. Pappas, and M. Morari, "Approximating Explicit Model Predictive Control Using Constrained Neural Networks," in *2018 Proceedings of the Annual American Control Conference (ACC)*, 2018, pp. 1520–1527.

[67] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl, "Learning by Cheating," in *Proceedings of the 3rd Conference on Robot Learning*, L. P. Kaelbling, D. Kragic, and K. Sugiura, Eds., 2019, pp. 66–75.

[68] S. Ross, G. Gordon, and D. Bagnell, "A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, FL, USA, 11–13 Apr 2011, pp. 627–635.

[69] M. Palan, S. Barratt, A. McCauley, D. Sadigh, V. Sindhwani, and S. Boyd, "Fitting a Linear Control Policy to Demonstrations with a Kalman Constraint," in *Proceedings of the 2nd Annual Conference on Learning for Dynamics and Control*, 2020.

[70] R. Kálmán, "When Is a Linear Control System Optimal," *Journal of Basic Engineering*, vol. 86, pp. 51–60, 1964.

[71] A. Havens and B. Hu, "On Imitation Learning of Linear Control Policies: Enforcing Stability and Robustness Constraints via LMI Conditions," in *2021 Proceedings of the American Control Conference (ACC)*, 2021, pp. 882–887.

[72] M. Sassano and A. Astolfi, "A Local Separation Principle via Dynamic Approximate Feedback and Observer Linearization for a Class of Nonlinear Systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 1, pp. 111–126, 2019.

[73] A. Sultangazin, L. Pannocchi, L. Fraile, and P. Tabuada, "Watch and Learn: Learning to control feedback linearizable systems from expert demonstrations," in *2022 Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2022, to appear.

[74] A. Sultangazin, L. Fraile, and P. Tabuada, "Exploiting the experts: Learning to control unknown siso feedback linearizable systems from expert demonstrations," in *Proceedings of the 60th IEEE Conference on Decision and Control*, 2021.

[75] L. Fraile, M. Marchi, and P. Tabuada, "Data-driven Stabilization of SISO Feedback Linearizable Systems," *arXiv e-prints*, p. arXiv:2003.14240, Mar. 2021.

[76] A. Sultangazin, L. Pannocchi, L. Fraile, and P. Tabuada, "Learning to control from expert demonstrations," *arXiv e-prints*, p. arXiv:2203.05012, Mar. 2022.

[77] H. K. Khalil, *Nonlinear systems*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[78] D. Nesic, 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*, vol. 54, no. 3, pp. 619–624, 2009.

[79] A. Isidori, *Nonlinear Control Systems*, ser. Communications and Control Engineering. London, United Kingdom: Springer-Verlag, 1995.

[80] P. T. Kabamba, “Control of Linear Systems Using Generalized Sampled-Data Hold Functions,” *IEEE Transactions on Automatic Control*, vol. 32, no. 9, pp. 772–783, 1987.

[81] E. D. Sontag, “Comments on integral variants of ISS,” *Systems & Control Letters*, vol. 34, no. 1, pp. 93–100, 1998.

[82] R. Geiselhart, R. H. Gielen, M. Lazar, and F. R. Wirth, “An alternative converse Lyapunov theorem for discrete-time systems,” *Systems & Control Letters*, vol. 70, pp. 49–59, 2014.

[83] A. R. Teel and L. Praly, “A smooth Lyapunov function from a class-KL estimate involving two positive semidefinite functions,” *ESAIM - Control Optimization and Calculus of Variations*, p. 313–367, 2000.

[84] S. Boyd and L. Vandenberghe, *Convex Optimization.* New York, USA: Cambridge University Press, 2004.

[85] T. H. Chang, L. T. Watson, T. C. H. Lux, B. Li, L. Xu, A. R. Butt, K. W. Cameron, and Y. Hong, “A Polynomial Time Algorithm for Multivariate Interpolation in Arbitrary Dimension via the Delaunay Triangulation,” in *2018 Proceedings of the ACM Southeast Conference*, 2018.

[86] Guisheng Zhai, Bo Hu, K. Yasuda, and A. N. Michel, “Qualitative analysis of discrete-time switched systems,” in *2002 Proceedings of the American Control Conference*, vol. 3, 2002, pp. 1880–1885 vol.3.

[87] P. Hartman, *Ordinary Differential Equations*, 2nd ed. Society for Industrial and Applied Mathematics, 2002.

[88] S. M. Omohundro, “Geometric learning algorithms,” *Physica D: Nonlinear Phenomena*, vol. 42, no. 1, pp. 307–321, 1990.

[89] M. Hehn and R. D’Andrea, “Quadrocopter Trajectory Generation and Control,” *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1485–1491, 2011, proceedings of the 18th IFAC World Congress.

[90] D. Mellinger and V. Kumar, “Minimum snap trajectory generation and control for quadrotors,” in *2011 Proceedings of the IEEE International Conference on Robotics and Automation*, 2011, pp. 2520–2525.

[91] J. Hauser, S. Sastry, and P. Kokotovic, “Nonlinear control via approximate input-output linearization: the ball and beam example,” *IEEE Transactions on Automatic Control*, vol. 37, no. 3, pp. 392–398, 1992.

[92] A. Isidori and C. Byrnes, “Output regulation of nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 35, no. 2, pp. 131–140, 1990.