UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

Hierarchies and Compositional Abstractions of Hybrid Systems

Paulo Alexandre Trigo Neri Tabuada
(Licenciado)

Dissertação para obtenção do grau de Doutor em
Engenharia Electrotécnica e de Computadores

Orientador:

Doutor Pedro Manuel Urbano de Almeida Lima

Júri:

Presidente: *Reitor da Universidade Técnica de Lisboa*
Vogais: *Doutor Fernando Manuel Ferreira Lobo Pereira*
Doutor António Manuel dos Santos Pascoal
Doutora Ana Margarida de Lencastre Godinho Cannas da Silva
Doutor George J. Pappas
Doutor Pedro Manuel Urbano de Almeida Lima
Doutor João Fernando Cardoso Silva Sequeira

Janeiro de 2002

Resumo

Na última década a modelação, análise e controlo de sistemas complexos, embebidos e de larga escala, tem vindo a ser alvo de atenção crescente. Os avanços e o reduzido custo de novos e mais performantes dispositivos de comunicação, cálculo e sensoreamento alargam consideravelmente os limites do que é hoje exequível. As aplicações actuais ultrapassam já o conhecimento formal e teórico que existe sobre estes sistemas pelo que uma abordagem formal reveste-se de particular importância. Neste sentido, propõe-se algumas soluções nesta tese ao considerar *Sistemas Híbridos* como modelo formal para sistemas embebidos.

Neste trabalho introduz-se um enquadramento teórico e abstracto para o estudo de sistemas de controlo incluíndo sistemas de controlo discretos, contínuos e híbridos. Uma noção de *abstração* é apresentada para sistemas de controlo híbridos que pode ser encarada como um sistema quociente que preserva as propriedades de interesse enquanto ignora detalhes de modelação. É dedicada especial atenção a sistemas de larga-escala que são usualmente construídos através da interligação de subsistemas. Uma noção formal de *composição* é também introduzida por forma a modelar a interligação e sincronização de subsistemas. Mostra-se que a noção de abstração é composicional no sentido em que a composição de abstrações de subsistemas é uma abstração do sistema global. É também proposto um algoritmo para calcular abstrações de sistemas híbridos. Estes resultados perspectivam uma metodologia hierárquica para efectuar tarefas de análise e síntese em sistemas de control híbridos.

Palavras Chave: Sistemas Híbridos, Sistemas de Controlo, Abstracções, Composicionalidade, Hierarquias.

Abstract

In the last decade an increasing attention has been paid to the modelling, analysis and control of large-scale, embedded, complex systems. The advances and the low cost of new and more powerful computing, sensing and communicating devices push further the limits of what is now possible to accomplish. Todays applications have gone way beyond the formal and theoretical understanding we have about those systems. This fact suggests a formal approach and this thesis provides some answers by regarding *Hybrid Systems* as a formal model for embedded systems.

In this work we introduce an abstract framework for the study of control systems capturing continuous, discrete and hybrid control systems. A notion of *abstraction* is defined for hybrid control systems which can be regarded as a quotient system that preserves properties of interest while ignoring modelling details. Special attention is devoted to large scale systems which are usually built by interconnecting smaller subsystems. A formal notion of *composition* is also introduced to model the interconnection and synchronization of subsystems. It is shown that the notion of abstraction is compositional in the sense that by composing abstractions of subsystems one obtains an abstraction of the overall system. An algorithm is proposed to compute abstractions of hybrid control systems providing a useful tool to deal with the inherent complexity of embedded systems. These results perspectivate a hierarchical methodology to perform analysis and design tasks for hybrid control systems.

Keywords: Hybrid Systems, Control Systems, Abstractions, Compositionality, Hierarchies.

Acknowledgments

Three years after the start of this endeavor I finally feel ready to start learning some interesting things. However, to reach this point, I benefited from the help of some people whose names I would like to leave associated with this work, although their influence goes far beyond it.

I start by thanking to my advisor Pedro Lima who was always able to give me enough scientific freedom to explore the many interesting problems I came across during my work, although they sometimes diverged from the research lines he envisioned for his group.

I would also like to acknowledge three people (in chronological order) who have shaped me as a researcher and as a person, and from whom I would like to consider myself a disciple. My friend and former colleague João Soares who always encouraged me to look for the essence of the problems and not to be satisfied with partial answers or superficial knowledge. I also owe him a lot for awaking my interests in mathematics that had been dormant since my *Licenciatura* math. courses. He was always able to highlight the beauty of the ideas behind the sometimes hermetic language, concepts and constructions of the several mathematical fields I came across during my work. The second person I whish to express my gratitude is my brother Gonçalo Tabuada. He always managed to keep my intellect alive by means of his questions, conversations or mere observations. There were few weekends during the past three years where I have not learned something new from him. From counter intuitive examples in set theory to many interesting conceptual enlightenments in group theory and in algebra in general. Our discussions on the interplay between the finite, the denumerable infinite and nondenumerable infinities and its interpretations in automata, dynamical and control systems were an important motivation for my work on Hybrid Systems. I will certainly never forget his teasing words *" there is a world of mathematics beyond differentiability"*. Finally I must mention George J. Pappas with whom I had the pleasure to work during my stays at the University of Pennsylvania. He has taught me some of the *tricks of the trade* and introduced me to some of the interesting topics discussed on this thesis. Besides his unquestionable technical talent I was always impressed by his outstanding perspective and vision for the open problems and important research directions for the medium-long range future of the control and hybrid communities.

I would also like to thank Esfandiar Hagverdi for the stimulating discussions on the categorical interpretations of dynamical and control systems, either in the discrete, continuous or hybrid world. His stay at the University of Pennsylvania was certainly a bonus for me.

To balance my everyday life I counted with the collaboration of many other people like Pedro Terruta and Pedro Vieira who made noyAghem such a delightful musical adventure. I will always be grateful for the overwhelming moments we lived together during our performances which relied on the availability and help from Anabela Matos and Raul, who sometimes offered us what we did not dare to ask. A special thanks for the SINDICATO gang, I will always be a fan of their forthcoming projects.

Some final words of acknowledgment to the members of the jury for serving in my committee.

Finally, I will always be indebted to my parents that have complied with all my needs during this last three years and to Celia Matos who lived this adventure by my side experiencing my mind absences and humor changes every time I found a counter example to my conjectures!

Contents

CHAPTER 1

Introduction

In the last decade an increasing attention has been paid to the modeling, analysis and control of large-scale, embedded, complex systems. The thrust from the application side has been tremendous and includes, among others:

- Automotive engines, where discrete phenomena such as torque generation and spark ignition interacts with the continuous evolution of the power train and air dynamics [9], see also [8].
- Air-Traffic management where discrete decisions about the continuous evolution of several aircrafts are addressed [82].
- Chemical batch plants operating in multi-batch mode where a discrete sequence of continuous actions such as mixing, heating or cooling products needs to be determined in order to produce the desired product [54].
- Manufacturing industry where some processes are modeled by a continuous and a discrete layer. In the continuous, time driven layer, the manufacturing of products is described by continuous dynamics whereas on the discrete layer, a discrete event system models the manufacturing system based on events generated by the continuous processes [22].
- Process control [68].
- TCP congestion control [28].
- Biomolecular networks [3].
- etc.

It is fair to say that embedded systems are now everywhere where we mean by embedded systems all those applications where computing systems interface the continuous world through sensors and actuators. The advances and the low cost of new and more powerful computing, sensing and communicating devices push further the limits of what is now possible to accomplish. Todays applications have gone way beyond the formal and theoretical understanding we have about those systems. In fact, designing embedded systems is a very difficult task since several different domain specific techniques must be combined together. Software engineering and concurrency theory techniques as well as real-time scheduling need to meet signal processing and control theory to accommodate the needs of embedded systems. The increasingly sophistication of the products, the large number of modes of operation as well as interactivity and dynamic reconfigurability render impossible for single engineer to completely design an embedded system. These difficulties call for a formal approach. In this spirit we regard *Hybrid Systems* as a formal model for embedded systems since it allows to specify both the continuous (world) dynamics as well as the discrete (computational) dynamics.

The emerging complexity of embedded systems also raises a fundamental question that we partially address in this work: how to ensure that embedded systems satisfy their specifications? The high complexity of these systems as well as the different scientific techniques used in their design makes almost impossible to formally prove that the system satisfies desired properties. Two approaches to this question seem specially promising: one is to satisfy the specifications by construction so that it is not required to prove that the final system meets its requirements. The other is to prove the desired properties by taking advantage of the structure of large-scale complex embedded systems. In any case, formal methods are necessary to understand how the properties of subsystems are propagated or preserved by the interconnection and synchronization of these subsystems. This clearly demands for formal notions of *compositionality* between subsystems or submodules. It is also necessary to have formal notions of *abstraction* for complexity reduction of these systems. Abstractions allows macro modeling by ignoring modeling details that are unimportant at a desired level of abstraction. When an engineer is developing a particular module he only needs to take into consideration the behavior of the general system that influences or is influenced by the specific module under development. He would therefore consider only two systems: the module to be designed and an abstraction of the remaining system that hides irrelevant details. The concepts of abstraction and compositionality will be recurrent themes through this thesis.

1. Hierarchies of Compositional Abstractions

It has been recognized and widely accepted that hierarchies are a very useful way of dealing with the complexity of large scale systems. Examples of the use of hierarchies are commonly spread throughout systems engineering. However, its use in real applications, and sometimes even in the academic world has not been followed by an effort to formalize and to understand the modeling power and expressiveness or the analysis and synthesis advantages/drawbacks when compared with single-layered models. Except for the theoretical computer science community which has already developed very mature notions of *abstraction* and *composition*, in particular, in the areas of concurrency theory [52] [89], and computer aided verification [48], no such effort was ever made in the control community. This effort, by the computer scientists, has resulted in formal and very meaningful notions of abstraction which are used to tackle exponential explosion of purely discrete systems. Given a discrete system, an abstraction can be seen as a quotient system that preserves some properties of interest while ignoring modeling details. Language equivalence, simulation, and bisimulation are established notions of abstraction for discrete systems that preserve properties expressed in various temporal logics.

We believe, however, that these ideas, notions and concepts are so general and useful that it is very worth it to transpose them to the continuous as well to the hybrid world. From the continuous side this line of research initiated in [62] and has resulted in automatic constructions of abstractions for linear control systems, nonlinear control systems [63, 64] and Hamiltonian control systems [77] while preserving control theoretic properties. Preliminary investigations trying to combine the continuous with the discrete results were presented in [77], however, we take a different and more general approach in this thesis that comprise those results as a special case. Other approaches to this problem in the hybrid case are described in [4, 15, 18, 69].

The approach taken in this work regards discrete, continuous and hybrid systems as particular examples of a more general notion of *abstract control systems*. It is within this class of systems that the notions of simulation, bisimulation and abstraction will be formulated. We identify the structure of abstract control systems and restrict the class of maps between them to those that respect that structure. This is elegantly presented by making use of some elementary notions of category theory. We therefore define the category of abstract control systems which will serve as the domain of mathematical discourse for our study. An abstraction of a given abstract control system will simply be another abstract control system such that there is a structure preserving (morphism) surjective map from the original system to the abstraction. This quotienting or aggregation map defines what is ignored and what remains from the original model. All the properties that will be preserved from the original system to the abstraction or reflected from the abstraction to the original system will depend critically on the structure that is preserved by the map relating both systems. We determine which further assumptions on the abstracting maps are required to preserve hybrid systems relevant properties.

Structure preserving maps are closed under composition and this property allows to build an hierarchy of different levels of abstraction. If one starts with system A, one can extract an abstraction B and then further abstract C from B. By composing the aggregation maps we ensure that C is still an abstraction from the original system, as displayed in Figure 1. By this process we can formalize an hierarchy with any finite number of levels and provide a conceptual basis for a hierarchical approach to proof, verification or design methodologies for large-scale systems. Suppose we want to prove that property P is true for system A. If the maps between system A and its abstractions are such that all the models are equivalent with respect to that property, then determining if the property holds for A is equivalent to determining if the property holds for C, which has lower complexity.

Another related concept that is extremely useful in dealing with the complexity of large scale systems is *compositionality*. Common large-scale systems are built by interconnecting smaller subsystems. This should be considered as structure for those particular systems that should be exploited to further reduce the complexity of analysis and/or design tasks. We introduce a formal notion of parallel composition with synchronization, modeling this aspect of large-scale systems, and show how we can use it to simplify

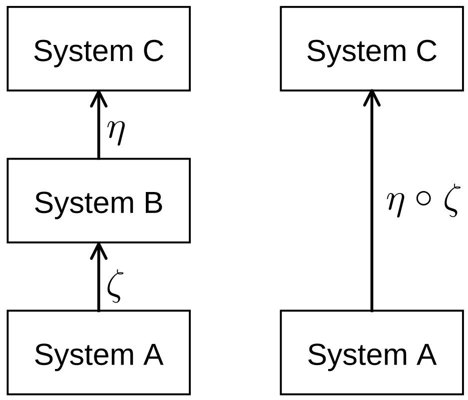

FIGURE 1. A hierarchy of abstractions of system A.

the task of computing abstractions. Indeed, we show that abstractions are compatible with parallel composition in the sense that if system A is in fact built by interconnecting subsystems A_1, A_2 and A_3, then we can abstract each A_i to B_i, individually. Compatibility now means that the system obtained by interconnecting the subsystems B_1, B_2 and B_3 is an abstraction of system A as displayed in Figure 2. Clearly the task of abstracting each subsystem will be easier to accomplish then abstracting the whole system A, specially for large-scale systems.

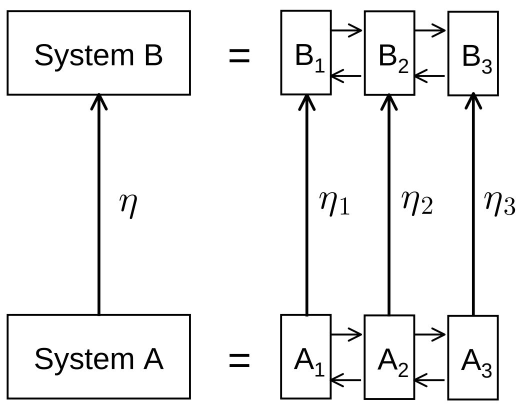

FIGURE 2. Abstraction of system A as a whole and subsystem by subsystem.

These ideas will be discussed in grater detail in the next section where we summarize the thesis chapters.

2. Thesis outline

This thesis is divided into 6 chapters covering some aspects of continuous abstractions, hybrid abstractions and multi-agent systems.

2.1. Mathematical Background. In this chapter we review some miscellaneous mathematical facts required through the thesis. We introduce some elementary notions of category theory which will provide the formal setup for our study of compositional abstractions. Some ideas from theoretical computer science provide the necessary background for the discrete part of hybrid systems while the continuous part requires some notions of differential geometry and differential geometric control theory which are also presented in this section.

2.2. A Walk Through the Continuous World. With the goal of developing a general theory of abstractions for hybrid control systems comprising the already existing theory for discrete systems and the recent developed theory for continuous system, we faced the need to extend the existing continuous results. Actually, we wanted to define a parallel composition operator with synchronization for hybrid systems that would have as a special case the existing results for composition of transition systems with event synchronization. It so happens that, in our interpretation, the events correspond to the inputs of a control system and the existing results for continuous abstractions did explicitly model the inputs. We have thus extended the continuous abstraction theory from the state space manifold to the corresponding control bundle. In this chapter we present a notion of simulation explicitly modeling the inputs that is equivalent to the existing one, and characterize the geometry of the control bundle of a simulation induced by an equivalence relation on the base space of the original control system. We were strongly influenced by some ideas of category theory and handled the problem in a categorical way. This turned out to be useful in various ways since we gained a much deeper insight into the structure of continuous control systems. But, perhaps even more important, is the fact that we were able to distinguish which properties of continuous control systems where intrinsic and which depended on the additional structure we assumed (smoothness). With these insights, provided by the categorical approach, we developed a similar theory for hybrid control systems in the next chapter. It was also extremely rewarding the fact that a large number on interesting problems and research directions were also unveiled in this walk trough the continuous world.

This chapter aimed at a conceptual and formal understanding of the structure of a hierarchy of control bundles induced by an hierarchy of abstractions. We have also exposed the structure of the maps relating the inputs of a control system to the inputs of its abstraction. Although the results enable the development of a hierarchical control theory for continuous systems it was never the purpose to proceed towards results directly useful to the practitioner. In fact, the scarce examples and the language of category theory may repeal some readers although we have only used some elementary facts from control theory and differential geometry in our approach. To overcome these difficulties we made Chapter 4 independent of Chapter 3, except for some references that can safely be ignored without risking the comprehension of that chapter.

2.3. Hybrid Control Systems. This chapter of the thesis contains the major contributions. A completely abstract and general theory of control systems is presented. In this general framework, strongly influenced by simple categorical ideas, we defined and proved all the relevant concepts and results that we later specified to hybrid control systems. On the first part of this chapter we provide a general notion of control system encompassing discrete, continuous and hybrid control systems. We introduce a notion of abstraction and determine some preserved properties. This notion of abstraction also defines an equivalence relation on the class of control systems if we render it symmetric since it was already transitive and reflexive. We give conditions for equivalence which are, in principle, easier to check than the definition and move towards compositionality. We define a composition operator that models a system built by the interconnection and synchronization of two (or any finite number of) subsystems. We also show that our operator is compatible with the introduced notion of abstraction. On the second part of the chapter all of these results are instantiated for the hybrid case and some sufficient results (which are easier to check then the sufficient and necessary ones) are also given. We also provide a very brief treatment of the additional assumptions required for abstractions to preserve a purely hybrid phenomena: Zeno sequences. It is fair to say that most of the work in this chapter was strongly influenced by computer science ideas specially in the fields of concurrency and computer aided verification and that we followed closely [89] in our developments. We have, however, taken a control theory twist in the interpretation of some of the concepts and results.

Although we provide the standard definition of hybrid control systems, the hybrid automaton, we preferred to work in the abstract setting introduced in the first part of the chapter. However, when specializing the developed results for hybrid control systems, we returned to the notation and concepts of the hybrid automaton to make the developed results accessible to a wider audience. As in the second chapter, the abstract formulation of the addressed problems and the language of category theory may not please all of the readers, specially those from the control community where computer science ideas and categorical language are rather new. We feel, however, that it is a risk worth taking as the technological advances are pushing the limits of our knowledge further and further with increasingly complicated problems. This can only be matched by an effort from the control community to use more sophisticated and diverse mathematical tools to address these new problems. In this sense, this work represents a step towards this new interdisciplinary vision of the *new systems and control theory*.

2.4. Formations and Abstractions of Multi-Agent Systems. This chapter collects some results on formations of multi-agent systems as an illustrative example of some of the concepts introduced in Chapter 4. Since the word *agent* may have different meaning according to the scientific community where it is employed it matters to stress that we mean by multi-agent systems, systems composed by several *control systems* that usually require some form of communication, coordination or cooperation to achieve the desired specifications. In this regard we introduce a formal model for formations allowing the study of the feasibility problem: *Given a set of agents, their kinematics, a set of inter-agent constraints defining the formation, determine if there are trajectories for the individual agents satisfying all the constraints.* This problem is solved and the computations necessary to determine the answer to this question lead also to the solution of the group abstraction problem: *Given a feasible formation, extract a smaller control system,* the group abstraction*, representing the formation as a whole.* This new control system that we call the formation or group abstraction has smaller complexity than the original control systems and also ensures that all its trajectories satisfy the formation constraints.

The group abstraction introduced in this chapter is in fact an instantiation of the notion of parallel composition with synchronization that was introduced for abstract control systems in Chapter 4. In particular, the group abstraction is no more then the parallel composition of the individual agents with synchronization over the formation constraints.

This work on formation was conceived in order to be accessible to wide audience comprising the robotics, control and aerospace communities. In this sense we have deliberately emphasized the readability over the mathematical sophistication. We have, therefore, preferred to talk about pointwise solving equations of the form $Ax = b$ on manifolds than to talk about exterior differential systems with independence conditions.

2.5. Conclusions. In the last chapter we review the contributions of this thesis, present the overall conclusions as well as several important topics for further research.

CHAPTER 2

Mathematical Background

In this section we review the basic mathematical concepts required for the presentation of the ideas in this work.

1. Miscellaneous

We start by reviewing some miscellaneous mathematical facts to set notation. If A is a set, we denote the set of all subsets of A, also called the power set of A, by $\mathcal{P}(A)$. Let $f : A \longrightarrow B$ be a map, if S is a subset of A we denote by $f(S)$ the subset of B defined by:

$$f(S) = \bigcup_{s \in S} f(s) \tag{2.1}$$

When f is a linear map between modules or vector spaces we denote the range of f by $\mathcal{R}(f) = f(A)$. We also use the set notation $f^{-1}(b)$ to refer to all the points $a \in A$ such that $f(a) = b$ and if S is a subset of B we denote by $f^{-1}(S)$ the set:

$$f^{-1}(S) = \bigcup_{s \in S} f^{-1}(s) \tag{2.2}$$

1.1. Relations. A relation is a generalization of a function in the sense that it assigns to each element in its domain a *set* of elements in its codomain. Mathematically a relation R between the sets S_1 and S_2 is simply a subset of their Cartesian product, that is:

$$R \subseteq S_1 \times S_2 \tag{2.3}$$

The domain of a relation is the set:

$$dom(R) = \{s_1 \in S_1 \ : \ \exists s_2 \in S_2 \quad (s_1, s_2) \in R\} \tag{2.4}$$

and the range of a relation is defined by:

$$range(R) = \{s_2 \in S_2 \ : \ \exists s_1 \in S_1 \quad (s_1, s_2) \in R\} \tag{2.5}$$

A relation is said surjective if $range(R) = S_2$. Given two relations $R \subseteq S_1 \times S_2$ and $R' \subseteq S_2 \times S_3$ we can define their composition to be the relation $R' \circ R \subseteq S_1 \times S_3$ defined by:

$$R' \circ R = \{(s_1, s_3) \in S_1 \times S_3 \ : \ \exists s_2 \in S_2 \quad (s_1, s_2) \in R \wedge (s_2, s_3) \in R'\} \tag{2.6}$$

Given a relation $R \subseteq S_1 \times S_2$ we denote its inverse relation as $R^{-1} \subseteq S_2 \times S_1$, given by:

$$R^{-1} = \{(s_2, s_1) \in S_2 \times S_1 \ : \ (s_1, s_2) \in R\} \tag{2.7}$$

An object that we will use frequently is the set valued map $R : S_1 \longrightarrow \mathcal{P}(S_2)$ induced by a relation R and defined by:

$$R(s_1) = \{s_2 \in S_2 \ : \ (s_1, s_2) \in R\} \tag{2.8}$$

Given a map $f : S_1 \longrightarrow S_2$ it induces the relation $R = \{(s_1, s_2) \in S_1 \times S_2 \ : \ s_2 = f(s_1)\}$. Conversely, every relation $R \subseteq S_1 \times S_2$ with domain $dom(R) = S_1$ and such that $R(s_1)$ is a singleton for every $s_1 \in S_1$ defines a map $f : S_1 \longrightarrow S_2$, by $f(s_1) = R(s_1)$.

We also introduce some notation for later use. Given relations $R_1 \subseteq S_1 \times S_2$, $R_2 \subseteq S_3 \times S_4$ and a subset $L \subseteq S_1 \times S_3$ we define the new relations $R_{1\times 2}$ and $R_{1\times 2}|_L$ as:

$$R_{1\times 2} = \{((s_1, s_3), (s_2, s_4)) \in (S_1 \times S_3) \times (S_2 \times S_4) \ : \ (s_1, s_2) \in R_1 \ \wedge \ (s_3, s_4) \in R_2\} \tag{2.9}$$

$$R_{1\times 2}|_L = \{((s_1, s_3), (s_2, s_4)) \in R_{1\times 2} \ : \ (s_1, s_3) \in L\} \tag{2.10}$$

The Cartesian product $S_1 \times S_2$ comes equipped with two projection maps $\pi_{S_1} : S_1 \times S_2 \longrightarrow S_1$ and $\pi_{S_2} : S_1 \times S_2 \longrightarrow S_2$. If we now choose a subset R of the product such that $\pi_{S_1}(R) = S_1$ we can regard this subset R as a (set theoretic) fiber bundle over the base space S_1 and we call R a fibering relation. The fiber over $s \in S_1$, denoted by $R_s = \pi_{S_1}^{-1}(s)$ is given by all the elements $r \in R$ such that $\pi_{S_1}(r) = s$. We also denote an element $r = (a, b) \in R$ by b_a when we whish to emphasize the fiber part of r.

1.2. Monoids. A monoid is a triple $(\mathcal{M}, \cdot, \varepsilon)$ where $\mathcal{M}$ is a set closed under the associative operation $\cdot : \mathcal{M} \times \mathcal{M} \longrightarrow \mathcal{M}$ and ε is a special element of $\mathcal{M}$ called identity. This element satisfies $\varepsilon \cdot m = m \cdot \varepsilon = m$ for any $m \in \mathcal{M}$. We will usually denote $m_1 \cdot m_2$ simply by $m_1 m_2$ and refer to the monoid simply as $\mathcal{M}$. Given two elements m_1 and m_2 from $\mathcal{M}$ we say that m_1 is a prefix of m_2 iff there exists another $m \in \mathcal{M}$ such that $m_1 m = m_2$. Suppose now that we have a fibering relation $R \subseteq S \times \mathcal{M}$ with base space S. If $\pi_S^{-1}(s)$ contains (s, ε) and is prefix closed for every $s \in S$ then we call R a fibering monoid.

We now relate relations with fiber bundles and monoids. Suppose that the sets S_1 and S_2 are in fact fiber bundles. Then a relation $R \subseteq S_1 \times S_2$ induces a relation $R_B \subseteq B_1 \times B_2$ on the base spaces B_1 and B_2 of S_1 and S_2, respectively, defined by:

$$(b_1, b_2) \in R_B \text{ iff } (b_1, b_2) = (\pi_{S_1}(s_1), \pi_{S_2}(s_2)) \text{ and } (s_1, s_2) \in R \tag{2.11}$$

If the fiber bundles have a richer structure such as fibering monoids we need the relation to respect that structure. We then say that a relation $R \subseteq S_1 \times S_2$ between two fibering monoids is fibering monoid respecting iff satisfies:

- **Identity:** $(b_1, b_2) \in R_B \ \Rightarrow \ ((b_1, \varepsilon), (b_2, \varepsilon)) \in R$

- **Semi-group:** $((b_1, m_1), (b_2, m_2)), ((b'_1, m'_1), (b'_2, m'_2)) \in R$ and $(b_1, m_1 m'_1) \in S_1$
  $\Rightarrow ((b_1, m_1 m'_1), (b_2, m_2 m'_2)) \in R$.

2. Category Theory

In this work we will not have the opportunity to fully take advantage of the doors opened by category theory, but we will rather make an elementary use of it. We point the reader to [43] for further details as well to [44] and [5] (by this order) for a sequence of books that provide the necessary "maturity" for [43]. Informally speaking, a category is a universe of mathematical discourse and is perhaps better described by examples. If one is interested in group theory one would certainly work in the universe of groups and group homomorphism, whereas if one is learning elementary topology the natural universe are topological spaces and continuous maps between then. In linear algebra one deals with vector spaces and linear maps, in differential geometry with smooth manifolds and smooth maps between then, etc. This idea of universe of mathematical discourse can be formally defined as follows:

Definition 2.1 (Category). A category is a tuple $(\mathcal{O}, hom, id, \circ)$ consisting of:

- A class of objects $\mathcal{O}$.
- For each pair of objects (A, B) belonging to $\mathcal{O}$, a set $hom(A, B)$. The elements of $hom(A, B)$ are called morphisms from A to B. An element of this set $f \in hom(A, B)$ is usually denoted graphically as $A \xrightarrow{f} B$.
- For each object $A \in \mathcal{O}$ a special morphism $A \xrightarrow{id_A} A$, called the identity on A.
- A binary operation which maps a pair of morphisms $(A \xrightarrow{f} B, B \xrightarrow{g} C)$ to the composite[1] $A \xrightarrow{g \circ f} C$ while satisfying:
  - Associativity: $h \circ (g \circ f) = (h \circ g) \circ f$ whenever the composition is defined.
  - Identity: for a morphism $A \xrightarrow{f} B$ we have $id_B \circ f = f = f \circ id_A$.
  - The sets $hom(A, B)$ are pairwise disjoint.

In the above examples the objects are the groups, topological spaces, etc, while the arrows are the group homomorphisms, continuous maps, etc, between them. As morphisms are displayed graphically, more elaborate relations between morphisms are usually displayed in commutative diagrams. We shall say that a diagram commutes iff the composition of morphisms in any path from one object to another object is

[1] Note that composition of f and g is only defined if the target of f equals the source of g.

the same. Consider for example the following diagram

$$\begin{tikzcd}
A \arrow[r, "f"] \arrow[d, "h"'] & B \arrow[d, "g"] \\
C \arrow[r, "j"'] & D
\end{tikzcd} \tag{2.12}$$

where commutativity simply means that the two existing paths from A to D are equal, that is $g \circ f = j \circ h$.

We will almost only use concrete categories where all the objects can be seen as sets with added structure and the morphisms are maps between the sets that preserve the structure. This is easily seen for topological spaces which are sets with the added collection of open sets as structure or manifolds which are sets equipped with a maximal atlas.

We shall make some use of the following objects:

Definition 2.2 (Product). Let A and B be objects in a category. The product of A and B is the triple (C, π_A, π_B) such that for any other triple (C', π'_A, π'_B) there exists one and only one morphism η making the following diagram commutative:

$$\begin{tikzcd}
A & C \arrow[l, "\pi_A"'] \arrow[r, "\pi_B"] & B \\
& C' \arrow[ul, "\pi'_A"] \arrow[u, "\eta"] \arrow[ur, "\pi'_B"'] &
\end{tikzcd} \tag{2.13}$$

Note that the product captures the relevant notion of product with respect to the corresponding category. The product on the category of sets and maps between them is the usual Cartesian product, while in the category of groups is the direct product, in the category of topological spaces is the Cartesian product of the supports equipped with the product topology, etc.

Another object that we will use to capture the notion of embedding a system into a larger system is the equalizer:

Definition 2.3 (Equalizer). Let g and h be morphisms in a category. The equalizer of g and h is the morphism f satisfying $g \circ f = h \circ f$ and such that for any other morphism f' satisfying $g \circ f' = h \circ f'$ there is one and only one morphism $\overline{f}$ such that the following diagram commutes:

$$\begin{tikzcd}
A \arrow[r, "f"] & B \arrow[r, shift left, "g"] \arrow[r, shift right, "h"'] & C \\
A' \arrow[u, "\overline{f}"] \arrow[ur, "f'"'] & &
\end{tikzcd}$$

The notion of co-equalizer, dual to the notion of equalizer, will also play an important role since co-equalizers can be regarded as the categorical formalization of the continuous abstraction process described in Chapter 3:

Definition 2.4 (co-Equalizer). Let g and h be morphisms in a category. The co-equalizer of g and h is the morphism f satisfying $f \circ g = f \circ h$ and such that for any other morphism f' satisfying $f' \circ g = f' \circ h$ there is one and only one morphism $\overline{f}$ such that the following diagram commutes:

$$\begin{tikzcd} A \arrow[r, shift left, "g"] \arrow[r, shift right, "h"'] & B \arrow[r, "f"] \arrow[dr, "f'"'] & C \arrow[d, "\overline{f}"] \\ & & C' \end{tikzcd}$$

Another relevant concept is that of free object, we now provide a particular version of the concept that is enough for our needs:

Definition 2.5 (Free Object). Let A be an object in a category, S a set and $i : S \longrightarrow A$ the inclusion map taking $s \in S$ to $i(s) = s \in A$. We say that A is free on the set S or that A is freely generated by S iff for every map i' from S to A' there exists one and only one morphism $\overline{i}$ such that the following diagram commutes:

$$\begin{tikzcd} S \arrow[r, "i"] \arrow[dr, "i'"'] & A \arrow[d, "\overline{i}"] \\ & A' \end{tikzcd}$$

The elements of S are also usually called the generators of A. We then see that in order to specify a morphism from a freely generated object to another object it suffices to define the morphism on the generators since it extends in a unique way to a morphism defined on its domain. This is something well known, for example, in the category of vector spaces. To define a linear map between vector spaces it suffices to define it on the basis of that space since it extends in a unique way to all the elements of the vector space by linearity.

3. Labeled Transition Systems

As already stated in the introduction several ideas from theoretical computer science play a crucial role in hybrid systems theory and also on this thesis. We now recall the concept of labeled transition systems:

Definition 2.6 (Labeled Transition Systems). A labeled transition system is a triple $(Q, \Sigma, \longrightarrow)$ where Q is a set of states, Σ is a set of labels or events and $\longrightarrow \subseteq Q \times \Sigma \times Q$ is a (transition) relation. If furthermore Q and Σ are finite we have a discrete labeled transition system.

Although this notion has its roots in theoretical computer science and digital systems [29] we shall interpret it in a control theoretic way which even differs from the discrete event systems community [70, 71, 16]:

The set Q is our model for the "state-space", Σ is a set of labels associated with the choices and the relation $\longrightarrow$ determines how the choices govern the evolution. An element $(q_1, \sigma, q_2) \in\longrightarrow$ is usually represented graphically as $q_1 \xrightarrow{\sigma} q_2$ and is interpreted as the choice σ effectuated at state q_1 has the effect of making the system evolve to the new state q_2. Note that by using a relation to model the evolutions we allow nondeterminism in the sense that both triples (q_1, σ, q_2) and (q_1, σ, q_3) may belong to $\longrightarrow$, for example. However in this work we will make the assumption that all the systems are deterministic so that we can replace the relation $\longrightarrow$ with the partially defined next-state map $\delta : Q \times \Sigma \longrightarrow Q$.

Definition 2.7 (Input Trajectories). Given a discrete transition system $(Q, \Sigma, \longrightarrow)$ and a state $q_0 \in Q$, an input trajectory (also called a sequence, string or trace) starting at q_0 is a finite sequence of labels $\sigma_1\sigma_2\ldots\sigma_i\ldots\sigma_n$ such that $q_0 \xrightarrow{\sigma_1} q_1$, $q_1 \xrightarrow{\sigma_2} q_3$, ... and $q_{n-1} \xrightarrow{\sigma_n} q_n$, for some $q_i \in Q$, $i = 1, \ldots, n$.

Although the emphasis on discrete control systems in on the input trajectories that can be feed (or that are accepted by) to the transition system, for continuous control systems the emphasis is on the sequence of states that are visited by some choice of inputs. In fact, we regard the labels $\sigma \in \Sigma$ as inputs that we can control to influence the evolution described by δ, where as in the computer science community events are triggered by some external element that is beyond our control.

4. Differential Geometry

We now review the necessary concepts from differential geometry following more or less closely [1] and [12].

In this work we understand by a smooth manifold an Hausdorff, second countable differentiable manifold. Let M be a smooth manifold and T_xM its tangent space at $x \in M$. The tangent bundle of M is denoted by $TM = \cup_{x\in M} T_xM$ and π_M is the canonical projection map $\pi_M : TM \longrightarrow M$ taking a tangent vector $X(x) \in T_xM \subset TM$ to the base point $x \in M$. We recall that T_xM has a vector space structure over the real field. Dually we define the cotangent bundle to be $T^*M = \cup_{x\in M} T_x^*M$, where T_x^*M is the linear space of all linear maps from T_xM to the real field. The cotangent bundle also comes equipped with a natural projection map from T^*M to M. Both TM and T^*M can be endowed with the structure of a module over the ring of smooth real functions on M. Now let M and N be smooth manifolds and $\phi : M \longrightarrow N$ a smooth map, we denote by $T_x\phi : T_xM \longrightarrow T_{\phi(x)}N$ the induced tangent map which maps tangent vectors from T_xM to tangent vectors at $T_{\phi(x)}N$. If ϕ is such that $T_x\phi$ is surjective at $x \in M$ we say that ϕ is a submersion at x. When ϕ is a submersion at every $x \in M$ we simply say that it is a submersion. When ϕ has an inverse which is also smooth we call ϕ a diffeomorphism. We say that a manifold M is diffeomorphic to a manifold N, denoted by $M \cong N$, when there is a diffeomorphism between M and N.

When this is the case we can define the pullback of a vector field $Y \in TN$, denoted by ϕ^*Y, as the vector field $X \in TM$ given by $X(x) = T_{\phi(x)}\phi^{-1}Y(\phi(x))$.

To later describe control systems we will need the concept of fiber bundle:

Definition 2.8 (Fiber Bundle). A fiber bundle is a tuple $(B, M, \pi_B, \mathcal{F}, \{O_i\}_{i\in I})$, where B, M and $\mathcal{F}$ are smooth manifolds called the *total space*, the *base space* and *standard fiber* respectively. The map $\pi_B : B \longrightarrow M$ is a surjective submersion and $\{O_i\}_{i\in I}$ is an open cover of M such that for every $i \in I$ there exists a diffeomorphism $\Psi_i : \pi_B^{-1}(O_i) \longrightarrow O_i \times \mathcal{F}$ making the following diagram commutative:

$$\begin{tikzcd} \pi_B^{-1}(O_i) \arrow[r, "\Psi_i"] \arrow[d, "\pi_B"'] & O_i \times \mathcal{F} \arrow[dl, "\pi_{o_i}"] \\ O_i & \end{tikzcd} \tag{2.14}$$

that is, satisfying $\pi_{o_i} \circ \Psi_i = \pi_B$, where π_{o_i} is the projection from $O_i \times \mathcal{F}$ to O_i. The submanifold $\pi_B^{-1}(x)$ is called the fiber at $x \in M$ and is diffeomorphic to $\mathcal{F}$.

We will usually denote a fiber bundle simply by $\pi : B \longrightarrow M$. The morphisms in the category that has as objects fiber bundles are called fiber preserving maps:

Definition 2.9 (Fiber Preserving Maps). Given a smooth map $\varphi : B_1 \longrightarrow B_2$ between two fiber bundles we say that φ is a fiber preserving map iff for any $a, b \in B_1$:

$$\pi_{B_1}(a) = \pi_{B_1}(b) \Rightarrow \pi_{B_2} \circ \varphi(a) = \pi_{B_2} \circ \varphi(b) \tag{2.15}$$

Note that a map $\varphi : B_1 \longrightarrow B_2$ being fiber preserving implies and is implied by the existence of a map $\phi : M_1 \longrightarrow M_2$ making the following diagram commutative:

$$\begin{tikzcd} B_1 \arrow[r, "\varphi"] \arrow[d, "\pi_{B_1}"'] & B_2 \arrow[d, "\pi_{B_2}"] \\ M_1 \arrow[r, "\phi"'] & M_2 \end{tikzcd} \tag{2.16}$$

Given fiber bundles B_1 and B_2 we will say that B_1 is a subbundle of B_2 if the inclusion map $i : B_1 \hookrightarrow B_2$ is fiber preserving.

Given a map $h : M \longrightarrow N$ defined on the base space of a fiber bundle we denote its extension to all of the bundle B by h^e, defined by the following commutative diagram:

$$\begin{tikzcd} B \arrow[d, "\pi_B"'] \arrow[dr, "h^e"] & \\ M \arrow[r, "h"'] & N \end{tikzcd} \tag{2.17}$$

that is $h^e = h \circ \pi_B$. We now consider the extension of a map $H : B \longrightarrow TM$ to a vector field in B. We will define local and global extensions of H. Globally, we define H^e as the set of all vector fields $X \in TB$ such that:

$$\begin{tikzcd} & TB \arrow[d, "T\pi_B"] \\ B \arrow[ur, "X"] \arrow[r, "H"'] & TM \end{tikzcd} \tag{2.18}$$

commutes, that is $T\pi_B(X) = H$. When working locally, one can be more specific and select a distinguished element of H^e, denoted by H^l, which satisfies in trivializing coordinates $T\pi_{\mathcal{F}}(H^l) = 0$, where $\pi_{\mathcal{F}}$ is the projection from $O_i \times \mathcal{F}$ to $\mathcal{F}$. Using trivializing coordinates (x, b) this simply means that $H^l = H\frac{\partial}{\partial x} + 0\frac{\partial}{\partial b}$. A vector field $Y : M \longrightarrow TM$ on the base space M of a fiber bundle can also be extended to a vector field on the whole bundle. It suffices to compose Y with the projection $\pi_B : B \longrightarrow M$ and recover the previous situation since $Y \circ \pi_B$ is a map from B to TM. Given a distribution $\mathcal{D}$ on M we denote by $\mathcal{D}^e$ the extension of $\mathcal{D}$ defined as:

$$\mathcal{D}^e = \bigcup_{X \in \mathcal{D}} X^e \tag{2.19}$$

Note that the previous definitions imply the equality $Ker(Th^e) = (Ker(Th))^e$ since $Ker(Th^e) = Ker(T(h \circ \pi_B)) = Ker(Th \circ T\pi_B) = \{Y \in TB \ : \ T\pi_B(Y) \in Ker(Th)\} = (Ker(Th))^e$.

We recall that a distribution is a smooth assignment of a subbundle of the tangent bundle, that is, at each point $x \in M$ a distribution Δ assigns a linear subspace of T_xM. Given vector fields $X_1, X_2, \ldots, X_n$ such that $Span\{X_1(x), X_2(x), \ldots, X_n(x)\} = \Delta(x)$ for every $x \in M$ we abuse notation and identify Δ with the set of vector fields $\{X_1, X_2, \ldots, X_n\}$. On the cotangent bundle we have similar objects, namely codistributions. A codistribution assigns in a smooth way a subspace of T_x^*M at each $x \in M$. Also in this case we identify a distribution ω with the set of covector fields or one-forms $\{\alpha^1, \alpha^2, \ldots, \alpha^n\}$ when $Span\{\alpha_x^1, \alpha_x^2, \ldots, \alpha_x^n\} = \omega_x$ for every $x \in M$. Given a distribution Δ there is a unique annihilating codistribution ω defining Δ. This codistribution is defined as:

$$\omega = \{\alpha \in T^*M \quad | \quad \alpha(X) = 0 \quad \forall X \in \Delta\} \tag{2.20}$$

Conversely, a codistribution ω defines a unique distribution $Ker(\omega)$ given by the set of all vector fields $X \in TM$ such that $\omega(X) = 0$. If a codistribution ω defines a distribution Δ by annihilation we have that $\Delta = Ker(\omega)$.

Consider for example a unicycle type robot. If we model its state space by the manifold $M = \mathbb{R}^2 \times S^1$, denoting a point in M by (x, y, θ) where x and y represent the position of the robot and θ its orientation we can define its kinematics by a distribution. Consider the following basis for TM:

$$X_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad X_2 = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} \quad X_3 = \begin{bmatrix} -\sin\theta \\ \cos\theta \\ 0 \end{bmatrix} \tag{2.21}$$

With respect to this basis the kinematics is described by the distribution:

$$\Delta = X_1 u_1 + X_2 u_2 \tag{2.22}$$

where $u_1 \in \mathbb{R}$ and $u_2 \in \mathbb{R}$ are control inputs. Equivalently the kinematics is given by the codistribution:

$$\omega = -\sin\theta \mathrm{d}x + \cos\theta \mathrm{d}y \tag{2.23}$$

since any vector field $X \in TM$ such that $\omega(X) = 0$ is of the form (2.22).

Given distributions Δ_1 on M_1 and Δ_2 on M_2 we denote their direct sum $\Delta_1 \oplus \Delta_2$ as the fiber bundle defined pointwise by:

$$(\Delta_1 \oplus \Delta_2)(x_1, x_2) = Ti_1(\Delta_1(x_1)) \oplus Ti_2(\Delta_2(x_2)) \tag{2.24}$$

where $i_1 : M_1 \longrightarrow M_1 \times M_2$ and $i_2 : M_2 \longrightarrow M_1 \times M_2$ are the canonical injections. Note that the direct sum on the right side of (2.24) is performed on the vector space $T_{(x_1,x_2)}(M_1 \times M_2)$.

5. Control Theory

We regard control systems as dynamical systems where *choices* influencing the evolution can be made *during* the evolution. Another interesting and useful interpretation of control systems are families of dynamical systems (or their trajectories if one adopts a behavioral point of view [66]) parameterized by one or more controls. By changing the controls we are changing the dynamical system, and therefore the trajectories or solutions.

Continuous control systems are usually described by differential equations on some manifold M with the choices parameterized by one or more control inputs influencing directly the differential equations. Consider, for example, the simplest mechanical system: a point mass on a line without any potential. If we denote by x the position and by v the velocity we can describe the equations of motion as:

$$\begin{aligned} \dot{v} &= 0 \\ \dot{x} &= v \end{aligned} \tag{2.25}$$

However if we have a mean of exerting a force F on that point mass the equations of motion would change to:

$$\begin{aligned} \dot{v} &= F \\ \dot{x} &= v \end{aligned} \tag{2.26}$$

which can be regarded as a family of differential equations parameterized by F. Changing the value of F will change the solutions of the differential equation.

Resorting to the concepts introduced in Subsection 4 we introduce the notion of control section that is closely related with control systems and which will be fundamental in our study of continuous abstractions:

Definition 2.10 (Control Section). Given a smooth manifold M, a control section on M is a subbundle $\pi_{\mathcal{S}_M} : \mathcal{S}_M \longrightarrow M$ of TM.

We denote by $\mathcal{S}_M(x)$ the set of vectors $X \in T_xM$ such that $X \in \pi_{\mathcal{S}_M}^{-1}(x)$. When we impose more structure on $\mathcal{S}_M$ we recover more familiar objects, such as if to each $x \in M$ we assign a linear subspace of T_xM, then $\mathcal{S}_M$ will be a distribution on M, if on the other hand, we assign an affine subspace then $\mathcal{S}_M$ will be an affine distribution. When $\mathcal{S}_M$ is an affine distribution we may need to refer to the associated distribution denoted by Δ and defined pointwise by:

$$\Delta(x) = \mathcal{S}(x) - \mathcal{S}(x) = \{X \in T_xM \;\;:\;\; X = Y - Z \text{ for some } Y, Z \in \mathcal{S}(x)\} \tag{2.27}$$

Since the early days of control theory it was clear that in order to give a global definition of control systems the notion of input could not be decoupled from the notion of state [13, 88]. The natural mathematical object to consider are fiber bundles:

Definition 2.11 (Control System). A control system $\Sigma_M = (U_M, F_M)$ consists of a fiber bundle $\pi_{U_M} : U_M \longrightarrow M$ called the control bundle and a smooth map $F_M : U_M \longrightarrow TM$ making the following diagram commutative:

$$\begin{tikzcd} U_M \arrow[r, "F_M"] \arrow[d, "\pi_{U_M}"'] & TM \arrow[dl, "\pi_M"] \\ M & \end{tikzcd} \tag{2.28}$$

that is, $\pi_M \circ F_M = \pi_{U_M}$, where $\pi_M : TM \longrightarrow M$ is the tangent bundle projection. Given a control system $\Sigma_M = (U_M, F_M)$, the control section $\mathcal{S}_M \subseteq TM$ of control system Σ_M, is naturally defined pointwise by:

$$\mathcal{S}_M(x) = F_M(\pi_{U_M}^{-1}(x)) \tag{2.29}$$

for all $x \in M$.

The control space U_M is modeled as a fiber bundle since in general the control inputs available may depend on the current state of the system. In local coordinates, Definition 2.11 reduces to the familiar expression $\dot{x} = f(x, u)$ with $u \in \pi_{U_M}^{-1}(x)$. The notion of control section allows us to refer in a concise way to the set of all vectors that belong to the image of F_M by saying that $X \in T_x M$ belongs to $\mathcal{S}_M(x)$ iff there exists a $u \in U_M$ such that $\pi_M(u) = x$ and $F(u) = X$.

We shall call a control system, control affine iff there exists coordinates around each $x \in M$ such that F_M can be written as:

$$F_M = f(x) + \sum_{i=1}^{n} g_i(x) v_i \tag{2.30}$$

where $f(x), g_1(x), g_2(x), \ldots, g_n(x)$ are (locally defined) vector fields and $v_1, v_2, \ldots, v_n$ are control inputs, that is, coordinates for the fiber above x. We also call vector field $f(x)$ the drift and call an affine control system drift-free when $f(x) = 0$. We shall use the expression full nonlinear control system to refer to a nonlinear control system that is not control affine.

Note that the structure of the control section depends on the structure of the control system. For control affine systems we have affine distributions as control sections, if there is no drift we recover distributions as control sections, however, in general, we will have to consider arbitrary control sections on M.

Returning to the example of the point mass moving on the line we see that the state space manifold M is $\mathbb{R}^2$ and the fiber bundle U_M is in fact the trivial bundle $U_M = \mathbb{R}^2 \times \mathbb{R}$. This control system is an example of a control affine system as can be seen by the expression of F_M in coordinates:

$$F_M = f(x, v) + g_1(x, v) v_1 = \begin{bmatrix} 0 \\ v \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} v_1 \tag{2.31}$$

where $v_1 = F \in \mathbb{R}$ is the control input.

A control system can alternatively be defined by a control section $\mathcal{S}_M$ on M in the sense that at each point $x \in M$, $\mathcal{S}_M(x)$ defines all the possible directions along which we can flow or steer our system. Since we will need to work with such control systems in Chapter 3 in a categorical framework we introduce them already using categorical language. Given a control section $\mathcal{S}_M$ there can be several control parameterizations for $\mathcal{S}_M$ and it matters to understand in what sense all those parameterizations represent the same control system. This will be accomplished by giving a categorical definition of control parameterization.

Definition 2.12 (Control Parameterization). Let $\mathcal{S}_M$ be control section on M, $g : TM \longrightarrow N$ and $h : TM \longrightarrow N$ two smooth maps such that $\mathcal{S}_M = \{X \in TM : g(X) = h(X)\}$. A control parameterization for $\mathcal{S}_M$ is a control system (U_M, F_M) such that $g \circ F_M = h \circ F_M$ and for any other control system (U'_M, F'_M) such that $g \circ F'_M = h \circ F'_M$ there exists one and only one fiber-preserving map $\overline{F_M}$ making the following diagram commutative:

$$\begin{tikzcd} U_M \arrow[r, "F_M"] & TM \arrow[r, shift left, "g"] \arrow[r, shift right, "h"'] & N \\ U'_M \arrow[u, "\overline{F_M}"] \arrow[ur, "F'_M"'] & & \end{tikzcd} \tag{2.32}$$

Since the control parameterization was defined through an universal property, any two control parameterizations are isomorphic. It is in this sense that we do not need to distinguish between control systems with the same control section. They are the same control system, up to a change of control coordinates. This will be important when considering the effect of feedback since this change of control coordinates can be regarded as feedback. It is also important to mention that a control parameterization is an equalizer in the category of smooth manifolds.

Having defined control systems the concept of trajectories or solutions of a control system is naturally expressed as:

Definition 2.13 (Trajectories of Control Systems). A curve $c : I \longrightarrow M$, $I \subseteq \mathbb{R}_0^+$ is called a trajectory of control system $\Sigma_M = (U_M, F_M)$, if there exists a curve $c^U : I \longrightarrow U_M$ making the following diagrams commutative:

$$\begin{tikzcd} & U_M \arrow[d, "\pi_{U_M}"] \\ I \arrow[ur, "c^U"] \arrow[r, "c"'] & M \end{tikzcd} \qquad \begin{tikzcd} & U_M \arrow[d, "F_M"] \\ I \arrow[ur, "c^U"] \arrow[r, "Tc"'] & TM \end{tikzcd} \tag{2.33}$$

where we have identified I with TI.

The above commutative diagrams are equivalent to the following equalities:

$$\begin{aligned} \pi_{U_M} \circ c^U &= c \\ Tc &= F_M(c^U) \end{aligned}$$

which mean in local coordinates that $x(t)$ is a trajectory of a control system if there exists an input $u(t)$ such that $x(t)$ satisfies $\dot{x}(t) = f(x(t), u(t))$ and $u(t) \in \pi_{U_M}^{-1}(x(t))$ for all $t \in I$.

CHAPTER 3

A Walk Through the Continuous World

1. Introduction

In the abstracting methodology proposed in [63, 64] it was implicit that certain states might become inputs on the abstracted model. It is perhaps surprising that this abstracting methodology interchanges the role of state and input. However, this fact is the crucial factor that perspectivates a hierarchical control theory. A control design performed on a abstracted model is a control law associated with certain inputs, but these are in fact states of a more detailed model. We can therefore regard a control design as a *specification* for the evolution certain state variables on the more detailed model. In a hierarchical design paradigm those specifications would then be *refined* to obtain a control law that could again be regarded as a specification for a even more detailed model. A complete and thorough understanding of how the states and inputs propagate from models to their abstractions will enable such a hierarchical design scheme. The purpose of this chapter is to give the first steps in this direction. We address the problem of describing the relation between states and inputs of different levels of abstraction. To accomplish this goal we will study quotients of control systems since they capture the notion of abstraction introduced in [63, 64].

We will build on several accumulated results of different authors that in one way or another have made contributions to this problem. One of the first approaches was given in [40] where the analysis of the Lie algebra of a control system lead to a decomposition into smaller systems. In [72], Lie algebraic conditions are formulated for the parallel and cascade decomposition of nonlinear control systems while the feedback version of the same problem was addressed in [56]. A different approach was based on reduction of mechanical systems by symmetries. In [83], symmetries were introduced for mechanical control systems, and further developed in [25] for general control systems. The existence of such symmetries was then used to decompose control systems as the interconnection of lower dimensionality subsystems. The notion of symmetry was further generalized in [57], where it was shown that the existence of symmetries implies that a certain distribution associated with the symmetries was controlled invariant. This related the notion of symmetry with the notion of controlled invariance for nonlinear systems. Controlled invariance [55, 32] was also used to decompose systems into smaller components. A different approach was taken in [50] where it was shown how to study controllability of systems evolving on principle fiber bundles through their projection on the base space. More recently, a modular approach to the modeling of mechanical systems has been proposed in [84], by studying how the interconnection of Hamiltonian control systems can still be regarded as a Hamiltonian control system.

In several of the above approaches, some notion of quotienting is involved. When symmetries exist, one of the blocks of the decompositions introduced in [25] is simply the original control system factored by the action of a Lie group representing the symmetry. If a control system admits a controlled invariant distribution, it is shown in [55, 32] that it has a simpler local representation. This simpler representation can be obtained by factoring the original control system by the equivalence relation defined by considering the leaves of the foliation induced by the controlled invariant distribution, equivalence classes. The notion of abstraction introduced in [64] can also be seen as a quotient since the abstraction is a control system on a smaller dimensional state space defined by an equivalence relation on the state space of the original control system. These facts motivate fundamental questions such as existence and characterization of quotient systems.

In this chapter, we take a new approach to the study of quotients by introducing the category of control systems as the natural setting for such problems in systems theory. The use of category theory for the study of problems in system theory also has a long history which can be traced back to the works of Arbib (see [6] for an introduction). More recently several authors have also adopted a categorical approach as in [45] where the category of affine control system is investigated. We mention also [74], where a categorical approach has been used to provide a general theory of systems.

We define the category of control systems whose objects are fully (non-affine) nonlinear control systems, and morphisms map trajectories between objects. The morphisms in this category extend the notion of ϕ-related systems in [60]. In this categorical setting we formulate the notion of quotient control systems, and show that under mild regularity assumptions on the state and control spaces, quotients always exist. This should be contrasted with several other approaches which rely on the existence of symmetries or controlled invariance to assert the existence of quotients. We also show that the construction proposed in [64] computes quotients up to isomorphism. We introduce the notion of projectable control sections, which will be a fundamental ingredient to characterize the structure of quotients. This notion is in fact equivalent to controlled invariance, and this allows to regard quotients based on symmetries or controlled invariance as a special type of quotients. General quotients, however, are not necessarily induced by symmetries or controlled invariance and have the property that some of their inputs are related to states of the original model. This fact, implicit in [64], is explicitly characterized in this paper by understanding, how the state and input space of the quotient is related to the state and input space of the original control system.

2. Abstractions of Control Bundles

We start by reviewing the abstraction framework developed in [60, 64] and single out the fundamental concepts that will support the desired extension towards control inputs. Then we present a categorical formalization of abstractions based on the notion of simulation and show that abstractions at the level of control bundles are equivalent to the abstraction theory in [63, 64].

2.1. ϕ-related Control Systems. We recall the notion of ϕ-related control systems which is the main pillar of the abstraction theory:

DEFINITION 3.1 (ϕ-related Control Systems). Let Σ_M and Σ_N be two control systems defined on smooth manifolds M and N, respectively. Given a smooth map $\phi : M \longrightarrow N$ we say that Σ_N is ϕ-related to Σ_M iff:

$$T_x\phi(\mathcal{S}_M(x)) \subseteq \mathcal{S}_N \circ \phi(x) \tag{3.1}$$

for every $x \in M$.

In [60] it is shown that this notion, local in nature, is equivalent to a more intuitive and global relation between Σ_M and Σ_N.

PROPOSITION 3.2 ([63]). *Let Σ_M and Σ_N be two control systems defined on smooth manifolds M and N, respectively and let $\phi : M \longrightarrow N$ be a smooth map. Control system Σ_N is ϕ-related to Σ_M iff for every trajectory $c(t)$ of Σ_M, $\phi(c(t))$ is a trajectory of Σ_N.*

Propagating trajectories from a system to another is clearly a desired property. If, in fact, system Σ_N is lower dimensional than system Σ_M, then we are clearly reducing the complexity (dimension) of Σ_M. We can therefore regard Σ_N as an abstraction on Σ_M in the sense that some aspects of Σ_M have been collapsed or abstracted away, while others remain in Σ_N. This motivated a notion of abstraction [60] based on trajectory propagation which defined an abstraction of a control system Σ_M as a ϕ-related control system Σ_N by a surjective submersion ϕ. However this process is described in terms of control sections and the control inputs are not explicitly modeled although they can be implicitly recovered by the algorithms proposed in [60, 64] to compute abstractions.

The idea of sending trajectories from one system to trajectories of another system has been used many times in control theory to study equivalence of control systems. We mention for example linearization by diffeomorphisms [39] or feedback linearization [14, 31, 34]. In these examples the maps ϕ relating the control systems were in fact diffeomorphisms so that no aggregation or abstraction was involved. However the concept of using other maps besides diffeomorphisms for control systems can be traced back to the works of Arbib (see [6] for an introduction) where it is shown that (discrete time) control systems and finite state automata are just different manifestations of the same phenomena.

2.2. Constructing ϕ-related Control Systems. We now recall the construction of ϕ-related control systems given in [64]. We shall restrict ourselves to a purely local treatment without explicit further mention of this fact.

Given an affine control system $\Sigma_M = (U_M, F_M)$ over a smooth manifold M and a smooth surjective submersion $\phi : M \longrightarrow N$, we want to build a new affine control system $\Sigma_N = (U_N, F_N)$ over N that is ϕ-related to Σ_M. We start by realizing that if Σ_M is an affine control system then the control section $\mathcal{S}_M$ is an affine subspace of TM so that it can be written as $\mathcal{S}_M = X_M + \Delta_M$, where X_M is a vector field and Δ_M a linear subspace of TM. We will denote by $\mathcal{K}$ the subbundle of TM given by $\mathcal{K} = Ker(T\phi)$ and note that it is an integrable subbundle in the Frobenius sense whose leaves correspond to points where ϕ is constant. We start by giving a characterization of affine subbundles invariant under a given vector field.

PROPOSITION 3.3 (Invariance of Affine Subbundles [64]). *Let $\mathcal{A} = X + \Delta$ be an affine subbundle of TM and $Y \in TM$ a vector field. $\mathcal{A}$ is invariant under Y iff:*

$$[Y, \mathcal{A}] \subseteq \Delta \tag{3.2}$$

Based on the above proposition we can give a constructive procedure to compute invariant affine subbundles:

DEFINITION 3.4. Let $\mathcal{S}_M = X_M + \Delta_M$ be an affine control section on M. The $\mathcal{K}$-invariant affine control section canonically associated with $\mathcal{S}_M$ is given by:

$$\overline{\mathcal{S}}_M = X_M + L_M \cup [\mathcal{K}, L_M] + [\mathcal{K}[\mathcal{K}, L_M]] + \dots \tag{3.3}$$

with $L_M = \mathcal{K} \cup \Delta_M \cup [\mathcal{K}, X_M]$.

The control section $\overline{\mathcal{S}}_M$ is canonical in the following sense:

PROPOSITION 3.5 ([64]). *The canonical $\mathcal{K}$-invariant affine control section $\overline{\mathcal{S}}_M$ canonically associated with $\mathcal{S}_M$ is the smallest $\mathcal{K}$-invariant affine control section that contains $\mathcal{S}_M$.*

Invariance under $\mathcal{K}$ allows to compute a control section on N as follows:

DEFINITION 3.6 (Canonical construction). Let $\Sigma_M = (U_M, F_M)$ be an affine control system on M with control section:

$$\mathcal{S}_M = X_M + \Delta_M \tag{3.4}$$

and let $\phi : M \longrightarrow N$ be a surjective submersion. The affine control section on N defined by:

$$\mathcal{S}_N(y) = T_x\phi(\overline{\mathcal{S}}_M(x)) \tag{3.5}$$

for any $x \in \phi^{-1}(y)$ is said to be canonically ϕ-related to $\mathcal{S}_M$. Any affine control system $\Sigma_N = (U_N, F_N)$ with control section $\mathcal{S}_N$ is said canonically ϕ-related to Σ_M.

Note that $\mathcal{S}_N$ is well defined since by $\mathcal{K}$-invariance $T_x\phi(\overline{\mathcal{S}}_M(x_1)) = T_x\phi(\overline{\mathcal{S}}_M(x_2))$ for any $x_1, x_2 \in M$ such that $\phi(x_1) = \phi(x_2)$. The control section $\mathcal{S}_N$ on N defines therefore an abstraction of $\mathcal{S}_M$ so that any control system Σ_N with control section $\mathcal{S}_N$ is the desired abstraction. It is also important to mention that in this process there is no explicit construction that allows to compute Σ_N from $\mathcal{S}_N$. The characterization of Σ_N, specially of U_N will be the topic of the remaining chapter.

2.3. From ϕ-related Control Systems to Abstractions of Control Bundles. There are two main motivations to work at the level of control bundles. The first one comes from concrete real problems where often it is necessary to build a hierarchy of different models (abstractions) that would allow to control the system with different levels of detail. A better understanding of how to transform control inputs between different levels of abstraction would allow the design of control laws for the coarser (abstracted) models and then refine then until obtaining control laws for the more detailed control systems. The second reason comes from the following proposition whose proof we delay for now.

PROPOSITION 3.7. *Let Σ_M and Σ_N be two control systems defined on smooth manifolds M and N, respectively and let $\phi : M \longrightarrow N$ be a smooth map. Control system Σ_M is ϕ-related to Σ_N iff there is a fiber-preserving lift of ϕ, denoted by $\varphi : U_M \longrightarrow U_N$ such that:*

$$T_u\varphi(S_M(x)^e) \subseteq (S_N \circ \phi(x))^e \tag{3.6}$$

for every $x \in M$ and $u \in \pi_M^{-1}(x)$.

The above proposition suggests that one should study control systems as dynamical or control systems evolving on the control bundle rather on the base state space. To proceed towards this direction we first introduce the category of control system, denoted by **Con**, which has as objects control systems as described in Definition 2.11. The morphisms in this category extend the concept of ϕ-related control systems described by Definition 3.1. Since the notion of ϕ-related control systems relates control sections and these can be parameterized by controls, the lifted notion should relate sections as well as control bundles.

DEFINITION 3.8 (Morphisms of Control Systems). Let Σ_M and Σ_N be two control systems defined on smooth manifolds M and N, respectively. A morphism f from Σ_M to Σ_N is a pair of maps $f = (\varphi, \phi)$,

$\varphi : U_M \longrightarrow U_N$ and $\phi : M \longrightarrow N$ such that both diagrams:

$$\begin{tikzcd} U_M \arrow[r, "\varphi"] \arrow[d, "\pi_{U_M}"'] & U_N \arrow[d, "\pi_{U_N}"] \\ M \arrow[r, "\phi"'] & N \end{tikzcd} \qquad \begin{tikzcd} U_M \arrow[r, "\varphi"] \arrow[d, "F_M"'] & U_N \arrow[d, "F_N"] \\ TM \arrow[r, "T\phi"'] & TN \end{tikzcd} \tag{3.7}$$

commute.

It will be important for later use to also define isomorphisms:

Definition 3.9 (Isomorphisms of Control Systems). Let Σ_M and Σ_N be two control systems defined on smooth manifolds M and N, respectively. System Σ_M is isomorphic to system Σ_N iff there exist morphisms f_1 from Σ_M to Σ_N and f_2 from Σ_N to Σ_M such that $f_1 \circ f_2 = id_{U_M}$ and $f_2 \circ f_1 = id_{U_N}$.

In this setting, feedback transformations can be seen as special isomorphisms. Consider an isomorphism $f = (\varphi, \phi)$ with $\varphi : U_M \longrightarrow U_M$ such that $\phi = id_M$. In local coordinates (x, u) adapted to the fibers, where x represents the base coordinates (the state) and u the coordinates on the fibers (the inputs), the isomorphism has a coordinate expression for φ of the form $\varphi = (x, \beta(x, u))$. The fiber term $\beta(x, u)$ representing the new control inputs is interpreted as a feedback transformation since it depends on the state at the current location as well as the former inputs u. We shall therefore refer to feedback transformations as isomorphisms over the identity map since we have $\phi = id_M$.

The relation between the notions of ϕ-related control systems (3.1) and **Con** morphisms (3.8) is of equivalence as stated in the next proposition:

Proposition 3.10. *Let Σ_M and Σ_N be two control systems defined on M and N, respectively. Control system Σ_N is ϕ-related to Σ_M iff $f = (\varphi, \phi)$ is a* **Con** *morphism from Σ_M to Σ_N for a fiber preserving lift φ of ϕ.*

Proof. Definition 3.8 trivially implies Definition 3.1 so let us prove that Definition 3.1 implies Definition 3.8. If Σ_N is ϕ-related to Σ_M then by Definition 3.1, $T_x\phi(\mathcal{S}_M(x)) \subseteq \mathcal{S}_N \circ \phi(x)$. But $\mathcal{S}_M$ is parameterized by U_M, so we can regard the map $T\phi \circ F_M : U_M \longrightarrow \mathcal{S}_N \subseteq TN$ (see the diagram below) as a parameterization of $\mathcal{S}_N$ and by definition of control parameterization there is a fiber preserving map μ such that the following diagram

$$\begin{tikzcd} U_M \arrow[r, "\mu"] \arrow[d, "F_M"'] & U_N \arrow[d, "F_N"] \\ TM \arrow[r, "T\phi"] \arrow[d, "\pi_M"'] & TN \arrow[d, "\pi_N"] \\ M \arrow[r, "\phi"'] & N \end{tikzcd}$$

commutes. By taking $\varphi = \mu$, $\pi_{U_M} = \pi_M \circ F_M$ and $\pi_{U_N} = \pi_N \circ F_N$ one recovers Definition 3.8 and the equivalence is proved. □

We now see that if there is a morphism $f = (\varphi, \phi)$ from Σ_M to Σ_N, then this morphism carries trajectories of Σ_M to trajectories of Σ_N in virtue of Proposition 3.2. In this sense Σ_N is also called in the literature a simulation of Σ_M since any trajectory $c_M(t)$ of Σ_M can be simulated by a trajectory $c_N(t) = \phi \circ c_M(t)$ of Σ_N.

We are now in conditions of proving Proposition 3.7 which shows that **Con** morphisms also admit a geometrical characterization at the level of control bundles:

PROPOSITION 3.7. *Let Σ_M and Σ_N be two control systems defined on smooth manifolds M and N, respectively. There exist a* **Con** *morphism $f = (\phi, \varphi)$ from Σ_M to Σ_N iff*

$$T_u\varphi(S_M(x)^e) \subseteq (S_N \circ \phi(x))^e \tag{3.8}$$

for every $x \in M$ and $u \in \pi_M^{-1}(x)$.

PROOF. Assume that Σ_M is ϕ-related to Σ_N and let $c_M(t)$ be a smooth trajectory of Σ_M such that $c_M(0) = x$. By definition of trajectory there is a curve $c_M^U(t)$ on U_M such that $\pi_{U_M} \circ c_M^U = c_M$ and $c_M^U(0) = u$. By ϕ-relatedness the curve $c_N = \phi(c_M)$ is a trajectory of Σ_N implying the existence of a curve c_N^U in U_N such that $\pi_{U_N} \circ c_N^U = c_N$. However Σ_M being ϕ-related to Σ_N implies that there is a **Con** morphism $f = (\phi, \varphi)$ from Σ_M to Σ_N and we have $\varphi(c_M^U) = c_N^U$. By time differentiation at $t = 0$ we get $T_u\varphi(X) = Y$ with $X = \frac{d}{dt}c_M^U(t)|_{t=0}$ and $Y = \frac{d}{dt}c_N^U(t)|_{t=0}$ showing that for any $X \in \mathcal{S}_M(x)^e$ we have $T_u\varphi(X) = Y \in (\mathcal{S}_N \circ \phi(x))^e$ as desired.

Assume now that (3.6) holds. Then, control system defined by control section $\mathcal{S}_M^e$ is φ-related to control system defined by control section $(\mathcal{S}_N \circ \phi)^e$ so that Proposition 3.2 ensures that for every trajectory $c_M^U(t)$ of $\mathcal{S}_M^e$, $\varphi(c_M^U(t)) = c_N^U(t)$ is a trajectory of $(\mathcal{S}_N \circ \phi)^e$. Projecting the equality

$$\varphi(c_M^U(t)) = c_N^U(t) \tag{3.9}$$

on the base space we get

$$\phi(c_M(t)) = c_N(t) \tag{3.10}$$

Time differentiation of (3.10) now gives:

$$\begin{aligned} T\phi \cdot F_M(c_M^U(t)) &= F_N(c_N^U(t)) \\ &= F_N \circ \varphi(c_M^U(t)) \end{aligned} \tag{3.11}$$

where the last equality holds by 3.9. We have thus shown that $T\phi \cdot F_M = F_N \circ \varphi$ since the trajectories c_M^U cover all of U_M. □

The previous proposition tell us that by working at the level of control bundles we can recover more familiar notions such as φ-relatedness of vectors. Besides the clarification that can be gained at the bundle level we will see at next section that we actually need to work at the bundle level when the control sections do not posses enough structure. The previous result can also be related with the notion of extended system described for example in [58]. Instead of considering all possible lifts of $\mathcal{S}_M$ to TU_M as isolated vector fields one can regard that collection of lifts as a control system on U_M. That control system turns out to be the extended control system of Σ_M. We will, however, not explore further this link on this chapter.

3. Quotients of Control Systems

Given a control system Σ_M and an equivalence relation on the manifold M we can regard the quotient control system as an abstraction since some modeling details propagate from Σ_M to the quotient while other modeling details disappear in the factorization process. This fact motivates the study of quotient control systems as they represent lower complexity (dimension) objects that can be used to verify properties of the original control system. Quotients are also important from a design perspective since a control law for the quotient object can be regarded as a specification for the desired behavior of the original control system. In this spirit we will address the following questions:

1. **Existence:** Given a control system Σ_M defined on a manifold M and an equivalence relation $\sim_M$ on M when does there exist a control system on $M/\sim_M$, the quotient manifold, and a fiber preserving lift p_U of the projection $p_M : M \to M/\sim_M$ such that (p_M, p_U) is a **Con** morphism?
2. **Uniqueness:** Is the lift p_U of p_M, when it exists, unique?
3. **Structure of the quotient control bundle:** What is the structure of the quotient control system control bundle?

We remark that the characterization of the quotient control system system map $F : U \to T(M/\sim_M)$ was already solved for the case of control affine systems in [64] where a constructive algorithm for its computation was proposed.

To clarify our discussion we formalize the notion of quotient control systems:

DEFINITION 3.8 (Quotient Control System). Let Σ_L, Σ_M, Σ_N be control systems defined on manifolds L, M and N, respectively and g, h two morphisms from Σ_L to Σ_M. The pair (f, Σ_N) is a quotient control system of Σ_M if $f \circ g = f \circ h$ and for any other pair (f', Σ_N') such that $f' \circ g = f' \circ h$ there exists one and only one morphism $\overline{f}$ from Σ_N to Σ_N' such that the following diagram commutes:

$$\begin{tikzcd} \Sigma_L \arrow[r, shift left, "g"] \arrow[r, shift right, "h"'] & \Sigma_M \arrow[r, "f"] \arrow[dr, "f'"'] & \Sigma_N \arrow[d, "\overline{f}"] \\ & & \Sigma_N' \end{tikzcd} \tag{3.12}$$

that is, $f' = \overline{f} \circ f$.

Intuitively, we can read diagram (3.12) as follows. Assume that the set $\sim = \{(u, v) \in U_M \times U_M \; : \; (u, v) = (g(l), h(l))$ for some $l \in U_L\}$ is a regular equivalence relation [1]. Then, the condition $f \circ g = f \circ h$ simply means that f respects the equivalence relation, that is, $u \sim v \quad \Rightarrow \quad f(u) = f(v)$. Furthermore it asks that for any other map f' respecting relation $\sim$, there exists a unique map $\overline{f}$ such that $f' = \overline{f} \circ f$. This is a usual characterization of quotient manifolds [1] that we here use as a definition. The same chain of reasoning shows that if we replace control systems by the corresponding state space and the morphisms by the maps between the state spaces, then diagram (3.12) asks for N to be also quotient manifold obtained by factoring M by a regular equivalence relation $\sim_M$ on M defined by g and h. The same idea must, therefore, hold for control systems and this means that control system Σ_N must also satisfy a unique factorization property in order to be a quotient control system.

From the above discussion it is clear that a necessary condition for the existence of the quotient control system is the existence of the quotient manifold $M/\sim_M$. When $\sim_M$ is a regular equivalence relation the quotient space $M/\sim_M$ will be a manifold [1] and the equivalence relation can be equivalently described by a surjective submersion. We will, therefore, assume that the regular equivalence relation $\sim_M$ is given by a surjective submersion $\phi : M \longrightarrow N$. Similarly, the fiber preserving lift φ of ϕ will also have to be a surjective submersion.

The first two questions of the previous list are answered in the next theorem which asserts that quotients exist under very moderate conditions:

THEOREM 3.9. *Let Σ_M be a control system on a manifold M and $\phi : M \longrightarrow N$ a surjective submersion. If the distribution $(T\mathcal{S}_M + Ker(TT\phi))/Ker(TT\phi)$ has constant rank, then there exists:*

1. *a control system Σ_N on N,*
2. *a* unique *fiber preserving lift $\varphi : U_M \longrightarrow U_N$ of ϕ such that the pair $((\phi, \varphi), \Sigma_N)$ is a quotient control system of Σ_M.*

PROOF. We start by defining control system Σ_N up to an isomorphism over the identity, that is, we define the control section of Σ_N to be $\mathcal{S}_N = T\phi(\mathcal{S}_M)$. As $\mathcal{S}_M$ is a subbundle of TM we can expand $T\phi(\mathcal{S}_M)$ as:

$$\mathcal{S}_M \overset{i_1}{\hookrightarrow} TM \xrightarrow{T\phi} T\phi(\mathcal{S}_M) = \mathcal{S}_N \overset{i_2}{\hookrightarrow} TN \tag{3.13}$$

It then follows that $TT\phi \circ Ti_1$ has constant rank since $rank(TT\phi \circ Ti_1) = dim(T\mathcal{S}_M) - dim(T\mathcal{S}_M \cap Ker(TT\phi)) = dim((T\mathcal{S}_M + Ker(TT\phi))/Ker(TT\phi))$ which is constant by assumption. Consequently $\mathcal{S}_N$ is a manifold and a fiber bundle over N as $\mathcal{S}_M$ is a fiber bundle over M and $T\phi \circ i_1$ is a fiber preserving map. Finally, it is not difficult to see that i_2 is also fiber preserving therefore making $\mathcal{S}_N$ a subbundle of TN.

We now show that there is a unique fiber preserving lift φ of ϕ such that $f = (\phi, \varphi)$ is a morphism from Σ_M to Σ_N. By definition of $\mathcal{S}_N$ we have $T\phi(\mathcal{S}_M(x)) \subseteq \mathcal{S}_N \circ \phi(x)$ for every $x \in M$. Consequently, the map $T\phi \cdot F_M : U_M \longrightarrow TN$ satisfies $g \circ T\phi \cdot F_M = h \circ T\phi \cdot F_M$ for maps $g : TN \longrightarrow P$ and $h : TN \longrightarrow P$ satisfying $\mathcal{S}_N = \{Y \in TN \ : \ g(Y) = h(Y)\}$. If we now consider any control parameterization (U_N, F_N) for $\mathcal{S}_N$ it follows, by definition of control parameterization, that there exists one and only one fiber preserving map $\overline{F_N} : U_M \longrightarrow U_N$ making diagram 2.32 commutative. It is not difficult to see that this map is the desired $\varphi : U_M \longrightarrow U_N$.

We have thus shown that ϕ defines φ uniquely and that $f = (\phi, \varphi)$ is a morphism. It remains to show that any other morphism $f' = (\phi', \varphi')$ such that ϕ' is compatible with the equivalence relation defined by ϕ factors uniquely through f. We start by recalling that since ϕ is a surjective submersion, ϕ' factors uniquely through ϕ in **Man** [1], that is, there exists one and only one map $\overline{\phi} : N \longrightarrow N'$ such that $\phi' = \overline{\phi} \circ \phi$. From the equality $\phi' = \overline{\phi} \circ \phi$ we conclude:

$$T_x\phi' = T_{\phi(x)}\overline{\phi} \circ T_x\phi \tag{3.14}$$

and it follows that:

$$T_y\overline{\phi}(\mathcal{S}_N(y)) \subseteq \mathcal{S}'_N \circ \overline{\phi}(y) \tag{3.15}$$

since, by definition of $\mathcal{S}_N$, for any $Y \in \mathcal{S}_N(y)$ there is a $X \in \mathcal{S}_M(x)$ such that $\phi(x) = y$ and $T_x\phi \cdot X = Y$, therefore:

$$\begin{aligned} T_y\overline{\phi} \cdot Y &= T_y\overline{\phi} \circ T_x\phi(X) \\ &= T_x\phi'(X) \in \mathcal{S}'_N \circ \phi'(x) = \mathcal{S}'_N \circ \overline{\phi}(y) \end{aligned} \tag{3.16}$$

By the same argument that was used to show that there is a unique fiber preserving lift of ϕ it follows that there is also a unique fiber preserving lift $\overline{\varphi}$ of $\overline{\phi}$ such that $\overline{f} = (\overline{\phi}, \overline{\varphi})$ is a morphism from Σ_N to Σ'_N and $f' = \overline{f} \circ f$. As both $\overline{\phi}$ and $\overline{\varphi}$ are unique so is $\overline{f}$. It remains yet to show that φ' is compatible with the equivalence relation defined by φ, but this is now trivial since the equality $f' = \overline{f} \circ f$ implies:

$$\begin{aligned} \varphi(u) &= \varphi(v) \\ \Rightarrow \overline{\varphi} \circ \varphi(u) &= \overline{\varphi} \circ \varphi(v) \\ \Rightarrow \varphi'(u) &= \varphi'(v) \end{aligned} \tag{3.17}$$

□

This result provides the first characterization of quotient objects in **Con**. It shows that given a regular equivalence relation on the base (state) space of a control system and a mild regularity condition[1], there always exists a quotient control system on the quotient manifold[2]. Furthermore it also shows that the regular equivalence relation on M or the map ϕ uniquely determines a fiber preserving lift φ which describes how pairs state/input of the control system on M relate to the pairs state/input of the quotient control system.

The factorization property expressed in diagram 3.12 allows to show that the constructive algorithm presented in [64] computes quotients of affine control systems up to isomorphism:

Corollary 3.10. *Let Σ_M be an affine control system on a manifold M and $\phi : M \longrightarrow N$ a surjective submersion. The quotient control system computed by the construction presented in [64] based on Σ_M and ϕ is unique up to isomorphism.*

Proof. Let $\overline{\mathcal{S}_N}$ be the control section obtained by the construction proposed in [64] and let $\overline{\overline{\mathcal{S}_N}}$ be the control section defined in the proof of Theorem 3.9, that is $\overline{\overline{\mathcal{S}_N}} \circ \phi = T\phi(\mathcal{S}_M)$. In [64] it is shown that $\overline{\mathcal{S}_N}$ is the smallest control section satisfying:

$$T\phi(\mathcal{S}_M) \subseteq \overline{\mathcal{S}_N} \circ \phi \tag{3.18}$$

As $\overline{\overline{\mathcal{S}_N}}$ also satisfies $T\phi(\mathcal{S}_M) \subseteq \overline{\overline{\mathcal{S}_N}} \circ \phi$ we have $\overline{\mathcal{S}_N} \subseteq \overline{\overline{\mathcal{S}_N}}$. However, by (3.18) we have $T\phi(\mathcal{S}_M) = \overline{\overline{\mathcal{S}_N}} \circ \phi \subseteq \overline{\mathcal{S}}_N \circ \phi \Rightarrow \overline{\overline{\mathcal{S}_N}} \subseteq \overline{\mathcal{S}}_N$ by surjectivity of ϕ and consequently $\overline{\mathcal{S}_N} = \overline{\overline{\mathcal{S}_N}}$. Theorem 3.9 and in particular commutativity of diagram 3.12 now imply that $\overline{\mathcal{S}_N}$ is unique up to isomorphism. □

Having answered the first two questions from the previous list, we concentrate on the characterization of the quotient control bundle. This problem requires a deeper understanding of how ϕ determines φ and will be the goal of the remaining paper. Since **Con** was defined over **Man**, that is morphisms in **Con** are smooth maps and control systems are defined on manifolds and fiber bundles, the characterization of φ will require an interplay of tools from differential geometry and category theory.

[1]The constant rank condition on $(Ker(TT\phi) + T\mathcal{S}_M)/Ker(TT\phi)$ is only required to ensure that $\mathcal{S}_N$ is a manifold. If one does not require a control section to be a manifold, then this condition can be weakened.

[2]This fact can be put in a more general context by introducing a forgetful functor from **Con** to **Man** that associates with each control system Σ_M defined over M the manifold M and to each morphism from Σ_M to Σ_N the map ϕ. In this context the previous result assumes the form of a universal arrow for this functor.

4. Projectable Control Sections

We now extend the notion of projectable vector fields from [49] and of projectable families of vector fields from [50] to control sections. The notion of projectable control sections is weaker then projectable vector field or families of vector fields but nonetheless stronger than **Con** morphisms. The motivation for introducing this notion comes from the fact that projectability of control sections will be a fundamental ingredient in characterizing the structure of the quotient control bundle. Furthermore, we will also see that projectability, as defined in this categorical setting, will correspond to the well known notion of controlled invariance.

Given a vector field X on M and a surjective submersion $\phi : M \longrightarrow N$ we say that X is projectable with respect to ϕ when $Y = T\phi \cdot X$, the projection of X, is a well defined vector field on N that satisfies $T\phi \cdot X = Y \circ \phi$ [49]. The vector field Y is also called ϕ-related to X [1]. This notion was extended to families of vector fields in [50] by requiring that the projection of each vector field in the family is a well defined vector field on N. However, when working with control sections, which can be regarded as *sets* of vectors at each base point $x \in M$, one should only require that the projection of these *sets* of vectors is the same *set* when the base points on M project on the same base point on N. This is formalized as follows:

Definition 3.11. Let M be a manifold, $\mathcal{S}_M$ a control section on M and $\phi : M \longrightarrow N$ a surjective submersion. We say that $\mathcal{S}_M$ is projectable with respect to ϕ iff $\mathcal{S}_M$ induces a control section $\mathcal{S}_N$ on N such that the following diagram commutes:

$$\begin{tikzcd} \mathcal{P}(TM) \arrow[r, "T\phi"] & \mathcal{P}(TN) \\ M \arrow[u, "\mathcal{S}_M"] \arrow[r, "\phi"'] & N \arrow[u, "\mathcal{S}_N"'] \end{tikzcd} \tag{3.19}$$

We see that if $\mathcal{S}_M$ is in fact a vector field we recover the notion of projectable vector fields. The notion of projectable control sections is stronger then the notion of **Con** morphism since for any $x_1, x_2 \in M$ such that $\phi(x_1) = \phi(x_2)$ we necessarily have $T\phi(\mathcal{S}_M(x_1)) = \mathcal{S}_N \circ \phi(x_1) = T\phi(\mathcal{S}_M(x_2))$ if $\mathcal{S}_M$ is projectable. On the other hand, if (ϕ, φ) is a **Con** morphism for a fiber preserving lift φ of ϕ, we only have the inclusions $T\phi(\mathcal{S}_M(x_1)) \subseteq \mathcal{S}_N \circ \phi(x_1)$ and $T\phi(\mathcal{S}_M(x_2)) \subseteq \mathcal{S}_N \circ \phi(x_1)$. Therefore projectability with respect to ϕ implies that ϕ can be extended to a **Con** morphism but given a **Con** morphism $f = (\phi, \varphi)$ from Σ_M to Σ_N it is not true, in general, that $\mathcal{S}_M$ is projectable with respect to ϕ.

To determine the relevant conditions on $\mathcal{S}_M$ that ensure projectability we will need an auxiliary result:

PROPOSITION 3.12. *Let $f : M \longrightarrow N$ be a map between manifolds and let X_t be the flow of a vector field $X \in TM$ such that $f \circ X_t = f$. Then the following equality holds for every $x \in M$:*

$$T_x f \, T_{X_t(x)} X_{-t} = T_{X_t(x)} f \tag{3.20}$$

PROOF. The equality $f \circ X_t = f$ is equivalent to:

$$\begin{aligned} & f \circ X_t(x) = f(x) \\ \Leftrightarrow \quad & f(X_t(x)) = f \circ (X_t)^{-1} \circ X_t(x) \\ \Leftrightarrow \quad & f(X_t(x)) = f \circ X_{-t}(X_t(x)) \end{aligned} \tag{3.21}$$

and by differentiation of the previous expression we arrive at the desired equality:

$$T_{X_t(x)} f = T_x f \, T_{X_t(x)} X_{-t} \tag{3.22}$$

□

We can now give sufficient and necessary conditions for projectability of control sections.

PROPOSITION 3.13 (Projectable Control Sections). *Let M be a manifold, $\mathcal{S}_M$ a control section on M and $\phi : M \longrightarrow N$ a surjective submersion. Given any control parameterization (U_M, F_M) of $\mathcal{S}_M$ and any $\overline{F_M} \in F_M^e$, $\mathcal{S}_M$ is projectable with respect to ϕ iff:*

$$[\overline{F_M}, Ker(T\phi^e)] \subseteq Ker(T\phi^e) + [\overline{F_M}, 0^e] \tag{3.23}$$

where $0^e = T\pi_{U_M}^{-1}(0)$.

PROOF. We show necessity first. Assume that diagram (3.19) commutes. Then we have:

$$T_x\phi(\mathcal{S}_M(x)) = T_{x'}\phi(\mathcal{S}_M(x')) \tag{3.24}$$

for all $x, x' \in M$ such that $\phi(x) = \phi(x')$, that is, for any x and x' on the same leaf of the foliation induced by $Ker(T\phi)$. If we denote by K_t the flow of any vector field $K \in Ker(T\phi^e)$, expression (3.24) implies that:

$$T_{\pi_{U_M} \circ K_t(u)}\phi(F_M \circ K_t(u)) \in T_x\phi(\mathcal{S}_M(x)) \tag{3.25}$$

for every $t \in \mathbb{R}$ such that K_t is defined and for every $u \in \pi_{U_M}^{-1}(x)$. Since the left hand side of (3.25) belongs to the right hand side we can always find a $Y \in 0^e$ such that its flow Y_t will parameterize the image of the left hand side, that is:

$$T_{\pi_{U_M} \circ K_t(u)}\phi(F_M \circ K_t(u)) = T_{\pi_{U_M} \circ Y_t(u)}\phi(F_M \circ Y_t(u)) \tag{3.26}$$

The previous equality implies that for any $\overline{F_M} \in F_M^e$ we have:

$$T_{K_t(u)}\phi^e(\overline{F_M} \circ K_t(u)) = T_{Y_t(u)}\phi^e(\overline{F_M} \circ Y_t(u)) \tag{3.27}$$

however, the equalities $\phi^e \circ K_t = K_t$, $\phi^e \circ Y_t = \phi^e$ and Proposition 3.12 allow to rewrite (3.27) as:

$$
\begin{aligned}
T_u\phi^e(T_{K_t(u)}K_{-t} \circ \overline{F_M} \circ K_t(u)) &= T_u\phi^e(T_{Y_t(u)}Y_{-t} \circ \overline{F_M} \circ Y_t(u)) \\
\Leftrightarrow \quad T_u\phi^e(K_t(u)^*\overline{F_M}) &= T_u\phi^e(Y_t(u)^*\overline{F_M})
\end{aligned}
\tag{3.28}
$$

Time differentiation at $t = 0$ now implies:

$$
\begin{aligned}
T_u\phi^e([K(u), \overline{F_M}(u)]) &= T_u\phi^e([Y(u), \overline{F_M}(u)]) \\
\Rightarrow [K, \overline{F_M}] &\in [Y, \overline{F_M}] + Ker(T\phi^e)
\end{aligned}
\tag{3.29}
$$

which trivially implies inclusion 3.23.

To show sufficiency we use a similar argument. Assume that (3.23) holds, then for any $K \in Ker(T\phi^e)$ there exists a $Y \in 0^e$ such that:

$$
\begin{aligned}
& T_u\phi^e([\overline{F_M}(u), K(u)]) = T_u\phi^e([\overline{F_M}(u), Y(u)]) \\
\Leftrightarrow \quad & T_u\phi^e([\overline{F_M}(u), K(u) - Y(u)]) = 0
\end{aligned}
\tag{3.30}
$$

Consider now the regular and involutive distribution $Ker(T\phi^e)$. Involutivity and regularity imply that $Z_t^* W \in Ker(T\phi^e)$ for any $W \in Ker(T\phi^e)$ and the flow Z_t of any vector field $Z \in Ker(T\phi^e)$ [76]. Since $K \in Ker(T\phi^e)$ and $Y \in Ker(T\phi^e)$ it follows that $K - Y \in Ker(T\phi^e)$, but from (3.30), $[\overline{F_M}, K - Y]$ also belongs to $Ker(T\phi^e)$ so that we conclude:

$$
T_u\phi^e((K - Y)_t(u)^*[\overline{F_M}, K - Y]) = 0 \tag{3.31}
$$

where $(K - Y)_t$ denotes the flow of the vector field $K - Y$. However, the previous expression is equivalent to:

$$
\begin{aligned}
& T_u\phi^e(\tfrac{d}{dt}(K - Y)_t(u)^*\overline{F_M}) = 0 \\
\Leftrightarrow \quad & \tfrac{d}{dt}T_u\phi^e((K - Y)_t(u)^*\overline{F_M}) = 0
\end{aligned}
\tag{3.32}
$$

where the last equality follows from the fact that $T\phi$ is a linear map. Since the time derivative is zero, we must have:

$$
T_u\phi^e((K - Y)_t(u)^*\overline{F_M}) = T_u\phi^e((K - Y)_0(u)^*\overline{F_M}) = T_u\phi^e(\overline{F_M}(u)) \tag{3.33}
$$

From the equality $\phi^e = \phi^e \circ (K - Y)_t$ we conclude that $T_u\phi^e\, T_{(K-Y)_t(u)}(K - Y)_{-t} = T_{(K-Y)_t(u)}\phi^e$ by Proposition 3.12 so that (3.33) can be written as:

$$
T_{(K-Y)_t(u)}\phi^e(\overline{F_M} \circ (K - Y)_t(u)) = T_u\phi^e(\overline{F_M}(u)) \tag{3.34}
$$

and projecting on TM we get:

$$
T_{\pi_{U_M}(K_t'(u))}\phi(F_M \circ (K')_t(u)) = T_x\phi(F_M(u)) \tag{3.35}
$$

with $K' = K - Y$. This equality shows that for any $X \in \mathcal{S}_M(x)$, $T_x\phi \cdot X \in T_{x'}\phi(\mathcal{S}_M(x'))$, therefore $T_x\phi(\mathcal{S}_M(x)) \subseteq T_{x'}\phi(\mathcal{S}_M(x'))$. However, replacing x by x' and K by $-K$ on (3.35) we get $T_{x'}\phi(\mathcal{S}_M(x')) \subseteq T_x\phi(\mathcal{S}_M(x))$ so that we conclude the equality:

$$T_x\phi(\mathcal{S}_M(x)) = T_{x'}\phi(\mathcal{S}_M(x')) \tag{3.36}$$

Since any point x'' satisfying $\phi(x'') = \phi(x)$ can be reached by a concatenation of flows induced by vector fields in $Ker(T\phi)$, transitivity of equality between sets implies that (3.36) holds for any two points $x, x' \in M$ such that $\phi(x) = \phi(x')$ from which commutativity of diagram (3.19) readily follows. □

It is interesting to note that if we assume some structure on $\mathcal{S}_M$ we can give conditions for projectability without explicitly mentioning the control parameterization. This is the case for control affine systems where the affine structure on $\mathcal{S}_M$ allows to simplify expression (3.23) as follows:

COROLLARY 3.14. *Let M be a manifold, $\mathcal{A}_M$ an affine distribution on M and $\phi : M \longrightarrow N$ a surjective submersion. $\mathcal{A}_M$ is projectable with respect to ϕ iff:*

$$[\mathcal{A}_M, Ker(T\phi)] \subseteq Ker(T\phi) + \Delta_M \tag{3.37}$$

where Δ_M is the distribution associated to $\mathcal{A}_M$.

By now it is already clear that projectability and local controlled invariance are equivalent concepts. We recall the notion of locally controlled invariant distribution:

DEFINITION 3.15 (Locally Controlled Invariant Distributions [58]). Let $\Sigma_M = (U_M, F_M)$ be a control system over a manifold M and let $\mathcal{D}$ be a distribution on M. The distribution $\mathcal{D}$ is locally controlled invariant for F_M if for every $x \in M$ there is an open set $O \subseteq M$, containing x and a local (feedback) isomorphism over the identity such that in trivializing coordinates (x, v) the new control system $F'_M = F_M \circ \alpha$ satisfies:

$$[F'_M(x, v), \mathcal{D}(x)] \subseteq \mathcal{D}(x) \tag{3.38}$$

for every (x, v) in the domain of α.

If a control section is projectable then locally we can always chose $\overline{F_M} = F^l_M$ and therefore recover the conditions for local controlled invariance from [24]:

THEOREM 3.16 ([24]). *Let Σ_M be a control system over a manifold M and $\phi : M \longrightarrow N$ a surjective submersion. The distribution $Ker(T\phi)$ is locally controlled invariant for F_M iff $\mathcal{S}_M$ is projectable with respect to ϕ.*

From the study of symmetries of nonlinear control systems [25, 57] it was already known that the existence of symmetries or partial symmetries implies controlled invariance of a certain distribution associated with the symmetries. This shows that control systems that are projectable comprise quotients by symmetry and controlled invariance. However there are also quotients for which projectability does not hold as we describe in the next section.

5. The Structure of Quotient Control Systems

We have already seen that the notion of **Con** morphisms generalizes the notion of projectable control sections. This shows that it is possible to quotient control systems whose control sections are not projectable. In this situation the map φ and the control bundle of the quotient control system will be significantly different from the projectable case. To understand this difference we start characterizing the fiber preserving lift φ of ϕ. Recall that if $f = (\phi, \varphi)$ is a morphism from Σ_M to Σ_N we have the following commutative diagram:

$$\begin{tikzcd} U_M \arrow[r, "\varphi"] \arrow[d, "F_M"'] & U_N \arrow[d, "F_N"] \\ TM \arrow[r, "T\phi"'] & TN \end{tikzcd} \tag{3.39}$$

Since φ is a surjective submersion we know that U_N is diffeomorphic to $U_M/\sim$, where $\sim$ is the regular equivalence relation induced by φ. This means that to understand the structure of U_N it is enough to determine the regular and involutive distribution on U_M given by $Ker(T\varphi)$. However the map φ is completely unknown, so we will resort to the elements that are available, namely F_M and ϕ to determine $Ker(T\varphi)$. Differentiating[3] diagram (3.39) we get:

$$\begin{tikzcd} TU_M \arrow[r, "T\varphi"] \arrow[d, "TF_M"'] & TU_N \arrow[d, "TF_N"] \\ TTM \arrow[r, "TT\phi"'] & TTN \end{tikzcd} \tag{3.40}$$

from which we conclude:

$$Ker(TT\phi \circ TF_M) = Ker(TF_N \circ T\varphi) = Ker(T\varphi) \tag{3.41}$$

where the last equality holds since F_N is an immersion by definition of control parameterization. We can now attempt to understand what is factored away and what is propagated from U_M to U_N since $Ker(T\varphi)$ is expressible in terms of F_M and ϕ. The first step is to clarify the relation between $Ker(T\varphi)$

[3]The operator sending manifolds to their tangent manifolds and maps to their tangent maps is an endofunctor on **Man**, also called the tangent functor [38].

and $Ker(T\phi)$. Since φ is a fiber preserving lift of ϕ the following diagram commutes:

$$\begin{tikzcd} TU_M \arrow[r, "T\varphi"] \arrow[d, "T\pi_{U_M}"'] & TU_N \arrow[d, "T\pi_{U_N}"] \\ TM \arrow[r, "T\phi"'] & TN \end{tikzcd} \tag{3.42}$$

which implies that:

$$T\pi_{U_M}(Ker(T\varphi)) \subseteq Ker(T\phi) \tag{3.43}$$

However this only tell us that the reduction on M due to ϕ cannot be "smaller" than the reduction on the base space of U_M due to φ. This leads to the interesting phenomena which occurs when, for *e.g.* :

$$T\pi_{U_M}(Ker(T\varphi)) = \{0\} \subseteq Ker(T\phi) \tag{3.44}$$

The above expression implies that the base space of U_M is not reduced by φ. However, U_N is a fiber bundle with base space N and therefore the points reduced by ϕ must necessarily lift to the fibers of U_N. This will not happen if we can ensure the existence of a distribution $\mathcal{D} \subseteq Ker(T\varphi)$ such that $T\pi_{U_M}(\mathcal{D}) = Ker(T\phi)$. The existence of such a distribution turns out to be related with projectability as asserted in the next proposition:

PROPOSITION 3.17. *Let $\Sigma_M = (U_M, F_M)$ be a control system over a manifold M, $\phi : M \longrightarrow N$ a surjective submersion and $\varphi : U_M \longrightarrow U_N$ a fiber preserving lift of ϕ. There exists a regular distribution $\mathcal{D}$ on U_M satisfying $\mathcal{D} \subseteq Ker(T\varphi)$ and $T\pi_{U_M}(\mathcal{D}) = Ker(T\phi)$ iff $\mathcal{S}_M$ is projectable with respect to ϕ.*

PROOF. We start by showing that projectability implies the existence of $\mathcal{D}$. If $\mathcal{S}_M$ is projectable with respect to ϕ then for every $x, x' \in M$ such that $\phi(x) = \phi(x')$ we have that $T_x\phi(\mathcal{S}_M(x)) = T_{x'}\phi(\mathcal{S}_M(x'))$. This means that for any $x \in M$, $u \in \pi_{U_M}^{-1}(x)$ and $X \in Ker(T\phi^e)$ there exists a $Y \in 0^e$ such that:

$$T_{\pi_{U_M} \circ X_t(u)}\phi(F_M \circ X_t(u)) = T_x\phi(F_M \circ Y_t(u)) \tag{3.45}$$

for all $t \in \mathbb{R}$ such that the flows X_t and Y_t of X and Y are defined. Considering now $T\phi$ as a map between the manifolds TM and TN, the time derivative of $T_{\alpha(t)}\phi(\beta(t))$ for $(\alpha, \beta) : \mathbb{R} \longrightarrow TM$ provides $T_{(\alpha(t),\beta(t))}T_{\alpha(t)}\phi(T\beta(t))$. The same considerations applied to (3.45) at $t = 0$ give:

$$T_{(x,F_M(u))}T_x\phi \circ T_uF_M(X(u)) = T_{(x,F_M(u))}T_x\phi \circ T_uF_M(Y(u)) \tag{3.46}$$

which we rewrite as:

$$T_{(x,F_M(u))}T_x\phi \circ T_uF_M(X(u) - Y(u)) = 0 \tag{3.47}$$

by linearity of the involved maps. Since (3.47) is true for any $X \in Ker(T\phi^e)$ we can define the distribution:

$$\mathcal{D} = \bigcup_{K \in Ker(T\phi)} \{Z = X - Y \ : \ X \in K^e \ \wedge \ Y \in 0^e \text{ is such that (3.47) holds}\} \tag{3.48}$$

This distribution clearly satisfies:

$$TT\phi \circ TF_M(\mathcal{D}) = \{0\} \quad \Leftrightarrow \quad \mathcal{D} \in Ker(T\varphi) \tag{3.49}$$

is regular since $dim(\mathcal{D}) = dim(Ker(T\phi))$ by construction, satisfies $T\pi_{U_M}(\mathcal{D}) = Ker(T\phi)$ also by construction and is therefore the desired distribution.

The converse is proved as follows. Assume the existence of the distribution $\mathcal{D}$, then $\mathcal{D} \subseteq Ker(T\varphi)$ is equivalent to:

$$TT\phi \circ TF_M(\mathcal{D}) = \{0\} \tag{3.50}$$

Let $Z \in \mathcal{D}$ and denote by Z_t the flow of Z. Expression (3.50) implies that:

$$\frac{d}{dt}\bigg|_{t=0} T_{\pi_{U_M} \circ Z_t(u)}\phi(F_M \circ Z_t(u)) = 0 \quad \Rightarrow \quad \frac{d}{dt}\bigg|_{t=0} T_{Z_t(u)}\phi^e(\overline{F_M} \circ Z_t(u)) = 0 \tag{3.51}$$

for any $\overline{F_M} \in F_M^e$ and for all $t \in \mathbb{R}$ such that Z_t is defined.

Noticing that $Z \in \mathcal{D} \subseteq Ker(T\varphi)$ implies $\varphi = \varphi \circ Z_t$ (since φ is constant on the leaves of the foliation induced by $Ker(T\varphi)$) and $\pi_{U_N} \circ \varphi = \phi \circ \pi_{U_M}$ by commutativity of diagram 4.29, we conclude that ϕ^e is also Z_t invariant:

$$\phi^e \circ Z_t = (\phi \circ \pi_{U_M}) \circ Z_t = (\pi_{U_N} \circ \varphi) \circ Z_t = \pi_{U_N} \circ \varphi = \phi \circ \pi_{U_M} = \phi^e \tag{3.52}$$

Proposition 3.12 now ensures that:

$$T_{Z_t(u)}\phi^e = T_u\phi^e \circ T_{Z_t(u)}Z_{-t} \tag{3.53}$$

and expression (3.53) allows to rewrite (3.51) as:

$$\begin{aligned} \frac{d}{dt}\bigg|_{t=0} T_{Z_t(u)}\phi^e(\overline{F_M} \circ Z_t(u)) = 0 \quad & \Leftrightarrow \quad \frac{d}{dt}\bigg|_{t=0} T_u\phi^e(T_{Z_t(u)}Z_{-t} \circ \overline{F_M} \circ Z_t(u)) = 0 \\ & \Leftrightarrow \quad \frac{d}{dt}\bigg|_{t=0} T_u\phi^e(Z_t(u)^*\overline{F_M}) = 0 \\ & \Leftrightarrow \quad T_u\phi^e([Z(u), \overline{F_M}(u)]) = 0 \end{aligned} \tag{3.54}$$

or equivalently $[Z, \overline{F_M}] \in Ker(T\phi^e)$. Since Z is any vector field in $Ker(T\phi^e)$ it follows that $[\overline{F_M}, Ker(T\phi^e)] \subseteq Ker(T\phi^e)$ which by Proposition 3.13 implies that $\mathcal{S}_M$ is projectable with respect to ϕ as desired. □

From the proof of the previous proposition it is clear that if $\mathcal{D}$ is locally of the form $\mathcal{D} = Ker(T\phi)^l$ then we can replace projectability by the more restrictive notion of invariance:

COROLLARY 3.18. *Let Σ_M be a control system over a manifold M, $\phi : M \longrightarrow N$ a surjective submersion and $\varphi : U_M \longrightarrow U_N$ a fiber preserving lift of ϕ. The locally defined distribution $Ker(T\phi)^l$ satisfies*

$Ker(T\phi)^l \subseteq Ker(T\varphi)$ iff $Ker(T\phi)^l$ is invariant for F_M^l, that is, iff:

$$[F_M^l(u), Ker(T\phi)^l(u)] \subseteq Ker(T\phi)^l(u) \tag{3.55}$$

for every u such that $Ker(T\phi)^l$ is defined.

Proposition 3.17 shows that projectability characterizes the structure of the quotient control system in the sense that states lift to the fibers when the control section is *not* projectable. However we can be a little more detailed in our analysis and try to determine if the fibers of U_M are reduced or if the fibers of U_M are in fact diffeomorphic to the fibers of U_N and reduction takes place only on the base space. The answer is given in the next proposition:

PROPOSITION 3.19. *Let $\Sigma_M = (U_M, F_M)$ be a control system over a manifold M, $\phi : M \longrightarrow N$ a surjective submersion, $\varphi : U_M \longrightarrow U_N$ a fiber preserving lift of ϕ and $\overline{F_M}$ any vector field in F_M^e. A regular and involutive distribution $\mathcal{E}$ on U_M such that $T\pi_{U_M}(\mathcal{E}) = \{0\}$ satisfies $\mathcal{E} \subseteq Ker(T\varphi)$ iff:*

$$[\overline{F_M}, \mathcal{E}] \subseteq Ker(T\phi^e) \tag{3.56}$$

PROOF. Assume that the distribution $\mathcal{E}$ belongs to $Ker(T\varphi)$, then following an argument similar to the proof of Proposition 3.17 shows that $[\overline{F_M}, \mathcal{E}] \subseteq Ker(T\phi^e)$.

Conversely assume that $[\overline{F_M}, \mathcal{E}] \subseteq Ker(T\phi^e)$ and let $X \in \mathcal{E}$. Then, the equality:

$$T\phi^e([\overline{F_M}, X]) = 0 \tag{3.57}$$

holds. However this expression is equivalent to:

$$\begin{aligned} T_u\phi^e([\overline{F_M}(u), X(u)]) = 0 \quad &\Leftrightarrow \quad \left.\frac{d}{dt}\right|_{t=0} T_u\phi^e(X_t(u)^*\overline{F_M}) = 0 \\ &\Leftrightarrow \quad \left.\frac{d}{dt}\right|_{t=0} T_{X_t(u)}\phi^e(\overline{F_M} \circ X_t(u)) \end{aligned} \tag{3.58}$$

where the last equality is a consequence of $\phi^e \circ X_t = \phi^e$ and Proposition 3.12. Projection on TM gives:

$$\left.\frac{d}{dt}\right|_{t=0} T_{\pi_{U_M} \circ X_t(u)}\phi(F_M \circ X_t(u)) = 0 \tag{3.59}$$

which also equals:

$$T_{(x, F_M(u))}T_x\phi \circ T_u F_M(X(u)) = 0 \tag{3.60}$$

therefore implying that $X \in Ker(T\varphi)$ and consequently $\mathcal{E} \subseteq Ker(T\varphi)$. □

Collecting the results given by Propositions 3.17 and 3.19 we can now characterize both φ and U_N. Intuitively, we will use projectability to determine if the standard fiber of the quotient control bundle will receive states from M and Proposition 3.19 to characterize the amount of reduction induced by φ.

THEOREM 3.20 (Structure of Control Systems Quotients). *Consider a control system $\Sigma_M = (U_M, F_M)$ over a manifold M, $(f, \Sigma_N) = ((\phi, \varphi), (U_N, F_N))$ a quotient of Σ_M, and any vector field $\overline{F_M}$ in F_M^e. Let*

$\mathcal{E}$ be the involutive distribution defined by $\mathcal{E} = \{X \in 0^e \ : \ [\overline{F_M}, X] \in Ker(T\phi^e)\}$, which we assume to be regular, and denote by $R_{\mathcal{E}}$ the regular equivalence relation induced by $\mathcal{E}$. Under these assumptions:

1. **Reduction from states to states and no reduction on inputs**
   Fiber bundle U_N has base space diffeomorphic to N, and standard fiber $\mathcal{F}_N$ diffeomorphic to $\mathcal{F}_M$ iff:
   - *$\mathcal{S}_M$ is projectable with respect to ϕ;*
   - *$\mathcal{E} = \{0\}$.*
2. **Reduction from states to states and from inputs to inputs**
   Fiber bundle U_N has base space diffeomorphic to N, and standard fiber $\mathcal{F}_N$ diffeomorphic to $\mathcal{F}_M/R_{\mathcal{E}}$ iff:
   - *$\mathcal{S}_M$ is projectable with respect to ϕ;*
   - *$\mathcal{E} \neq \{0\}$.*
3. **Reduction from states to inputs and no reduction on inputs**
   Fiber bundle U_N has base space diffeomorphic to N, and standard fiber $\mathcal{F}_N$ diffeomorphic to $\mathcal{F}_M \times \mathcal{K}$ iff:
   - *$[\overline{F_M}, Ker(T\phi^e)] \cap \left(Ker(T\phi^e) + [\overline{F_M}, 0^e]\right) = \{0\}$;*
   - *$[\overline{F_M}, Ker(T\phi^e)] \neq \{0\}$;*
   - *$\mathcal{E} = \{0\}$.*
4. **Reduction from states to inputs and from inputs to inputs**
   Fiber bundle U_N has base space diffeomorphic to N, and standard fiber $\mathcal{F}_N$ diffeomorphic to $(\mathcal{F}_M/R_{\mathcal{E}}) \times \mathcal{K}$ iff:
   - *$[\overline{F_M}, Ker(T\phi^e)] \cap \left(Ker(T\phi^e) + [\overline{F_M}, 0^e]\right) = \{0\}$;*
   - *$[\overline{F_M}, Ker(T\phi^e)] \neq \{0\}$;*
   - *$\mathcal{E} \neq \{0\}$.*

where $\mathcal{K}$ is any leaf of the foliation on M induced by the distribution $Ker(T\phi)$.

PROOF. We will follow the enumeration of the theorem.

1. By definition of fiber bundle the fibers of U_N are diffeomorphic so that it suffices to show that the fiber at some point $y \in N$ has the desired structure. Let x be a point in M, since $\mathcal{S}_M$ is projectable it follows from Theorem 3.16 the existence of an open set O^x in M, containing x and a local isomorphism over the identity $\alpha : O_U^x \to O_U^x$, with $O_U^x = \pi_{U_M}^{-1}(O^x)$, such that $[(F_M \circ \alpha)^l, Ker(T\phi)^l] \subseteq Ker(T\phi)^l$. Invoking Corollary 3.18 we see that $Ker(T\phi)^l \subseteq Ker(T(\varphi \circ \alpha))$ however, by assumption, $\mathcal{E} = \{0\}$ so that dimension counting implies that $Ker(T\phi)^l = Ker(T(\varphi \circ \alpha))$. We thus have the following local situation, by shrinking O^x if necessary: $O_U^x \cong O^x \times \mathcal{F}_M$ and $Ker(T(\varphi \circ \alpha)) = Ker(T\phi) \times \{0\}$. Since $\varphi \circ \alpha$ is a submersion it follows that $\varphi \circ \alpha(O_U^x) \cong O_U^x / R_K$,

where R_K is the regular equivalence relation induced by $Ker(T\phi) \times \{0\}$. However O_U^x being diffeomorphic to $O^x \times \mathcal{F}_M$ implies that $O_U^x/R_K \cong (O^x \times \mathcal{F}_M)/R_K \cong \phi(O^x) \times \mathcal{F}_M$, which shows that the standard fiber over every $y \in \phi(O^x)$ is diffeomorphic to $\mathcal{F}_M$.

Conversely if $\mathcal{F}_M$ is diffeomorphic to $\mathcal{F}_N$ there does not exist a distribution $\mathcal{E} \subseteq Ker(T\varphi)$ such that $T\pi_{U_M}(\mathcal{E}) = 0$, which by Proposition 3.19 implies that $\mathcal{E} = \{0\}$. Since no states lift into the fibers of U_N there exists a distribution $\mathcal{D} \subseteq Ker(T\phi)$ such that $T\pi_{U_M}(\mathcal{D}) = Ker(T\phi)$ which by Proposition 3.17 is equivalent to projectability of $\mathcal{S}_M$ with respect to ϕ.

2. As in item 1 there exists a local isomorphism $\alpha : O_U^x \longrightarrow O_U^x$ such that $Ker(T\phi)^l \subseteq Ker(T(\phi \circ \alpha))$. Since α is an isomorphism over the identity all the vector fields $X \in Ker(T\varphi)$ such that $T\pi_{U_M}(X) = 0$ will satisfy $T\pi_{U_M}(\alpha^* X) = 0$. This means that the distribution $Ker(T(\varphi \circ \alpha))$ locally splits as $Ker(T(\varphi \circ \alpha)) = \mathcal{B} \oplus \mathcal{E}$ with $\mathcal{B} = Ker(T\phi)^l$ and $\mathcal{E} = \{X \in Ker(T\varphi) \ : \ T\pi_{U_M}(X) = 0\}$. By the same arguments as in item 1, this decomposition shows that the standard fiber of U_N is diffeomorphic to $\mathcal{F}_M$ factored by the regular equivalence relation induced by $\mathcal{E}$ resulting in $\mathcal{F}_M/R_{\mathcal{E}}$.

   Conversely, since $\mathcal{F}_N$ is diffeomorphic to $\mathcal{F}_M/R_{\mathcal{E}}$, there exists a distribution $\mathcal{E} \subseteq Ker(T\varphi)$ such that $T\pi_{U_M}(\mathcal{E}) = \{0\}$ and this implies the second condition by Proposition 3.19. The proof of projectability now follows the same arguments as in item 1.
3. The first two conditions combined with Proposition 3.17 and (3.43) show that for every $X \in Ker(T\varphi)$, $T\pi_{U_M}(X) = 0$. However since $\mathcal{E} = \{0\}$, by Proposition 3.19 there are no vectors $X \in Ker(T\varphi)$ such that $T\pi_{U_M}(X) = 0$. This implies $dim(Ker(T\varphi)) = 0$ or equivalently that φ is in fact a local isomorphism between U_M and U_N regarded as manifolds without the fiber bundle structure. Nevertheless U_N possesses also a structure of fiber bundle over N induced by the map $\phi \circ \pi_{U_M} : U_M \longrightarrow N$, see [1] for details. This means that the standard fiber of U_N is diffeomorphic to $(\phi \circ \pi_{U_M})^{-1}(y) = \pi_{U_M}^{-1} \circ \phi^{-1}(y)$ which locally assumes the form $\mathcal{F}_M \times \mathcal{K}$.

   The converse is proved by realizing that U_M and U_N are locally diffeomorphic as manifolds via φ. The conditions in item 3 follow directly from this observation.
4. The first two conditions and Proposition 3.17 imply that $T\pi_{U_M}(Ker(T\varphi)) = \{0\}$. Therefore the reduced states by ϕ on M, modeled by $\mathcal{K}$ will lift to the fibers. Since $\mathcal{E} \neq \{0\}$ $\mathcal{F}_N$ will be diffeomorphic to $\mathcal{F}_M/R_{\mathcal{E}} \times \mathcal{F}$.

   The fact that M can be seen as a submanifold of U_N and Proposition 3.17 imply the first two conditions. Since $\mathcal{F}_M$ was reduced by $R_{\mathcal{E}}$ we must have $\mathcal{E} \subseteq Ker(T\varphi)$ and $T\pi_{U_M}(\mathcal{E}) = \{0\}$ which by Proposition 3.19 implies $\mathcal{E} \neq \{0\}$.

□

It is useful to specialize the above results for the case of control affine systems due to their importance in real applications:

COROLLARY 3.21 (Structure of Control Affine Quotients). *Consider a control system* $\Sigma_M = (U_M, F_M)$ *over a manifold* M, $(f, \Sigma_N) = ((\phi, \varphi), (U_N, F_N))$ *a quotient of* Σ_M *and any vector field* $\overline{F_M}$ *in* F_M^e. *Let* $\mathcal{E}$ *be the involutive distribution defined by* $\mathcal{E} = \{X \in 0^e \ : \ [\overline{F_M}, X] \in Ker(T\phi^e)\}$, *which we assume to be regular and denote by* $R_{\mathcal{E}}$ *the regular equivalence relation induced by* $\mathcal{E}$. *Under these assumptions:*

1. **Reduction from states to states and no reduction on inputs**
   Fiber bundle U_N *has base space diffeomorphic to* N, *and standard fiber* $\mathcal{F}_N$ *diffeomorphic to* $\mathcal{F}_M$ *iff:*
   - $\mathcal{S}_M$ *is projectable with respect to* ϕ;
   - $\mathcal{E} = \{0\}$.
2. **Reduction from states to states and from inputs to inputs**
   Fiber bundle U_N *has base space diffeomorphic to* N, *and standard fiber* $\mathcal{F}_N$ *diffeomorphic to* $\mathcal{F}_M / R_{\mathcal{E}}$ *iff:*
   - $\mathcal{S}_M$ *is projectable with respect to* ϕ;
   - $\mathcal{E} \neq \{0\}$.
3. **Reduction from states to inputs and no reduction on inputs**
   Fiber bundle U_N *has base space diffeomorphic to* N, *and standard fiber* $\mathcal{F}_N$ *diffeomorphic to* $\mathcal{F}_M \times \mathcal{K}$ *iff:*
   - $[F_M, Ker(T\phi)] \cap \big(Ker(T\phi) + \Delta\big) = \{0\}$;
   - $[F_M, Ker(T\phi)] \neq \{0\}$;
   - $\mathcal{E} = \{0\}$.
4. **Reduction from states to inputs and from inputs to inputs**
   Fiber bundle U_N *has base space diffeomorphic to* N, *and standard fiber* $\mathcal{F}_N$ *diffeomorphic to* $(\mathcal{F}_M / R_{\mathcal{E}}) \times \mathcal{K}$ *iff:*
   - $[F_M, Ker(T\phi)] \cap \big(Ker(T\phi) + \Delta\big) = \{0\}$;
   - $[F_M, Ker(T\phi)] \neq \{0\}$;
   - $\mathcal{E} \neq \{0\}$.

where $\mathcal{K}$ *is any leave of the foliation on* M *induced by the distribution* $Ker(T\phi)$.

We see that the notion of projectability is fundamentally related to the structure of the abstracted control bundles. If the control section $\mathcal{S}_M$ is projectable then the control inputs of the abstracted system are the same or a quotient of the original control inputs. Projectability can therefore be seen as a structural property of a control system in the sense that it admits special decompositions [33, 58] whenever it is projectable. However, for general systems not admitting this special structure, that is, for systems that are not projectable, the process of abstraction is still possible and it consists of lifting the neglected state information to the fibers. The states of the original system that are abstracted away by ϕ are regarded as control inputs in the abstracted system. This shows that from a hierarchical synthesis point of view, control systems that are not projectable are much more appealing since one can design control laws for the abstracted system, that when pulled-down to the original one are regarded as specifications for the dynamics on the neglected states.

Between cases 1 and 2 of projectability and cases 3 and 4 of non projectability there are more intricate descriptions for the structure of the control bundle related with the decompositionality of the Lie subalgebra defined by $Ker(T\varphi)$. A detailed account of this situation will be given elsewhere.

It is also important to mention that all the abstracting methodology is strongly rooted on the fiber bundle model of control systems. If one assumes a Cartesian product between the state space and the input space, then it is not possible to lift states to inputs since product respecting maps are of the form $\varphi(x,u) = (\varphi_1(x), \varphi_2(u))$. We thus see that a hierarchical view of control design simply means interchanging the role of state and input through the different layers in a hierarchy. This presents a compelling reason to place the distinction between states and inputs as a modeling question and not as a characteristic of physical systems.

6. Examples

In this section we will provide simple examples to illustrate the characterization of the abstracted control bundles.

Example 3.22. We start with a very simple but very characteristic example. Consider a simple mechanical system on the real line described as a double integrator. The control bundle is given by $U_M = \mathbb{R}^2 \times \mathbb{R}$ and the base space $M = \mathbb{R}^2$. Choosing as coordinates for M position x_1 and velocity v_1 we have the following description for F_M:

$$F_M = f + gu = \begin{bmatrix} v_1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \tag{3.61}$$

We now introduce the abstracting map $\phi : \mathbb{R}^2 \longrightarrow \mathbb{R}$ defined by $\phi(x_1, v_1) = x_1$. Its tangent map is given by $T\phi = [1 \ 0]$ and $Ker(T\phi) = span\{\begin{bmatrix} 0 \\ 1 \end{bmatrix}\}$. Computing $[F_M, Ker(T\phi)]$ one obtains:

$$[F_M, Ker(T\phi)] = [\begin{bmatrix} v \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}] + [\begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}] = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{3.62}$$

and we see that $[F_M, Ker(T\phi)] \cap \big(Ker(T\phi) + span\{g\}\big) = \{0\}$ and $[F_M, Ker(T\phi)] \neq \{0\}$ which tell us that all the neglected states will lift into the fibers of the abstracting system. This means that the integral manifold of the distribution $Ker(T\phi)$ which can be coordinatized by the variable v will become an input at the abstracted model. Let us see now what will happen to the input u. Computing

$Ker(T\phi) \cap span\{g\}$ which equals $Ker(T\phi)$ we realize that the control fiber $\mathcal{F}_M = \mathbb{R}$ will be factored by $\mathcal{E}$. Theorem 3.20 tells us that the fibers of the control bundle U_N of the abstracted system will be diffeomorphic to $(\mathbb{R}/R_{\mathcal{E}}) \circ \phi^{-1}(y) \cong \mathbb{R}$.

We now compute the abstraction of control system (3.61) by the methods reviewed in Subsection 2.2. The affine distribution $\mathcal{S}_M$ is defined by:

$$X_M = \begin{bmatrix} v \\ 0 \end{bmatrix} \qquad \Delta_M = span\{\begin{bmatrix} 0 \\ 1 \end{bmatrix}\} \tag{3.63}$$

We now compute L_M as:

$$L_M = \Delta_M + [Ker(T\phi), X_M] = span\{\begin{bmatrix} 0 \\ 1 \end{bmatrix}\} + span\{\begin{bmatrix} 1 \\ 0 \end{bmatrix}\} \tag{3.64}$$

and the abstracting affine bundle $\mathcal{S}_N$ is given by:

$$\begin{aligned} \mathcal{S}_N(y) &= T_x\phi(\mathcal{S}_M(x)) \\ &= T_x\phi(X_M(x) + \Delta_M(x)) \\ &= [1 \;\; 0](\begin{bmatrix} v \\ 0 \end{bmatrix} + T_x M) \\ &= v_1 \end{aligned} \tag{3.65}$$

The last equality holds since $\mathcal{S}_N(y)$ is given by $T_x\phi(\mathcal{S}_M(x))$ for any $x \in \phi^{-1}(y)$. From the affine bundle $\mathcal{S}_N$ we easily obtain the abstraction of (3.61) as:

$$\dot{y} = \dot{x}_1 = v_1 \tag{3.66}$$

where v is now a control input. The fiber respecting map φ induced by ϕ will then be defined as $\varphi((x, v), u) = (x, v)$ which simply abstracts away the input u and lifts v from the base space to the fibers, "promoting" it to a new control input. This example is characteristic in the sense that it is probably the simplest example of hierarchical control. On the abstracted system a control law is a specification of velocity as a function of position and this will correspond on the original model as a specification to be achieved by properly designing an acceleration control law.

Example 3.23. Next we consider a simple example of a full nonlinear control system where no state information is lifted into the fibers. Consider the nonlinear control system described by:

$$\begin{aligned} \dot{x}_1 &= x_2 u_1 u_2 \\ \dot{x}_2 &= x_1^2 u_2^3 \end{aligned}$$

where u_1 and u_2 are the control inputs. The state space is given by $M = \mathbb{R}^2$ and the control bundle by the trivial bundle $U_M = \mathbb{R}^2 \times \mathbb{R}^2$. We now consider the abstraction of this control system defined by the map $\phi : \mathbb{R}^2 \longrightarrow \mathbb{R}$, $\phi(x_1, x_2) = x_2$. We take advantage of the fact that the bundle is trivial by choosing

$\overline{F}_M = F^l_M$ and decomposing $Ker(T\phi^e)$ as $Ker(T\phi^e) = Ker(T\phi)^l + 0^e$. Projectability is now determined by the inclusion:

$$[F^l_M, Ker(T\phi)^l] + [F^l_M, 0^e] \subseteq Ker(T\phi^e) + [F^l_M, 0^e] \tag{3.67}$$

Computing:

$$[F^l_M, Ker(T\phi)^l] = span\{[\begin{bmatrix} x_2u_1u_2 \\ x_1^2u_2^3 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}]\} = span\{-\begin{bmatrix} 0 \\ 2x_1u_2^3 \\ 0 \\ 0 \end{bmatrix}\} = span\{X\} \tag{3.68}$$

$$[F^l_M, 0^e] = span\{[F^l_M, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}], [F^l_M, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}]\} = span\{-\begin{bmatrix} x_2u_2 \\ 0 \\ 0 \\ 0 \end{bmatrix}, -\begin{bmatrix} x_2u_1 \\ 3x_1^2u_2^2 \\ 0 \\ 0 \end{bmatrix}\} \tag{3.69}$$

and defining:

$$Y = -\begin{bmatrix} x_2u_2 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad Z = -\begin{bmatrix} x_2u_1 \\ 3x_1^2u_2^2 \\ 0 \\ 0 \end{bmatrix} \tag{3.70}$$

we see that $\frac{3}{2}x_1u_2X = -u_1Y + u_2Z$ so that $[F^l_M, Ker(T\phi^e)] \subseteq Ker(T\phi^e) + [F^l_M, 0^e]$ and by Theorem 3.20 no states will be lifted into the fibers. With respect to inputs we have $[F^l_M, 0^e] \cap Ker(T\phi)^l \neq \{0\}$ which tell us that the fibers will be factored by the regular equivalence relation $R_{\mathcal{E}}$ induced by $\mathcal{E} = span\{[0\,0\,1\,0]^T\}$. Theorem 3.20 then asserts that the new control bundle is diffeomorphic to $\mathbb{R} \times \mathbb{R}$. Although the methods proposed in [63, 64] to compute abstractions only deal with control affine systems we can compute the abstraction "manually" for this simple example. Let $\mathcal{S}_M$ be the control section associated with F_M, then by computing $T_x\phi(X)$ for every $X \in \mathcal{S}_M(x)$ we obtain:

$$T_x\phi(\begin{bmatrix} x_2u_1u_2 \\ x_1^2u_2^2 \end{bmatrix}) = x_1^2u_2^3 \tag{3.71}$$

so that the control section $\mathcal{S}_N$ is defined by $\mathcal{S}_N = \{x_1^2u_2^3 \in T\mathbb{R} \ : \ x_1 \in \mathbb{R} \wedge u_2 \in \mathbb{R}\}$ and can equivalently be described by $\mathcal{S}_N = \{u \in T\mathbb{R} \ : \ u \in \mathbb{R}\}$. A control parameterization for $\mathcal{S}_N$ is then given by $U_N = \mathbb{R} \times \mathbb{R}$ and control system F_N defined by:

$$\dot{x} = u \tag{3.72}$$

which agrees with the results given by Theorem 3.20

CHAPTER 4

Abstractions of Hybrid Control Systems

1. Introduction

In this chapter we develop a formal framework to introduce abstractions for hybrid control systems and study their properties. Based on the insights obtained in the last chapter we use again simple ideas from category theory and introduce the category of *abstract control systems*. The objects will be abstract control systems capturing discrete, continuous and hybrid control systems. To be able to work at such a general level we start from the hybrid automaton and extract its mathematical structure by defining an hybrid control system as a partial monoid action. This characterization of hybrid control systems emphasizes its similarity with labeled transition systems and smooth control systems thereby suggesting the general notion of abstract control systems. As morphisms, in the category of abstract control systems, we will consider relations that preserve the partial monoid action structure. There are two main reasons to adopt relations instead of functions. The first is that it allows to define the concept of bisimulation through the use of the inverse relations. While for relations there always exist inverse relations, the same is no longer true for functions. Although this problem could be solved by adopting other formulations of bisimulation, of which we mention [36] by its intuitive elegance, there is still a much more compelling reason to use relations. When aggregating continuous to discrete information we will face the problem of abstracting continuous evolutions to discrete jumps. This, as we will see, will require to map points in the state space of the original hybrid system to *sets* of points in the state space of its abstraction and relations are flexible enough to accommodate these requirements.

As in the continuous case we propose a notion of abstraction based on simulations which are captured by the morphisms of the category, that is, system A is a simulation of system B if there is a morphism from B to A. However we will also provide a stronger notion of abstraction, namely bisimulations. We define bisimulations as symmetric simulations, that is, system A is a bisimulation of system B if there is a morphism (which is a relation is this case) from B to A and the inverse relation is also a morphism from A to B. Bisimulation defines a very special equivalence relation of the class of abstract control systems since cardinality (or dimension, when we can talk about it) is not constant on the equivalence classes. This fact is the essence of complexity reduction since analysis or synthesis tasks can be performed much more efficiently on lower cardinality equivalent systems.

We also introduce a composition operator in the category of abstract control systems modeling the interconnection and synchronization of subsystems. This operator is based on the categorical view of concurrency described in [89] and is another powerful tool for complexity reduction. In fact, we show that simulations are compositional in the sense that composing simulations of subsystems results in a simulation of the overall system. We also show that bisimulations are compatible with composition under certain conditions on the synchronization of the subsystems.

All of these results are them specialized to hybrid control systems where simpler versions of some results are given. We also provide an algorithm to compute abstractions of hybrid control systems and show that under certain assumptions the algorithm computes bisimulations.

2. Hybrid Automata: An operational perspective

Hybrid systems originally appeared as a model for systems comprising discrete and continuous evolution. Examples range from man engineered systems such as computer controlled physical processes to several examples from nature like the motion of a bouncing ball. To capture all of these similarly different systems in a common model, ideas from computer science and control theory were merged into what is usually called an *hybrid automaton* [26]:

DEFINITION 4.1 (Hybrid Automata). An hybrid automaton is a tuple $H = (Q, M, Init, Inv, Guard, Reset, F)$ consisting of:

- Q is a finite set of discrete states.
- M is a smooth manifold.
- $Init \subseteq Q \times M$ is a set of initial states.
- $Inv : Q \to \mathcal{P}(M)$ is a map assigning to each $q \in Q$ a subset of M called the invariant.
- $Guard : Q \times Q \to \mathcal{P}(M)$ is a map assigning to a pair of discrete states a subset of M called the guard.
- $Reset : Q \times Q \times M \to M$ is a map such that given a pair of discrete states, maps points in M to a set of points in M.
- $F : Q \times M \to TM$ is a map assigning a vector field $F(\cdot, x) \in TM$ for each $q \in Q$.

If F is not a vector field, but a control system, then we have an hybrid control system as opposed to an hybrid dynamical system. The state space associated with an hybrid system is given by $Q \times M$ and a point is represented by the pair (q, x). The semantics associated with a trajectory of an hybrid automaton is the following: a trajectory originates in a state $(q_0, x_0) \in Init$ and consists of concatenations of *discrete jumps* and *continuous flows*. A continuous flow keeps the discrete part q of the state (q, x) constant while the continuous part x evolves according to $\frac{d}{dt}x(t) = F(q, x(t))$ while $x(t)$ belongs to $Inv(q)$. When the continuous part of the state attempts to leave the invariant either $x \in Guard(q, q')$ for some $q' \in Q$ and a discrete jump from q to q' is *forced* or the trajectory is not defined beyond that point and we say that the hybrid automaton has blocked or is blocking. If a discrete jump is forced, the state jumps instantaneously from (q, x) to (q', x') where $x' \in Reset(q, q', x)$. A discrete jump may also happen in a controlled way as opposed to being forced. Whenever the continuous part of the state belongs to both the invariant and the guard associated to some discrete transition, the jump can be taken, but is not forced to. A choice is then made between taking the discrete jump or continuing to flow continuously. After a discrete jump, if the continuous part of the state belongs to the invariant of the new discrete state another continuous evolution takes place. The trajectory continues then evolving by continuous flows and discrete evolutions or blocks at some state.

An hybrid automaton is usually displayed graphically as a directed graph where the vertices are represented by circles containing the vector field F and the invariant. The discrete transitions between states are represented by arrows labeled by the guard and the reset associated with that transition. Consider, for example, an hybrid automaton modeling a thermostat as displayed in Figure 1. The thermostat has two modes of operation: OFF and ON. When the OFF mode is active, the temperature decreases according to the law $\dot{x} = -kx$, where k is a constant depending on the room characteristics. When in the ON mode, the temperature evolution is described by $\dot{x} = k(h - x)$, where h is a constant modeling the heater performance. The goal of the thermostat is to keep the temperature between T_{MAX} and T_{MIN} which dictates the switching logic between the ON and OFF modes. This hybrid automaton is therefore defined by:

$$
\begin{aligned}
&Q = \{ON, OFF\} \\
&M = \mathbb{R} \\
&Init = Q \times M \\
&Inv(ON) =]-\infty, T_{MAX}] \\
&Inv(OFF) = [T_{MIN}, +\infty[\\
&Guard(ON, OFF) = \{T_{MAX}\} \\
&Guard(OFF, ON) = \{T_{MIN}\} \\
&Reset(ON, OFF, x) = \{x\} \\
&Reset(OFF, ON, x) = \{x\} \\
&F(ON, x) = k(h - x) \\
&F(OFF, x) = -kx
\end{aligned}
$$

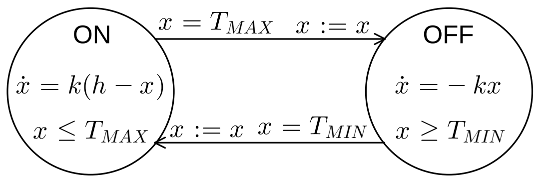

FIGURE 1. Hybrid automaton model of a thermostat.

The hybrid automaton model provides an operational description of hybrid systems in the sense that it provides a way of computing or implementing the trajectories of an hybrid system. However, it does not emphasize the structure of hybrid systems as a mathematical object. It is towards this objective that we proceed in the next section, where we will provide an alternative description of hybrid systems emphasizing their mathematical structure.

3. Abstract Control Systems

In order to capture continuous, discrete, and hybrid systems under an unified model, we need an abstract definition of control systems. The essence of a control system is reflected into two different aspects: a notion of evolution, and the ability to control the evolution. These two fundamental aspects are captured in the following definition:

DEFINITION 4.2 (Abstract Control System). Let S be a set, $\mathcal{M}$ a monoid and A a fibering relation on $S \times \mathcal{M}$ with base space S such that A_s is a prefix closed subset of $\mathcal{M}$ containing the identity for every $s \in S$. An abstract control system over S is a map $\Phi : A \longrightarrow S$ respecting the monoid structure, that is $\Phi_s : A_s \longrightarrow S$ verifies:

1. **Identity:** $\Phi_s(\varepsilon) = s$
2. **Semi-group:** $\Phi_{\Phi_s(a_s)}(a_{s'}) = \Phi_s(a_s a_{s'})$

Intuitively, we can think of the set S as the state space, and the fiber bundle A, also called in this work a fibering monoid, as the set of possible actions, that depend on the base point. The map Φ assigns to each point $s \in S$ a function from A_s to S representing all the input choices that can be made at the point s. Given an input choice $a_s \in A_s$, $\Phi_S(a_s)$ returns the state reached under the action of the control input a_s.

We adopt the following intuitive graphical notation to denote evolution from s controlled by a and described by Φ, that is, $\Phi_s(a) = s'$ is represented by $s \xrightarrow{a} s'$.

We could model abstract control systems in a more elegant way by defining them to be a generalized monoid, that is a small category. We would then have as objects the elements of S and every $a_s \in A_s$ would be considered a morphism from s to $\Phi(s, a_s)$. However, we will use the above definition since it is more easily associated and compared with standard notions such as monoid and group actions. To get a better understanding of the above definition we will see how it applies to three classes of systems.

3.1. Discrete Control Systems as Abstract Control Systems. The usual model for discrete control systems are automata however it will be enough to work with transition systems. Let (Q, Σ, δ) be a discrete labeled transition system, where Q is a finite set of states, Σ is a finite set of input symbols, and $\delta : Q \times \Sigma \longrightarrow Q$ is the next-state function. Usually, transitions are modeled by a transition relation $\longrightarrow \in Q \times \Sigma \times Q$, but we will restrict to deterministic transition systems. Note also that δ is in general a partial function. Let us denote by Σ^* the set of all finite strings obtained by concatenating elements in Σ. In particular the empty string ε also belongs to Σ^*. With concatenation as a monoid operation, Σ^* can be taken as the monoid $\mathcal{M}$. The state space is naturally $S = Q$. The transition function δ defines a *unique* partial map from $Q \times \Sigma^*$ to Q which is just an abstract control system $\Phi : (S \times \mathcal{M})|_R = A \longrightarrow S$, where R is the fibering monoid given by $R = \{(s, m) \in S \times \mathcal{M} \ : \ \Phi(s, m) \text{ is defined}\}$.

To clarify the resemblances to the continuous case that we describe next, we elaborate a little on the structure of the monoid Σ^*. This monoid has been defined as the set of all finite sequences of elements in Σ. Alternatively we can regard Σ^* as the disjoint union of the collection of maps $\Sigma^{\varnothing} \cup \Sigma^{\{1,2,\ldots,t\}}$ with $t \in \{1, 2, \ldots, n\}$. Given any string $s = m_1 m_2 m_3 m_4 \ldots m_n \in \Sigma^*$ we can identify it with the map $u : \{1, 2 \ldots, n\} \longrightarrow \Sigma$ defined by $u(1) = m_1$, $u(2) = m_2$, $\ldots$, $u(n) = m_n$. The empty string ε is identified with the map $u : \varnothing \longrightarrow \Sigma$ and concatenation of strings can be seen as concatenation of maps defined as follows:

$$
\begin{aligned}
\cdot : U^{1,2,\ldots,t_1} \times U^{1,2,\ldots,t_2} &\longrightarrow U^{1,2,\ldots,t_1+t_2} \\
(u(t), v(t)) &\mapsto (u \cdot v)(t) = \begin{cases} u(t) & \text{if} \quad 1 \leq t \leq t_1 \\ v(t - t_1) & \text{if} \quad t_1 + 1 \leq t \leq t_1 + t_2 \end{cases}
\end{aligned}
\tag{4.1}
$$

The above operation only allows to concatenate maps such that its domain ends in a finite number, since it is not possible to append the second map at the end of the first one, if the end is a non-finite[1] number. This forces to work with the class of maps defined on intervals with finite end point, that is:

$$
\Sigma^* = \coprod_{t \in \mathbb{N}_0} \Sigma^t \tag{4.2}
$$

which is closed for concatenation of maps, posses identity ε and therefore it is a monoid since concatenation is an associative operation. Note that in this case all the maps we are considering are defined on finite

[1]In fact this is possible but one would have to resort to ω-monoids, see for example [65]. This construction will be sketched when dealing with the Zeno phenomena.

subsets of the naturals and the condition that the end point of the domain is finite is equivalent to saying that the number of symbols in the string is finite. This will not be the case for continuous systems as we will see shortly.

3.2. Continuous Control Systems as Abstract Control Systems. For simplicity of presentation, we consider only time-invariant control systems, although the construction to be presented is generalizable to time varying systems. Let U be the space of admissible inputs. Define the set U^t as:

$$U^t = \{u : [0,t[\longrightarrow U \quad | \quad [0,t[\subseteq \mathbb{R}_0^+\} \tag{4.3}$$

An element of U^t is denoted by u^t, and represents a map from $[0,t[$ to U. Consider now the set U^* which is the disjoint union of all U^t for $t \in \mathbb{R}_0^+$:

$$U^* = \coprod_{t\in\mathbb{R}_0^+} U^t \tag{4.4}$$

The set U^* can be regarded as a monoid under the operation of concatenation, that is, if $u^{t_1} \in U^{t_1} \subset U^*$ and $u^{t_2} \in U^{t_2} \subset U^*$ then $u^{t_1}u^{t_2} = u^{t_1+t_2} \in U^{t_1+t_2} \subset U^*$ with concatenation given by:

$$u^{t_1}u^{t_2}(t) = \begin{cases} u^{t_1}(t) & \text{if} \quad 0 \leq t < t_1 \\ u^{t_2}(t-t_1) & \text{if} \quad t_1 \leq t < t_1+t_2 \end{cases} \tag{4.5}$$

The identity element is given by the empty input, that is $\varepsilon = u^0$. This construction parallels the construction that obtains Σ^* from Σ, however in this case the finiteness condition on the end point of the domain of the map u^t no longer implies that each string has only a finite number of elements. We can have an infinite number of concatenations as long as the sum of the duration times converges.

We now show how this monoid is used to describe any smooth control system as an abstract control system. Let $\dot{x} = f(x,u)$ be a smooth control system, where $x \in M$, a smooth manifold and $u \in U$, the set of admissible inputs. Choosing an admissible input trajectory u^t, $f(x,u^t)$ is a well defined vector field and as such it induces a flow which we denote by $\gamma_x : [0,t[\longrightarrow M$, such that $\gamma_x(0) = x$. We can then cast any smooth control system into our framework by defining:

$$\begin{aligned} \Phi : M \times U^* &\longrightarrow M \\ (x,u^t) &\mapsto \gamma_x(t) \end{aligned} \tag{4.6}$$

It is not difficult to see that Φ is in fact a well defined abstract control system since $\Phi(x,\varepsilon) = \gamma_x(0) = x$ and $\Phi(x,u^{t_1}u^{t_2}) = \gamma_x(t_1+t_2) = \gamma_{\gamma_x(t_1)}(t_2) = \Phi(\Phi(x,u^{t_1}),u^{t_2})$. In general the set of admissible control inputs may change with the point x so that the domain of Φ will be in fact a fiber bundle over M. It is also interesting to note that when U is a singleton for every $x \in M$ (there are no choices to be made) the set U^t can be identified with the number t so that U^* is given by $U^* = \coprod_{t\in\mathbb{R}_0^+} t = \mathbb{R}_0^+$ and our abstract control system Φ degenerates into an action of $\mathbb{R}_0^+$ on M, that is, the solution of a differential equation (a degenerate control system).

3.3. Hybrid Control Systems as Abstract Control Systems. Hybrid control systems also fit in the abstract control system framework. The state space of an hybrid control system is usually described as $Q \times M$, where Q is a finite set of states and M a smooth manifold. However it will be convenient to relax this concept and look at the state space as a fiber bundle. Instead of considering the same manifold M for every $q \in Q$ we consider a set of smooth manifolds X_q parameterized by the discrete states and denoted by $X = \{X_q\}_{q \in Q}$. The discrete set Q is thought as the base space, and for each base point $q \in Q$ we attach a fiber X_q. A point in X is represented by the pair (q, x).

As action monoid we will use the set:

$$\mathcal{M} = \coprod_{t \in \mathbb{N}_0} (U^* \cup \Sigma^*)^t \tag{4.7}$$

assuming that $U^* \cap \Sigma^* = \{\varepsilon\}$ and regarding U^* and Σ^* simply as sets. Let us elaborate on the product operation on $\mathcal{M}$. This operation is defined as the usual concatenation and therefore it requires finite length strings. To accommodate this requirement and still be able to have an infinite number of concatenations of elements in U^* we proceed as follows. Suppose that we want to show that $\sigma_1 u^{t_1} u^{t_2} \ldots u^{t_n} \ldots \sigma_2$ belongs to $\mathcal{M}$, where t_n is a convergent sequence. Instead of regarding each element in the string as an element in $\mathcal{M}$, which would not allow us to define the last concatenation since it would happen after ∞, we regard σ_1 and σ_2 as elements of $\mathcal{M}$ and $u^{t_1} u^{t_2} \ldots u^{t_n} \ldots = u^{t'}$ as an element of U^* and consequently as an element of $\mathcal{M}$, where $t' = \lim_{n \longrightarrow \infty} t_n$. This string is then regarded as the map $u : \{1, 2, 3\} \longrightarrow \mathcal{M}$ defined by $u(1) = \sigma_1$, $u(2) = u^{t'}$ and $u(3) = \sigma_3$. The product in $\mathcal{M}$ is then the usual concatenation on reduced strings, that is, strings where all consequent sequences of elements of U^* or Σ^* have been replaced by their product in U^* or Σ^*, respectively. The monoid $\mathcal{M}$ obtained by this construction is called the free product of U^* and Σ^* and is is fact the coproduct in the category of monoids. Furthermore we have the following characterization of $\mathcal{M}$:

Proposition 4.3 ([30]). *The monoid $\mathcal{M}$ is freely generated by the symbols $U^* \cup \Sigma^*$.*

Since the continuous control systems will, in general, be different at each fiber X_q, U will be a finite family of admissible continuous control input spaces parameterized by the discrete states, that is $U = \{U_q\}_{q \in Q}$. Hybrid control systems are now cast into the abstract control systems framework as:

Definition 4.4 (Hybrid Control System). An hybrid control system $H = (X, A_X, \Phi_X)$ consists of:

- The state space $X = \{X_q\}_{q \in Q}$.
- A fibering relation A_X on $X \times \mathcal{M}$ defined by:
  $A_X = \{((q, x), m) \in X \times \mathcal{M} \ : \ \Phi_X((q, x), m) \text{ is defined}\}$.

- A map $\Phi_X : A_X \longrightarrow X$ respecting the monoid structure such that for all $q \in Q$, there is a set $Inv(q) \subseteq X_q$ and for all $x \in Inv(q)$, $A_{(q,x)} \cap U^* \neq \{\varepsilon\}$ and $\Phi((q,x), u^{t'}) \in Inv(q)$ for every prefix $u^{t'}$ of every $u^t \in A_{(q,x)}$.

The semantics associated with the evolution from (q,x) governed by Φ and controlled by $a \in A_{(q,x)}$ is the standard transition semantics of hybrid automata [26]. Suppose that $a = u^{t_1}\sigma_1\sigma_2 u^{t_2}$, then $\Phi((q,x),a) = (q',x')$ means that the system starting at (q,x) evolves during t_1 units of time under continuous input u^{t_1}, jumps under input σ_1 and them jumps again under σ_2. After the two consecutive jumps, the system evolves under the continuous control input u^{t_2} reaching (q',x'), t_2 units of time after the last jump. From the hybrid system construction we can clearly extract the purely discrete case (X_q is a singleton and $U_q = \varnothing$ for each $q \in Q$) as well as the purely continuous case (Q is a singleton and $\Sigma = \varnothing$).

3.4. Control System Abstractions. Having characterized the structure of hybrid systems we now consider simulation relations, and in particular abstractions, between the general systems considered in Definition 4.2. These notions will be specified by requiring that the structure is preserved between the original system and its abstraction. Although for discrete and smooth systems a notion of simulation based on a map between fibering monoids is able to model the relevant concepts and constructions, that will not be the case for hybrid control systems. A map between fibering monoids turns out to be too restrictive and one is forced to look into more general notions of simulation. The link between the fibering monoids will be provided by a *relation*[2] which is general enough for our purposes. A notion of simulation will involve a relation between fibering monoids that respects the control structure given by the map Φ. This is formalized as follows:

Definition 4.5 (Simulations of Abstract Control Systems). Let Φ_X and Φ_Y be two abstract control systems over X and Y with fibering monoids A_X and A_Y, respectively. Let $R \subseteq A_X \times A_Y$ be a fibering monoid respecting relation. Then Φ_Y is a simulation of Φ_X with respect to R or a R-simulation if and only if:

$$\forall_{x\in X}\ (x,y) \in R_B \ \Rightarrow\ \forall_{(x,a_x)\in dom(R)}\ \exists_{(x,a_x,y,a_y)\in R} \qquad (\Phi_X(x,a_x), \Phi_Y(y,a_y)) \in R_B \tag{4.8}$$

This definition slightly generalizes the usual notions of morphisms between transition systems as in [89], since we allow the control inputs to depend on the state space and since we use relations instead of functions. It is not difficult to see that abstract control systems and relations satisfying condition (4.8) form a category, that we call the abstract control systems category. It is also clear that the category of discrete control systems and also the category of smooth control systems are subcategories of this larger category.

[2]In fact it was by means of a relation that the notion of bisimulation was introduced in [52]

It may seem that checking if R is fibering monoid preserving might be a difficult task in concrete examples. We will see, however, that for hybrid systems the relations we will consider are fibering monoid respecting by construction.

If we regard an abstract control system as a small category, then a simulation is a functor between categories that may be multi-valued on both objects and morphisms.

We now propose the following notion of abstraction based on simulations:

DEFINITION 4.6 (Abstractions of Abstract Control Systems). Let Φ_X and Φ_Y be abstract control systems over X and Y with fibering monoids A_X and A_Y, respectively. If $R \subseteq A_X \times A_Y$ is a fiber respecting relation we say that Φ_Y is an R-abstraction of Φ_X iff Φ_Y is an R-simulation of Φ_X and R is a surjective relation with domain A_X.

The notion of bisimulation also follows naturally:

DEFINITION 4.7 (Bisimulations of Abstract Control Systems). Let Φ_X and Φ_Y be abstract control systems over X and Y with fibering monoids A_X and A_Y respectively. If $R \subseteq A_X \times A_Y$ is a fiber respecting relation we say that Φ_X is R-bisimilar to Φ_Y iff Φ_Y is a R-simulation of Φ_X and Φ_X is a R^{-1}-simulation of Φ_Y.

The approach taken to define bisimulation is similar in spirit to the one in [52], however instead of preserving labels of the abstract control systems, we relate them through the relation. Several other approaches to bisimulation are reported in the literature and we point the reader to the comparative study in [73] and the references therein. How this notion relates with the others is an important issue that will be discussed elsewhere.

The importance of simulations lies on the fact that simulations capture all trajectories of the simulated abstract control system. We now make this fact precise. Instead of trying to define trajectories of abstract control systems (which would be as difficult as defining trajectories of hybrid control systems, see the different approaches in [35, 53, 67, 85]) we will restrict our attention to the orbits of abstract control systems.

DEFINITION 4.8. Let Φ_X be an abstract control system over X with fibering monoid A_X. The set $\mathcal{O}_x$ is an orbit from the point $x \in X$ iff:

$$\exists_{a_x \in A_x} \text{ such that } \mathcal{O}_x = \{x' \in X \ : \ x' = \Phi_X(x, a'_x) \text{ for every prefix } a'_x \text{ of } a_x\} \tag{4.9}$$

Intuitively, the orbit $\mathcal{O}_x$ through x is the set of all the points visited by Φ_X while being controlled by a_x. We can now relate the orbits of abstract control systems to the orbits of the corresponding simulations:

PROPOSITION 4.9. *Let Φ_X and Φ_Y be abstract control systems over X and Y with fibering monoids A_X and A_Y, respectively. If Φ_Y is a R-simulation of Φ_X with respect to a fibering monoid respecting relation R induced by a map $\varphi : A_X \longrightarrow A_Y$ then:*

$$\phi(\mathcal{O}_x) = \mathcal{O}_{\phi(x)} \quad \forall_{x \in X} \quad \forall_{\mathcal{O}_x} \tag{4.10}$$

where $\phi : X \longrightarrow Y$ is the map induced by R_B.

PROOF. Assume that Φ_Y is a R-simulation of Φ_X and let $x \in X$ be R_B related to $y \in Y$. For any $(x, a_x) \in dom(R)$ there exists a pair $((x, a_x), (y, a_y)) \in R$ such that $\phi \circ \Phi_X(x, a_x) = \Phi_Y(y, a_y) = \Phi_Y(\varphi(x, a_x))$ by definition of simulation and the fact that R is induced by φ. Therefore:

$$\begin{aligned} \phi(\mathcal{O}_x) &= \bigcup_{a'_x \text{ prefix of } a_x} \phi(\Phi_X(x, a'_x)) \\ &= \bigcup_{a'_x \text{ prefix of } a_x} \Phi_Y(\varphi(x, a'_x)) \end{aligned}$$

Since φ maps prefixes of a_x to prefixes of a_y (as it is a fibering monoid respecting map) for $(y, a_y) = \varphi(x, a_x)$ the previous expression can also be written as:

$$\begin{aligned} \bigcup_{a'_x \text{ prefix of } a_x} \Phi_Y(\varphi(x, a'_x)) &= \bigcup_{a'_y \text{ prefix of } a_y} \Phi_Y(y, a'_y) \\ &= \mathcal{O}_y \\ &= \mathcal{O}_{\phi(x)} \end{aligned}$$

and the proof is finished. $\square$

If the fibering monoids are related by a relation that is not induced by a function, then we only have a weaker version of Proposition 4.9 as illustrated in the next example.

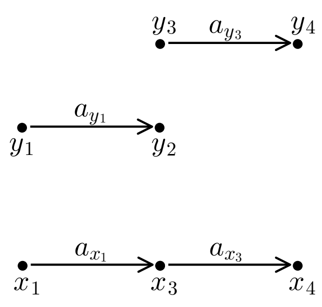

FIGURE 2. An abstract control system and one possible simulation.

EXAMPLE 4.10. Consider the abstract control system H_X displayed in the lower part of Figure 2, where the ε transitions are not displayed. The abstract control system displayed in the top part of the figure is a simulation of H_X with respect to the relation:

$$\begin{aligned} R &= \{((x_1,a_{x_1}),(y_1,a_{y_1})),((x_1,\varepsilon),(y_1,\varepsilon)),((x_3,a_{x_3}),(y_3,a_{y_3})) \\ &\quad ((x_3,\varepsilon),(y_3,\varepsilon)),((x_3,\varepsilon),(y_2,\varepsilon)),((x_4,\varepsilon),(y_4,\varepsilon))\} \end{aligned} \tag{4.11}$$

We then see that the evolution $x_1 \xrightarrow{a_{x_1}} x_3$ is simulated by $y_1 \xrightarrow{a_{y_1}} y_2$ while the evolution $x_3 \xrightarrow{a_{x_3}} x_4$ is simulated by $y_3 \xrightarrow{a_{y_3}} y_4$. However, $y_2 \neq y_3$ as a consequence of the nondeterminism imposed by $R_B(x_3) = \{y_2, y_3\}$. Nevertheless, relations will be play an important role in describing simulations for hybrid control systems.

We have already seen that abstractions preserve orbits but in the next section we will see in detail that abstractions may preserve other properties as well.

3.5. Preservation of Properties. In this section we will study preservation of properties that will become important for the later study of hybrid systems.

3.5.1. *Reachability.*

DEFINITION 4.11 (Reachable Space). Let Φ_X be an abstract control system over X. The reachable space from a point $x \in X$, and denoted by $Reach_x(\Phi_X)$ is given by:

$$Reach_x(\Phi_X) = \bigcup_{a \in A_x} \Phi_X(x,a) \tag{4.12}$$

The reachable space from a set $X' \subset X$ is denoted by $Reach_{X'}(\Phi_X)$ and is defined as:

$$Reach_{X'}(\Phi_X) = \bigcup_{x \in X'} Reach_x(\Phi_X) \tag{4.13}$$

Simulations preserve reachable sets in the sense that given an initial condition $x' \in X$ there exists a choice function $\phi : X \longrightarrow Y$ relating the reachable space of and abstract control system with the reachable space of its simulation:

PROPOSITION 4.12. *Let Φ_X and Φ_Y be two abstract control systems on X and Y, respectively. If Φ_Y is a R-simulation of Φ_X for a relation R with domain A_X, then for every $x' \in X$ there exists a map $\phi : X \longrightarrow Y$ such that $(x, \phi(x)) \in R_B$ and $\phi(Reach_{x'}(\Phi_X)) \subseteq Reach_{\phi(x')}(\Phi_Y)$*

PROOF. Let us define ϕ. For x', $\phi(x)$ is any $y' \in Y$ such that $(x', y') \in R_B$. For any $x \in Reach_{x'}(\Phi_X)$, $\phi(x) = \Phi_Y(\phi(x'), a_{\phi(x')})$, where $((x', a_{x'}), (\phi(x'), a_{\phi(x')})) \in R$ and $x = \Phi_X(x', a_{x'})$. Note that $(\phi(x'), a_{\phi(x')})$ exists since the domain of R is X and by definition of simulation $(x, \Phi_Y(\phi(x'), a_{\phi(x')})) \in R_B$. This allow us to conclude that for any $x \in Reach_{x'}(\Phi_X)$, $\phi(x) = \Phi_Y(\phi(x'), a_{\phi(x')}) \in Reach_{\phi(x')}(\Phi_Y)$ as desired. We have already shown the desired inclusion so that the definition of ϕ for points not belonging to $Reach_{x'}(\Phi_X)$ is arbitrary. $\square$

This result is in fact a natural consequence of the fact that simulations preserve trajectories. Safety properties expressed in several temporal logics can also be shown to be preserved based on the notion of simulation, however, we shall not explore further this aspect.

3.5.2. *Blocking.* Another important property is the absence of dead-locks on the system being modeled by a discrete or hybrid control system. The analogue for continuous systems is the non-existence of finite explosion times. This property is usually called non-blocking and is defined as:

DEFINITION 4.13. Let Φ_X be an abstract control system. Φ_X is said to be non-blocking from $S \subseteq X$ iff for every $x \in Reach_S(\Phi_X)$, $A_x \neq \{\varepsilon\}$.

In general simulations do not preserve non-blocking, however this can be achieved under the proper assumptions:

PROPOSITION 4.14. *Let Φ_X be an abstract control system and Φ_Y a R-abstraction of Φ_X. If:*

- *Φ_X is non-blocking from S*
- *for any $(x, y) \in R_B$ such that:*
  - *$x \in Reach_S(\Phi_X)$*
  - *$y \in Reach_{R_B(S)}(\Phi_Y)$*
  - *$(x', y) \in R_B$ for every $x' \in Reach_x(\Phi_X)$*

  there exists an action $a_x \in \cup_{x \in R_B^{-1}(y)} A_x$ such that $R(x, a_x) \neq \{(y, \varepsilon)\}$

then Φ_Y is non-blocking.

PROOF. We will proceed by contradiction. Assume that Φ_Y is blocking from $R_B(S)$ and that the proposition conditions hold. Since Φ_Y is blocking from $R_B(S)$ there is a $y \in Reach_{R_B(S)}(\Phi_Y)$ such that $A_y = \{\varepsilon\}$. By surjectivity of R there is a $(x, a_x) \in A_X$ that is R-related to (y, ε). Let W be the set of all $(x, a_x) \in A_X$ R-related to (y, ε). This set satisfies $Reach_x(\Phi_X) \subseteq \pi_X(W)$ for every $x \in \pi_X(W)$ since from $dom(R) = A_X$ it follows that for any $a_x \in A_x$, $(x, a_x) \in W$ and this in turn implies that $(\Phi_X(x, a_x), y) \in R_B$ by (4.8). It follows that $\Phi_X(x, a_x) \in \pi_X(W)$ and therefore $Reach_x(\Phi_X) \subseteq \pi_X(W)$. However, we know that there is an action $a_x \in \cup_{x \in \pi_X(W)} A_x$ such that $((x, a_x), (y, a_y)) \in R$ with $a_y \neq \varepsilon$ which contradicts the fact that Φ_Y is blocking at y. □

This condition is also necessary as we now show:

PROPOSITION 4.15. *Let Φ_X be an abstract control system and Φ_Y a R-abstraction of Φ_X. If Φ_X is non-blocking from S and Φ_Y is non-blocking from $R_B(S)$ then for any $(x, y) \in R_B$ such that:*

- *$x \in Reach(\Phi_X)$*
- *$y \in Reach(\Phi_Y)$*

- $(x', y) \in R_B$ *for every* $x' \in Reach_x(\Phi_X)$

there exists an action $a_x \in \cup_{x \in R_B^{-1}(y)} A_x$ *such that* $R(x, a_x) \neq \{(y, \varepsilon)\}$.

PROOF. Admit that Φ_Y is nonblocking from $R_B(S)$. Let W be the set of all elements from X that are R_B-related to some $y \in Reach_{R_B(S)}(\Phi_Y)$. If $Reach_x(\Phi_X) \nsubseteq W$ for any $x \in W$ then the result is vacuously true. If $Reach_x(\Phi_X) \subseteq W$ for some $x \in W$ then since Φ_Y is nonblocking from y there is an action $a_y \in A_y$, $a_y \neq \varepsilon$ such that the pair (y, a_y) is R-related to (x, a_x) with $a_x \in \cup_{x \in W} A_x$ by surjectivity of R. □

This result is clearly unpractical since it involves conditions that are not possible to check in practice. However it is difficult to give checkable conditions at this level of generality. When dealing specifically with hybrid control systems at Section 4 we will be able to take advantage of the structure of hybrid control systems to be able to give results based on more easily verifiable conditions.

3.6. When are two abstract control systems bisimilar? When synthesis (and not analysis) is the important issue one is interested in ensuring that every trajectory of the abstraction has a feasible implementation on the original, more detailed model. This allows to design controllers for the abstraction and then *refine* them on the original system by incorporating the modeling details not present on the abstraction. Feasibility of implementations or refinements asks for the original model to be a simulation of the abstraction, emphasizing the role of bisimulations. They allow analysis as well as synthesis processes to be performed more efficiently since they render both models equivalent, although one of the models has preferably lower complexity than the other. Furthermore when dealing with hybrid control systems we will provide a constructive algorithm to compute simulations of hybrid control systems. Ideally, one would like to produce bisimulations through the algorithm and therefore we need to develop alternative characterizations of bisimilar systems to determine when we are in fact computing bisimulations. To accomplish this we will restrict attention to fibering monoids A_X freely generated by fiber bundles of generators G_X. This means that any element a_x in the fiber A_x over $x \in X$ can be obtained by multiplying elements $g_x^1, g_x^2, \ldots$ on the fiber G_x over $x \in X$. This assumption is justified by the fact that in the hybrid case the monoid $\mathcal{M}$ is free on the set $\Sigma^* \cup U^*$. Furthermore Σ^* is free on the set Σ and U^* is also free on a set of infinitesimal generators. We restrict our attention to abstract control systems factored by equivalence relations on the state space, since they capture the essence of the abstraction methodology we will later propose for hybrid control systems. Let $\phi : X \longrightarrow Y$ be a surjective map and define the equivalence relation $\sim \subseteq X \times X$ by $x_1 \sim x_2$ iff $\phi(x_1) = \phi(x_2)$. Based on this relation we can quotient Φ_X obtaining $\Phi_Y = \Phi_X / \sim$. To define the quotient abstract control system Φ_Y we introduce the operator $\mathcal{R}_x \Phi_X$ returning the subset of X reachable from $x \in X$ by Φ_X when controlled by elements in G_X, that is:

$$\mathcal{R}_x \Phi_X = \bigcup_{g_x \in G_x} \Phi_X(x, g_x) \tag{4.14}$$

This operator allows to introduce the fiber bundle G_Y of the generators of A_Y defined as a fiber bundle over $Y = \phi(X)$ with fiber over any $y \in Y$ isomorphic (as a set) to:

$$\bigcup_{x \in \phi^{-1}(y)} \phi \circ \mathcal{R}_x \Phi_X \tag{4.15}$$

The quotient control system Φ_Y can now be defined by:

$$\Phi_Y(y, a_y) = y' \quad \text{iff} \quad \exists_{a_x \in A_x} \; \Phi_X(x, a_x) = x' \; \wedge \; \phi(x) = y \; \wedge \; \phi(x') = y' \; \wedge \; \varphi(x, a_x) = (y, a_y) \tag{4.16}$$

for a surjective, fibering monoid respecting map $\varphi : A_X \longrightarrow A_Y$ implicitly defined by the following commutative diagram:

$$\begin{tikzcd} A_Y \arrow[r, "\Phi_Y"] & Y \\ A_X \arrow[u, "\varphi"] \arrow[r, "\Phi_X"'] & X \arrow[u, "\phi"'] \end{tikzcd} \tag{4.17}$$

or equivalently, by the following equality:

$$\phi \circ \Phi_X(x, a_x) = \Phi_Y(\varphi(x, a_x)) \tag{4.18}$$

To show that such a map φ exists (and is uniquely defined) we note that it suffices to define it from G_X to G_Y. For any $a_x \in G_x$, $\varphi(x, a_x)$ is defined to be the unique element $(y, a_y) \in G_Y$ such that $\phi(x) = y$ and $\Phi_Y(y, a_y) = \varphi \circ \Phi_X(x, a_x)$. Such an element a_y always exists and is unique by definition of G_Y. We emphasize that the map φ is uniquely determined by the choice of the map ϕ. This fact will be important when dealing with hybrid control systems where this construction will be used several times. We resume the above discussion in the following result:

PROPOSITION 4.16. *Let Φ_X be an abstract control system over a set X with fibering monoid A_X freely generated. Given a surjective map $\phi : X \longrightarrow Y$, there exists a unique fibering monoid preserving lift $\varphi : A_X \longrightarrow A_Y$ of ϕ and a quotient abstract control system on Y with fibering monoid A_Y which is a φ-simulation of Φ_X.*

PROOF. The existence of Φ_Y and φ has been shown in the previous paragraph as well as the uniqueness of φ. We will only show that Φ_Y is a φ-simulation of Φ_X, which is a direct consequence of the commutativity of (4.17).

Assume that $x \xrightarrow{a_x} x'$ for some $a_x \in A_X$. The element a_x can be written as a product of generators as $a_x = g_x^1 g_x^2 \ldots g_x^n$ and in particular we have $n = 1$ if $a_x \in G_X$. The evolution $x \xrightarrow{a_x} x'$ can then be written as $x \xrightarrow{g_x^1} x_1 \xrightarrow{g_x^2} x_2 \xrightarrow{g_x^3} \ldots \xrightarrow{g_x^n} x'$. By construction of Φ_Y we know that we have:

$$\phi \circ \Phi_X(x, g_x^1) = \phi(x_1) = y_1 = \Phi_Y(y, g_y^1) = \Phi_Y(\varphi(x, g_x^1)) \tag{4.19}$$

But since $g_x^2 \in G_X$ we also have:

$$\phi \circ \Phi_X(x_1, g_x^2) = \phi(x_2) = y_2 = \Phi_Y(y_1, g_y^2) = \Phi_Y(\varphi(x_1, g_x^2)) \tag{4.20}$$

so that by making use of the semi-group property of abstract control systems we conclude that:

$$\begin{aligned} \Phi_Y(y, g_y^1 g_y^2) &= \Phi_Y(\Phi_Y(y, g_y^1), g_y^2) \\ &= \Phi_Y(y_1, g_y^2) \\ &= y_2 \\ &= \phi(x_2) \end{aligned} \tag{4.21}$$

A finite induction argument now shows that $y \xrightarrow{a_y} y'$ for $(y, a_y) = (y, g_y^1 g_y^2 \ldots g_y^n) = \varphi(x, g_x^1 g_x^2 \ldots g_x^n) = \varphi(x, a_x)$ and $\phi(x') = y'$ implying that Φ_Y φ-simulates Φ_X since for any $(x, y) \in \phi$ and any $(x, a_x) \in$ *dom*(φ) the tuple $((x, a_x), (y, a_y)) \in \varphi$ previously described satisfies $(\Phi_X(x, a_x), \Phi_Y(y, a_y)) \in \phi$. $\square$

The use of a fiber respecting map φ instead of a product respecting map shows a different perspective from the computer science approaches as described in [89]. This different approach is a consequence of modeling abstract control systems as deterministic systems which naturally requires extra flexibility when modeling state and input aggregation as illustrated in the next example.

Example 4.17. Consider the following fibering monoid $A_X = \{(x_1, a), (x_1, \varepsilon), (x_2, a), (x_2, \varepsilon), (x_3, \varepsilon), (x_4, \varepsilon)\}$ and $\Phi_X(x_1, a) = x_3$, $\Phi_X(x_2, a) = x_4$. If we model the state and input aggregation by a product respecting map of the form $\varphi = (\phi, \phi_{\mathcal{M}})$ with $\phi : X \to Y$ and $\phi_{\mathcal{M}} : \mathcal{M}_X \to \mathcal{M}_Y$ defined by:

$$\phi(x_1) = x_1 \quad \phi(x_2) = x_1 \quad \phi(x_3) = x_3 \quad \phi(x_4) = x_4$$
$$\phi_{\mathcal{M}}(a) = a \quad \phi_{\mathcal{M}}(\varepsilon) = \varepsilon$$

The abstraction would satisfy:

$$\Phi_Y(x_1, a) = \{x_3, x_4\} \tag{4.22}$$

which is clearly nondeterministic. This modeling problem can be overcome by using a fiber respecting map $\varphi : G_X \to G_Y$ defined by:

$$\varphi(x_1, \varepsilon) = (x_1, \varepsilon) \quad \varphi(x_1, a) = (x_1, a) \quad \varphi(x_2, \varepsilon) = (x_1, \varepsilon)$$
$$\varphi(x_2, a) = (x_1, b) \quad \varphi(x_3, \varepsilon) = (x_3, \varepsilon) \quad \varphi(x_4, \varepsilon) = (x_4, \varepsilon)$$

that assigns a different generator for each different state reachable from x_1.

Inspired by the results in [59] we characterize bisimilar systems in terms of the reachable space previously defined:

PROPOSITION 4.18. *Let Φ_X and Φ_Y be two abstract control systems over X and Y respectively. Given an equivalence relation $\sim \subseteq X \times X$, Φ_X is $R^\sim$-bisimilar to $\Phi_Y = \Phi_X / \sim$ iff:*

$$R_B^\sim\big(\mathcal{R}_x \Phi_X\big) = R_B^\sim\big(\mathcal{R}_{R_B^{\sim -1} \circ R_B^\sim(x)} \Phi_X\big) \tag{4.23}$$

PROOF. Assuming $R^\sim$-bisimilarity we will show that $R_B^\sim\big(\mathcal{R}_{R_B^{\sim -1} \circ R^\sim(x_1)} \Phi_X\big) \subseteq R_B^\sim\big(\mathcal{R}_{x_1} \Phi_X\big)$ since the other inclusion is obvious. Let $x_2 \in R_B^{\sim -1} \circ R_B^\sim(x_1)$ and $x_2' \in \mathcal{R}_{x_2}\Phi_X$, that is $x_2 \xrightarrow{a_{x_2}} x_2'$ for some $a_{x_2} \in G_{x_2}$. By the fact that Φ_Y is a $R^\sim$-simulation of Φ_X we get that $R_B^\sim(x_2) \xrightarrow{a_{R_B^\sim(x_2)}} R_B^\sim(x_2')$ for some $a_{R_B^\sim(x_2)} \in G_{R_B^\sim(x_2)}$. Using now the fact that Φ_X is a $R^{\sim -1}$-simulation of Φ_Y and $x_2 \in R_B^{\sim -1} \circ R_B^\sim(x_1)$ we conclude that $x_1 \xrightarrow{a_{x_1}} x_1'$ for some $a_{x_1} \in A_{x_1}$ and for a state x_1' such that $(x_1', x_2') \in R_B$. However by construction of Φ_Y, the preimages of $a_{R_B^\sim(x_2)}$ under $R^\sim$ have non empty intersection with G_X and therefore we can assume that $a_{x_1} \in G_X$ implying that $x_1' \in \mathcal{R}_{x_1}\Phi_X$. This allows to conclude that for any $x_2' \in \mathcal{R}_{R_B^{\sim -1} \circ R_B^\sim(x_1)}$ we have $R_B^\sim(x_2') = R_B^\sim\big(x_1'\big) \in R_B^\sim\big(\mathcal{R}_{x_1}\Phi_X\big)$ thereby showing the desired inclusion.

To show the converse, we recall that by Proposition 4.16 the quotient system Φ_Y is a simulation of Φ_X so that we only need to show that Φ_X $R^{\sim -1}$-simulates Φ_Y. Let $y' \in Reach_y \Phi_Y$, that is, there is a $a_y \in A_y$ such that $y \xrightarrow{a_y} y'$ and assume that $\phi(x) = y$ and $\phi(x') = y'$ (which can always be done since ϕ is a surjective map). The element a_y can be written as a finite multiplication of generators as $a_y = g_y^1 g_y^2 \ldots g_y^n$, where n equals 1 if $a_y \in G_Y$ and the evolution $y \xrightarrow{a_y} y'$ decomposes as $y \xrightarrow{g_y^1} y_1 \xrightarrow{g_y^2} y_2 \xrightarrow{g_y^3} \ldots \xrightarrow{g_y^n} y'$. By construction of $\Phi_X / \sim$ we have that g_y^1 is the image under $R^\sim$ of some $g_x^1 \in G_X$ and the equality $\phi \circ \Phi_X(x, g_x^1) = \Phi_Y(y, g_y^1)$ holds meaning that the evolution $y \xrightarrow{g_y^1} y_1$ is simulated by the evolution $x \xrightarrow{g_x^1} x_1$. But by the same argument the evolution $y_1 \xrightarrow{g_y^2} y_2$ is simulated by the evolution $x_1 \xrightarrow{g_y^2} x_2$ and the semi group property of abstract control systems allows to conclude that $\phi \circ \Phi_X(x, g_x^1 g_x^2) = \Phi_Y(y, g_y^1 g_y^2)$. An induction argument now shows that the evolution $y \xrightarrow{a_y} y'$ is simulated by the evolution $x \xrightarrow{a_x} x'$ with $a_x = g_x^1 g_x^2 \ldots g_x^n$ thereby showing that Φ_X $R^{\sim -1}$-simulates Φ_Y since $R^\sim(a_x) = R^\sim(g_x^1 g_x^2 \ldots g_x^n) = R^\sim(g_x^1) R^\sim(g_x^2) \ldots R^\sim(g_x^n) = g_y^1 g_y^2 \ldots g_y^n = a_y$. $\square$

At this level of generality this characterization of bisimulation is as unpractical as the definition since we have no means of computing the relevant *Reach* sets. However for discrete systems the *Reach* sets can be computed algorithmically and for continuous systems there are reasonable infinitesimal characterizations. When dealing specifically with hybrid control systems we will be able to give sufficient conditions for the desired equality between the relevant *Reach* sets.

3.7. Compositional Abstractions. In this section, we follow the categorical description of transition systems in [89], and introduce a notion of parallel composition for abstract control systems, then we determine under what conditions does this notion of parallel composition respect simulations and bisimulations.

3.7.1. *Parallel Composition with Synchronization.* The first step of composition combines two abstract control systems into a single one by forming their product. Given two abstract control systems $\Phi_X : A_X \longrightarrow X$ and $\Phi_Y : A_Y \longrightarrow Y$ we define their product to be the abstract control system $\Phi_X \times \Phi_Y : (A_X \times A_Y) \longrightarrow (X \times Y)$, $\Phi_X \times \Phi_Y(a_x, a_y) = (\Phi_X(a_x), \Phi_Y(a_y))$, where the fibers of $(A_X \times A_Y)$ are subsets of the direct product monoid $\mathcal{M}_X \otimes \mathcal{M}_Y$. The trajectories of the product control system consist of all possible combinations of the initial control systems trajectories. The product can also be defined in a categorical manner.

Definition 4.19 (Product of abstract control systems). Let $\Phi_X : A_X \longrightarrow X$ and $\Phi_Y : A_Y \longrightarrow Y$ be two abstract control systems. The product of these abstract control systems is a triple $(\Phi_X \times \Phi_Y, \pi_X, \pi_Y)$ where $\Phi_X \times \Phi_Y$ is an abstract control system and $\pi_X \subseteq (X \times Y) \times X$ and $\pi_Y \subseteq (X \times Y) \times Y$ are projection relations such that Φ_X is a π_X-simulation of $\Phi_X \times \Phi_Y$, Φ_Y is a π_Y-simulation of $\Phi_X \times \Phi_Y$, and for any other triple (Φ_Z, p_X, p_Y) of this type there is one and only one relation $\zeta \subseteq Z \times (X \times Y)$ such that $\Phi_X \times \Phi_Y$ is a ζ-simulation of Φ_Z, and the following diagram commutes:

$$
\begin{tikzcd}
\Phi_X & \Phi_X \times \Phi_Y \arrow[l, "\pi_X"'] \arrow[r, "\pi_Y"] & \Phi_Y \\
 & \Phi_Z \arrow[ul, "p_X"] \arrow[u, "\zeta"] \arrow[ur, "p_Y"'] &
\end{tikzcd}
\tag{4.24}
$$

The relations π_X and π_Y are in fact those induced by the canonical projection maps $\pi_X : X \times Y \longrightarrow X$, $\pi_Y : X \times Y \longrightarrow Y$ and the relation ζ is easily seen to be given by $\zeta = (p_X, p_Y)$. This definition of product may seem unnecessarily abstract and complicated at the first contact, it will, however, render the proof of the main result on the compatibility of parallel composition with respect to simulations an almost trivial task.

Example 4.20. Consider the transition system inspired from [89] and displayed on the left of Figure 3 where the ε evolutions are not represented. The product of these transitions systems will consist of all possible evolutions of both systems as displayed on the right of Figure 3.

In the product system we capture all possible trajectories of both systems and consequently several non physically meaningful trajectories. One allows, for example, input trajectories of the form (ε, u^t) where no time elapses in system Φ_X and t units of time elapse in system Φ_Y. These trajectories need to

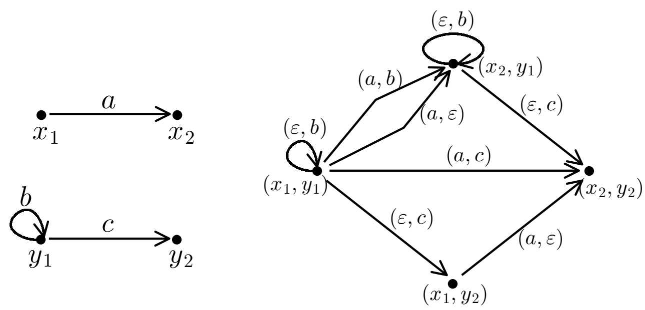

FIGURE 3. Two transition systems on the left and the corresponding product transition system on the right.

be removed from the product system in order to faithfully model a physical system. Another reason to remove transitions from the product system comes from the fact that in the product system, the behavior of one system does not influence the behavior of the other system. Since in general the behavior of a system composed of several subsystems depends strongly on the interaction between the subsystems, one tries to capture this interaction by removing undesired evolutions from the product system $\Phi_X \times \Phi_Y$ through the operation of restriction.

Given a fibering submonoid[3] $A_L \subseteq A_W$ we define the restriction of control system $\Phi_W : A_W \longrightarrow W$ to A_L as a new control system $\Phi_W|_{A_L} : A_L \longrightarrow L$ which is given by $\Phi_W|_{A_L}(x, a) = \Phi_W(x, a)$ iff $(x, a) \in A_L$ and $\Phi_W(x, a')$ belongs to L for any prefix a' of a. If the fibering submonoid A_L has the same base space as A_W but "smaller" fibers, then restriction is modeling synchronization of both systems on the control inputs. If on the other hand the fibers are equal but the base space of A_L is "smaller" then the base space of A_W then both systems are being synchronized on the state space. Synchronization on inputs and states is also captured by the operation of restriction by choosing a fibering submonoid with "smaller" fibers and base space. This operation also admits a categorical characterization.

DEFINITION 4.21 (Restriction of abstract control systems). Let $\Phi_W : A_W \longrightarrow W$ be an abstract control system, A_L a fibering submonoid of A_W and g and h two simulation relations such that $A_L = \{(w, a_w) \in A_W \mid g(w, a_w) = h(w, a_w)\}$. The restriction of Φ_W to A_L is a pair $(\Phi_W|_{A_L}, i_L)$ where $\Phi_W|_{A_L}$ is an abstract control system and $i_L \subseteq L \times W$ is an inclusion relation such that Φ_W is a i_L-simulation of $\Phi_W|_{A_L}$ satisfying $g \circ i_L = h \circ i_L$ and for any other pair (Φ_Z, i_Z) of this type such that $g \circ i_Z = h \circ i_Z$ there is one and only one relation η such that $\Phi_W|_{A_L}$ is a η-simulation of Φ_Z, and the following diagram

[3]A fibering submonoid A of a fibering monoid B is understood as a fibering monoid such that the inclusion map $i : A \hookrightarrow B$ is fibering monoid preserving.

$$(x_1, y_1) \xrightarrow{(a,b)} (x_2, y_1) \xrightarrow{(\varepsilon, c)} (x_2, y_2)$$

FIGURE 4. Parallel composition with synchronization of the transition systems displayed on the left of Figure 3.

commutes:

$$\begin{tikzcd} \Phi_W|_{A_L} \arrow[r, "i_L"] & \Phi_W \arrow[r, shift left, "g"] \arrow[r, shift right, "h"'] & \Phi_V \\ \Phi_Z \arrow[u, "\eta"] \arrow[ur, "i_Z"'] & & \end{tikzcd} \tag{4.25}$$

In general the domain of $\Phi_W|_{A_L}$, $\overline{A_L}$, may be strictly contained in A_L since restricting the base space implies also restricting the fibers to the actions that do not force the abstract control system to leave the restricted base. In any case the relation i_L is simply the inclusion $i_L(a_l) = a_l \in A_W$ for every $a_l \in \overline{A_L}$. With the notions of products and restriction at hand, we can now define a general operation of parallel composition with synchronization.

DEFINITION 4.22 (Parallel Composition with synchronization). Let $\Phi_X : A_X \to X$ and $\Phi_Y : A_Y \to Y$ be two abstract control systems and consider a fibering submonoid $A_L \subseteq A_X \times A_Y$. The parallel composition of Φ_X and Φ_Y with synchronization over A_L is the abstract control system denoted by $\Phi_X \|_{A_L} \Phi_Y$ and defined as:

$$\Phi_X \|_{A_L} \Phi_Y = (\Phi_X \times \Phi_Y)|_{A_L} \tag{4.26}$$

EXAMPLE 4.23. Consider the transition system displayed on the left of Figure 3. By specifying the subbundle:

$$\begin{aligned} A_L \quad = \quad & \{((x_1, y_1), (a, b)), ((x_1, y_1), (\varepsilon, \varepsilon)), ((x_1, y_1), (a, bc)), \\ & ((x_2, y_1), (\varepsilon, c)), ((x_2, y_1), (\varepsilon, \varepsilon)), ((x_2, y_2), (\varepsilon, \varepsilon))\} \end{aligned} \tag{4.27}$$

it is possible to synchronize the event a with the event b on the parallel composition of these systems. The resulting transition system is displayed in Figure 4. For purely continuous examples of parallel composition with synchronization we defer the reader to Chapter 5 where the abstractions of directed formations can be seen as the parallel composition of the individual agents with synchronization on the submanifold of the state space defined by the formation constraints. Note that contrary to the construction described in this section, in Chapter 5 only the control system is the parallel composition of the individual control systems, since the state space remains the product state space.

3.7.2. *Compositionality of Simulations.* We now determine if composition of subsystems is compatible with abstraction. A positive answer to this question is given by the next theorem which describes how the process of computing abstractions can be rendered more efficient by exploring the interconnection structure of hybrid systems.

THEOREM 4.24 (Compositionality of Simulations). *Given abstract control systems Φ_X, Φ_Z (which is a R_X-simulation of Φ_X), Φ_Y, Φ_W (which is a R_Y-simulation of Φ_Y) and the fibering submonoid $A_L \subseteq A_X \times A_Y$, the parallel composition of the simulations Φ_Z and Φ_W with synchronization over $R_{X\times Y}(A_L)$ is a $R_{X\times Y}|_{\overline{A_L}}$-simulation of the parallel composition of Φ_X and Φ_Y with synchronization over A_L, where $\overline{A_L} = dom(\Phi_X \parallel_{A_L} \Phi_Y)$.*

PROOF. Consider the product system $(\Phi_Z \times \Phi_W, \pi_Z, \pi_W)$ and the triple $(\Phi_X \times \Phi_Y, R_X \circ \pi_X, R_Y \circ \pi_Y)$. By definition of product we know that there is one and only one relation ζ such that:

$$\begin{tikzcd}
\Phi_Z & \Phi_Z \times \Phi_W \arrow[l, "\pi_Z"'] \arrow[r, "\pi_W"] & \Phi_W \\
 & \Phi_X \times \Phi_Y \arrow[ul, "R_X \circ \pi_X"] \arrow[u, "\zeta"] \arrow[ur, "R_Y \circ \pi_Y"'] &
\end{tikzcd}$$

commutes and this relation is given by $\zeta = (R_X, R_Y) = R_{X\times Y}$, meaning that $\Phi_Z \times \Phi_W$ is a $R_{X\times Y}$-simulation of $\Phi_X \times \Phi_Y$. Consider now the following diagram:

$$\begin{tikzcd}
(\Phi_Z \times \Phi_W)|_{\zeta(A_L)} \arrow[r, "i_{\zeta(A_L)}"] & \Phi_Z \times \Phi_W \arrow[r, shift left, "g"] \arrow[r, shift right, "h"'] & \Phi_V \\
(\Phi_X \times \Phi_Y)|_{A_L} \arrow[ur, "\zeta \circ i_{A_L}"'] & &
\end{tikzcd} \tag{4.28}$$

where g and h are equal on the fibering submonoid $\zeta(A_L)$. It is clear that $g \circ \zeta \circ i_{A_L} = h \circ \zeta \circ i_{A_L}$ since $\overline{A_L} \subseteq A_L$ implies $\zeta \circ i_{A_L}(\overline{A_L}) = \zeta(\overline{A_L}) \subseteq \zeta(A_L)$. Therefore there exists one and only one simulation relation η from $\Phi_X \parallel_{A_L} \Phi_Y$ to $\Phi_Z \parallel_{\zeta(A_L)} \Phi_W$ which is given by $\eta = \zeta \circ i_{A_L} = R_{X\times Y} \circ i_{A_L} = R_{X\times Y}|_{\overline{A_L}}$. □

The above result was stated for parallel composition of two abstract control systems but it can be easily extended to any finite number of abstract control systems. The relevance of the result lies in the fact that, in general, it is much easier to abstract each individual subsystem and by parallel composition obtain an abstraction of the overall system.

3.7.3. *Compositionality of Bisimulations.* We have already seen that bisimulation is a very powerful tool to deal with the complexity of large scale systems. In this subsection we will try to extend the previous compatibility results from simulations to bisimulations. We start with a very simple lemma stating that product respects bisimulations:

LEMMA 4.25. *Given abstract control systems Φ_X, Φ_Z (a R_X-bisimulation of Φ_X), Φ_Y and Φ_W (a R_Y-bisimulation of Φ_Y) the product abstract control system $\Phi_Z \times \Phi_W$ is a $R_{X\times Y}$-bisimulation of $\Phi_X \times \Phi_Y$.*

PROOF. Consider the following commutative diagrams:

$$\begin{tikzcd}
& \Phi_X \times \Phi_Y \arrow[dl, "\pi_X"'] \arrow[dr, "\pi_Y"] \arrow[ddd, "\eta_1"'] & \\
\Phi_X \arrow[d, "R_X"'] & & \Phi_Y \arrow[d, "R_Y"'] \\
\Phi_Z & & \Phi_W \\
& \Phi_Z \times \Phi_W \arrow[ul, "\pi_Z"] \arrow[ur, "\pi_W"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
& \Phi_X \times \Phi_Y \arrow[dl, "\pi_X"'] \arrow[dr, "\pi_Y"] & \\
\Phi_X & & \Phi_Y \\
\Phi_Z \arrow[u, "R_X^{-1}"'] & & \Phi_W \arrow[u, "R_Y^{-1}"'] \\
& \Phi_Z \times \Phi_W \arrow[ul, "\pi_Z"] \arrow[ur, "\pi_W"'] \arrow[uuu, "\eta_2"'] &
\end{tikzcd}
\tag{4.29}$$

By definition of product there exists one and only one relation η_1 and one and only one relation η_2 such that both diagrams commute. In fact, η_1 is the relation $\eta_1 = (R_X \circ \pi_X, R_Y \circ \pi_Y) = R_{X\times Y}$ and $\eta_2 = (R_X^{-1} \circ \pi_Z, R_Y^{-1} \circ \pi_W) = R_{X\times Y}^{-1}$ meaning that $\Phi_X \times \Phi_Y$ is $R_{X\times Y}$-bisimilar to $\Phi_Z \times \Phi_W$. $\square$

Although the product respects bisimulations the same does not happen with the operation of restriction. Consider the example displayed in Figure 5 where the abstract control system on top is bisimilar to the

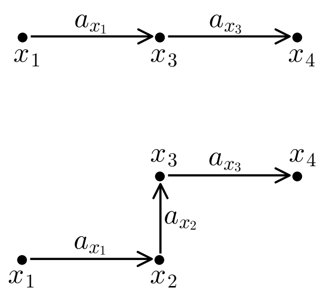

FIGURE 5. Bisimilar abstract control systems.

system below with respect to the relation:

$$\begin{aligned} R \;=\; & \{((x_1,\varepsilon),(x_1,\varepsilon)),((x_1,a_{x_1}),(x_1,a_{x_1})),((x_2,\varepsilon),(x_3,\varepsilon)),\\ & ((x_2,a_{x_2}),(x_3,\varepsilon)),((x_3,\varepsilon),(x_3,\varepsilon)),((x_3,a_{x_3}),(x_3,a_{x_3})),((x_4,\varepsilon),(x_4,\varepsilon))\} \end{aligned} \tag{4.30}$$

If we now restrict the fibers of the system below to the set $\{\varepsilon, a_{x_1}, a_{x_3}\}$ through the fibering submonoid:

$$A_L = \{(x_1,\varepsilon),(x_1,a_{x_1}),(x_2,\varepsilon),(x_3,\varepsilon),(x_3,a_{x_3}),(x_4,\varepsilon)\} \tag{4.31}$$

and restrict the fibers of the bisimilar system on top to $R(A_L)$ the systems will cease to be bisimilar since the system on top can move from x_3 to x_4 by a_{x_3} but the system below can not simulate that evolution when on $x_2 \in R_B^{-1}(x_3)$.

Assuming some extra structure on the relation R we can overcome this difficulty as stated in the following result:

Proposition 4.26. *Let Φ_X be an abstract control system, Φ_Y a R-bisimulation of Φ_X and A_L a fibering submonoid of A_X such that $R^{-1}|_{\overline{R(A_L)}} \circ R|_{\overline{A_L}} = id_{\overline{A_L}}$ and $R|_{\overline{A_L}} \circ R|_{\overline{A_L}}^{-1} = id_{\overline{R(A_L)}}$ for $\overline{A_L} = dom(\Phi_X|_{A_L})$ and $\overline{R(A_L)} = dom(\Phi_Y|_{R(A_L)})$. The restriction $\Phi_X|_{A_L}$ is a $R|_{\overline{A_L}}$-bisimulation of $\Phi_Y|_{R(A_L)}$.*

Proof. A similar argument to the proof of Proposition 4.24 shows that Φ_Y is a $R|_{\overline{A_L}}$-simulation of Φ_X so that we will only show that Φ_X is a $R|_{\overline{A_L}}^{-1}$-simulation of Φ_Y. Consider the following diagram:

$$\begin{tikzcd} \Phi_Y|_{R(A_L)} \arrow[dr, "R^{-1}\circ i_{R(A_L)}"] & & \\ \Phi_X|_{A_L} \arrow[r, "i_{A_L}"'] & \Phi_X \arrow[r, shift left, "g"] \arrow[r, shift right, "h"'] & \Phi_V \end{tikzcd} \tag{4.32}$$

where g and h are equal on the fibering submonoid A_L. We will show that (4.32) commutes by proving the only nontrivial equality, $g \circ R^{-1} \circ i_{R(A_L)} = h \circ R^{-1} \circ i_{R(A_L)}$. Recall that the assumptions $R^{-1}|_{\overline{R(A_L)}} \circ R|_{\overline{A_L}} = id_{\overline{A_L}}$ and $R|_{\overline{A_L}} \circ R|_{\overline{A_L}}^{-1} = id_{\overline{R(A_L)}}$ imply that $R^{-1}|_{\overline{R(A_L)}}$ and $R|_{\overline{A_L}}^{-1}$ are right and left inverses of $R|_{\overline{A_L}}$, respectively. However, by associativity of composition, inverses are unique and we must have $R^{-1}|_{\overline{R(A_L)}} = R|_{\overline{A_L}}^{-1}$ and $\overline{R(A_L)} = R(\overline{A_L})$. This allows to conclude that:

$$\begin{aligned} R^{-1} \circ i_{R(A_L)}(\overline{R(A_L)}) &= R^{-1} \circ i_{R(A_L)} \circ R(\overline{A_L}) \\ &= R^{-1}|_{\overline{R(A_L)}} \circ R|_{\overline{A_L}}(\overline{A_L}) = id_{\overline{A_L}}(\overline{A_L}) = \overline{A_L} \subseteq A_L \end{aligned} \tag{4.33}$$

Since (4.32) commutes we can invoke the definition of restriction to ensure the existence of a unique simulation relation from $\Phi_Y|_{R(A_L)}$ to $\Phi_X|_{A_L}$ which is given by $\eta = R^{-1} \circ i_{R(A_L)} = R^{-1}|_{\overline{R(A_L)}} = R|_{\overline{A_L}}^{-1}$ thereby showing bisimilarity. $\square$

The conditions $R^{-1}|_{\overline{R(A_L)}} \circ R|_{\overline{A_L}} = id_{\overline{A_L}}$ and $R|_{\overline{A_L}} \circ R|_{\overline{A_L}}^{-1} = id_{\overline{R(A_L)}}$ are very strong since they imply that $R|_{A_L}$ induces a set isomorphism between $\overline{A_L}$ and $\overline{R(A_L)}$. However this condition is in fact necessary as we now show:

PROPOSITION 4.27. *Let Φ_X be an abstract control system, Φ_Y a R-bisimulation of Φ_X and A_L a fibering submonoid of A_X. If the restriction $\Phi_X|_{\overline{A_L}}$ is a $R|_{\overline{A_L}}$-bisimulation of $\Phi_Y|_{R(A_L)}$ then $R^{-1}|_{\overline{R(A_L)}} \circ R|_{\overline{A_L}} = id_{\overline{A_L}}$ and $R|_{\overline{A_L}} \circ R|_{\overline{A_L}}^{-1} = id_{\overline{R(A_L)}}$, for $\overline{A_L} = dom(\Phi_X|_{A_L})$ and $\overline{R(A_L)} = dom(\Phi_Y|_{R(A_L)})$.*

PROOF. Consider the following commutative diagrams:

$$\begin{tikzcd}
\Phi_Y|_{R(A_L)} \arrow[r, "i_{R(A_L)}"] & \Phi_Y \arrow[d, "R^{-1}"] \\
\Phi_X|_{A_L} \arrow[u, "R|_{\overline{A_L}}"] \arrow[r, "i_{A_L}"] & \Phi_X
\end{tikzcd}
\qquad
\begin{tikzcd}
\Phi_Y|_{R(A_L)} \arrow[r, "i_{R(A_L)}"] \arrow[d, "R|_{\overline{A_L}}^{-1}"] & \Phi_Y \arrow[d, "R"'] \\
\Phi_X|_{A_L} \arrow[r, "i_{A_L}"] & \Phi_X
\end{tikzcd} \tag{4.34}$$

From the left diagram we get the equality:

$$\begin{aligned} i_{A_L} &= R^{-1} \circ i_{R(A_L)} \circ R|_{\overline{A_L}} \\ & R^{-1}|_{\overline{R(A_L)}} \circ R|_{\overline{A_L}} \end{aligned} \tag{4.35}$$

which gives $R^{-1}|_{\overline{R(A_L)}} \circ R|_{\overline{A_L}} = id_{\overline{A_L}}$ by restricting the codomains to $\overline{A_L}$. A similar argument for the diagram on the right allows to obtain $R|_{\overline{A_L}} \circ R|_{\overline{A_L}}^{-1} = id_{\overline{R(A_L)}}$. □

The above propositions lead to the following result concerning the compositionality of bisimulations:

THEOREM 4.28 (Compositionality of Bisimulations). *Given abstract control systems Φ_X, Φ_Z (a R_X-bisimulation of Φ_X), Φ_Y, Φ_W (a R_Y-bisimulation of Φ_Y) and a fibering submonoid $A_L \subseteq A_X \times A_Y$ we have that the parallel composition of the bisimulations Φ_Z and Φ_W with synchronization over $R_{X\times Y}(A_L)$ is a $R_{X\times Y}|_{\overline{A_L}}$-bisimulation of the parallel composition of Φ_X with Φ_Y with synchronization over A_L iff $R_{X\times Y}^{-1}|_{\overline{A_L}} \circ R_{X\times Y}|_{\overline{A_L}} = id_{\overline{A_L}}$ and $R_{X\times Y}|_{\overline{A_L}} \circ R_{X\times Y}|_{\overline{A_L}}^{-1} = id_{\overline{R_{X\times Y}(A_L)}}$ for $\overline{A_L} = dom(\Phi_X \parallel_{A_L} \Phi_Y)$ and $\overline{R_{X\times Y}(A_L)} = dom(\Phi_Z \parallel_{R(A_L)} \Phi_W)$.*

From the previous result we conclude that if we have a mean of computing bisimulations and if we choose the synchronization fibering submonoid carefully we can compute bisimulations by exploring the interconnecting structure of large-scale systems. In the next section we provide an algorithm to effectively compute abstractions and in certain situations bisimulations for hybrid control systems. We thus see that these results of compositionality of simulations and bisimulations provide efficient tools to handle the complexity of today's applications.

4. Hybrid Control Systems

4.1. Abstractions. Simulations of hybrid control systems are a simple instantiation of the previously introduced notion of simulation for abstract control systems. However, hybrid control systems usually come equipped with a set of initial conditions $X_0 \subseteq X$ which must also be related with the set of initial conditions of its simulation. The proper relation is expressed as follows:

DEFINITION 4.29 (Simulations of Hybrid Control Systems). Let $H_X = (X_0, X, A_X, \Phi_X)$ and $H_Y = (Y_0, Y, A_Y, \Phi_Y)$ be two hybrid control systems over X and Y respectively and let $R \subseteq A_X \times A_Y$ be a fibering monoid respecting relation. H_Y is a R-simulation of H_X iff:

1. $R_B(X_0) \subseteq Y_0$.
2. $\forall_{x \in X}\ (x,y) \in R_B \ \Rightarrow\ \forall_{(x,a_x) \in dom(R)}\ \exists_{(x,a_x,y,a_y) \in R} \qquad (\Phi_X(x,a_x), \Phi_Y(y,a_y)) \in R_B$.

The notion of abstraction is an instantiation of abstract control systems abstractions:

DEFINITION 4.30 (Abstractions of Hybrid Control Systems). Let H_X and H_Y be two hybrid control systems over X and Y respectively and let $R \subseteq A_X \times A_Y$ be a fiber respecting relation. H_Y is a R-abstraction of H_X iff R is a surjective relation with domain A_X and H_Y is a R-simulation of H_X.

as is the notion of bisimulation:

DEFINITION 4.31 (Bisimulation of Hybrid Control Systems). Let H_X and H_Y be two hybrid control systems over X and Y respectively and let $R \subseteq A_X \times A_Y$ be a fiber respecting relation. H_Y is R-bisimilar to H_X or a R-bisimulation of H_X iff H_Y is a R-simulation of H_X and H_X is a R^{-1}-simulation of H_Y.

4.2. Computing Abstractions. The goal of obtaining algorithmic procedures for computing abstractions guide us to more amenable characterizations of hybrid control systems. A first step in this direction is given by characterizing hybrid control systems in terms of its generators. From this point on we will simplify the notation by writing an element of $A_{(q,x)}$ as (q,x,a) instead of $((q,x),a)$.

PROPOSITION 4.32 (Hybrid Generators). *A set of initial conditions $X_0 \subseteq X$, a finite set of symbols Σ_X, a family of smooth fiber bundles $\pi_X^q : U_X^q \longrightarrow X_q$, a partially defined map $\delta_X : X \times \Sigma_X \longrightarrow X$ and a family of smooth control systems $F_X = \{F_X^q\}_{q \in Q}$, $F_X^q : U_X^q \longrightarrow TX_q$ defined on fiber bundle U_X^q over an open subset of X_q for each $q \in Q$ uniquely define a hybrid control system H_X. The maps δ_X and F_X are called the discrete and continuous generators of H_X, respectively.*

PROOF. We start by showing that δ_X extends uniquely to a partial map ${\delta_X}^* : X \times \Sigma_X^* \to X$. This action is obtained from δ_X by:

$$\begin{aligned} {\delta_X}^*(q,x,\varepsilon) &= (q,x) && (4.36)\\ {\delta_X}^*(q,x,\sigma_1\sigma) &= {\delta_X}^*({\delta_X}^*(q,x,\sigma_1),\sigma) \qquad \sigma_1\sigma \in \Sigma_X^*,\ \sigma_1 \in \Sigma_X && (4.37)\end{aligned}$$

defining ${\delta_X}^*$ uniquely since Σ_X^* is the monoid freely generated by Σ_X.

A similar construction is possible for F_X. Denote by C_q the projection under $\pi_X^q : U_X^q \to X_q$ of the open subset of X_q where each F_X^q is defined. A unique action ${F_X^q}^* : cl(C_q) \times {U_X^q}^* \to cl(C_q)$ can be obtained from F_X^q, where we denote by $cl(C_q)$ the closure of C_q in the topology of X_q. This is accomplished by defining ${F_X^q}^*$ as:

$$ {F_X^q}^*(x, u^{t'}) = \gamma_x(t') \tag{4.38}$$

where $\gamma_x(t)$ is the integral curve of the vector field $F_X^q(\gamma_x(t), u^t)$ satisfying $\gamma_x(0) = x$. By existence and uniqueness of integral curves of vector fields follows existence and uniqueness of the action ${F_X^q}^* : C_q \times U_q^* \to C_q$ since F_X^q is smooth. Moreover, we can extend ${F_X^q}^* : C_q \times U_q^* \to C_q$ to ${F_X^q}^* : cl(C_q) \times {U_X^q}^* \to cl(C_q)$ in a unique way by continuity since C_q is dense on $cl(C_q)$ and X_q is an Hausdorff, second countable topological space.

We can now combine ${\delta_X}^*$ and ${F_X^q}^*$ to get an hybrid control system $H_X = (X_0, X, A_X, \Phi_X)$ with $A_X \subseteq X \times \mathcal{M}$ and $\mathcal{M} = \coprod_{t \in \mathbb{N}_0}(U_X^* \cup \Sigma_X^*)^t$. Let $a \in U_X^* \cup \Sigma_X^*$ and define:

$$\Phi_X(q,x,a) = \begin{cases} (q,x) & \text{if} \quad a = \varepsilon \\ {\delta_X}^*(q,x,a) & \text{if} \quad a \in \Sigma_X^* \\ {F_X^q}^*(x,a) & \text{if} \quad a \in U_X^* \end{cases} \tag{4.39}$$

For a general $a \in \mathcal{M}$, split a into $a = a_1a_2$ with $a_1 \in U_X^* \cup \Sigma_X^*$, then Φ_X is given by:

$$\Phi_X(q,x,a) = \Phi_X(q,x,a_1a_2) = \Phi_X(\Phi_X(q,x,a_1),a_2) \tag{4.40}$$

and $\Phi_X(q,x,\varepsilon) = (q,x)$. This construction always provides a unique abstract control system Φ_X since we are using as monoid, the monoid freely generated by $U_X^* \cup \Sigma_X^*$ as asserted in Proposition 4.3. $\square$

This result tells us that it is enough to work with vector fields and single event jumps, which is how hybrid automata are usually defined in the literature [27]. In the light of this result we will also denote an hybrid control system by the tuple $H_X = (X, X_0, \Sigma_X, U_X, \delta_X, F_X)$. This representation of hybrid control systems will allow constructive methods to generate abstractions by combining discrete and continuous abstraction methodologies.

In order to benefit from the continuous abstraction methodology developed in [60, 63, 64] we will consider abstractions of hybrid control systems defined by equivalence relations on the state space. Other possible alternatives would consider equivalence relations on the inputs or on states and inputs. However, from a systems engineering point of view, it seems more natural to specify which state information should be ignored since the inputs are regarded as a means of obtaining the desired state behavior. This contrasts with the computer science approaches where the emphasis is put on the inputs which describe the behavior of the systems being analyzed through the language accepted by some automaton [29].

In this spirit, we start with a surjective map $\phi : X \longrightarrow Y$ which specifies the state aggregation. It will be useful to decompose ϕ into its discrete and continuous components. We shall denote by $\phi_D : X \longrightarrow P$ the discrete component of ϕ. Note that since we allow continuous to discrete aggregation the map ϕ_D does depend on X_q as well as on Q. Specifically, we assume that there is a finite covering of $\pi_X^q(dom(F_X^q)) \subseteq X_q$ for every $q \in Q$ denoted by $\Gamma_q = \{\Gamma_q^i\}_{i\in I}$ such that $\Gamma_q^i \cap \Gamma_q^j = \varnothing$ for $i \neq j$. We denote the set covering the point (q,x) by $\Gamma_q(x)$ and we call $\Gamma_q(x)$ adjacent to $\Gamma_q(x')$ iff $cl(\Gamma_q(x)) \cap cl(\Gamma_q(x')) \neq \varnothing$, where cl denotes the closure in the topology of X_q. Note that ϕ_D when restricted to the sets $\Gamma_q(x)$ simply gives the discrete state associated with the covering sets $\Gamma_q(x)$. We also introduce the set $\Pi \subseteq Q \times P$ for later use. It contains all the pairs of points (q,p) for which there exists a $x \in X_q$ such that $\phi_D(q,x) = p$. The continuous component of ϕ will be denoted by ϕ_C and consists of a family of smooth surjective submersions $\phi_C = \{\phi_{qp}\}_{(q,p)\in\Pi}$ with $\phi_{qp} : X_q \longrightarrow Y_p$. Having defined the state aggregation to be performed in the abstraction process we have also implicitly defined the surjective map $\varphi : A_X \longrightarrow A_Y$ relating the fibering monoids of the original system and its abstraction. This map is determined by the methods described in Subsection 3.6 and once again it is useful to have notation for its continuous and discrete components. The continuous part of φ, will be a family of smooth surjective fiber respecting maps $\varphi_C = \{\varphi_{qp}\}_{(q,p)\in\Pi}$, $\varphi_{qp} : U_X^q \longrightarrow U_Y^p$ which can be computed by the methods described in [78] and Chapter 3. The discrete component of φ, will be denoted by $\varphi_D = (\phi_D, \varphi_\Sigma)$.

Another important point to mention, and which is a consequence of the difference between continuous and discrete systems, is that although we have partitioned the sets $\pi_X^q(dom(F_X^q))$ into a finite number of subsets, the continuous flows generated by F_X^q can cross an infinite number of adjacent coverings sets in finite time. This will cause difficulties in the current framework since we are using as monoid the monoid free on the set $\Sigma_X^* \cup U_X^*$ which consists of finite length strings. We will, therefore, assume that the covering of $\pi_X^q(dom(F_X^q))$ is such that the flows generated by F_X^q only cross adjacent covering sets a finite number of times in any finite time interval. Any covering satisfying this assumption will be called finitely compatible with F_X^q. Sufficient conditions for finite compatibility, involving subanalytic stratifications for example, are given in [41]. This assumption can be dropped in two different scenarios:

- If there is no continuous to discrete aggregation,
- or if one extends the monoid $\mathcal{M}$ to a ω-monoid which can accommodate non finite length strings.

We now show how it is possible to specify a fibering monoid respecting relation based on the above maps. We start by defining several relations that will induce a unique fibering monoid respecting relation.

DEFINITION 4.33. Given a hybrid control system H_X and:

- A finite covering $\Gamma_q = \{\Gamma_q^i\}_{i\in I}$ by pairwise disjoint sets of $\pi_X^q(dom(F_X^q))$ finitely compatible with F_X^q for every $q \in Q$.
- A family of smooth surjective fiber preserving submersions $\varphi_C = \{\varphi_{qp}\}_{(q,p)\in\Pi}$, $\varphi_{qp} : U_X^q \longrightarrow U_Y^p$ induced by a family of smooth surjective submersions $\phi_C = \{\phi_{qp}\}_{(q,p)\in\Pi}$, $\phi_{qp} : X_q \longrightarrow Y_p$.
- A partial map $\varphi_\Sigma : X \times \Sigma_X^* \longrightarrow \Sigma_Y^*$, induced by a surjective map $\phi : X \longrightarrow Y$.

we define the following relations:

- $R_c^j \subseteq A_X \times A_Y$ for $j \in \Pi$, capturing continuous flows remaining inside a single covering set:

$$\big((q,x,u_x^t),(\phi_D(q,x),\varphi_{q\phi_D(q,x)}(x,u_x^t)\big) \in R_c^j \quad \text{iff} \quad \exists_{i\in I}\ \forall_{0<t'<t} \quad \Phi_X(q,x,u_x^{t'}) \in \Gamma_q^i \ \wedge\ (q,x,u_x^t) \in A_X \tag{4.41}$$

- $R_\varepsilon \subseteq A_X \times A_Y$, capturing the discrete jumps induced by the crossing of adjacent covering sets:

$$\big((q,x,\varepsilon),(p_j,y_j,\varepsilon)\big) \in R_\varepsilon \quad \forall_{j\in J} \tag{4.42}$$

$$\big((q,x,\varepsilon),(p_j,y_j,\sigma_{p_jp_k})\big) \in R_\varepsilon \quad \forall_{j\in J, j\neq k} \text{ where } \sigma_{p_jp_k} \in \Sigma_Y \text{ and } \Phi_Y(p_j,q_j,\sigma_{p_jp_k}) = (p_k,y_k) \tag{4.43}$$

iff the following holds:

$$\exists_{J\subseteq I} \quad (q,x) \in \bigcap_{j\in J} cl(\Gamma_q^j) \quad \wedge \quad \exists_{k\in J,\ t>0,\ u_x^t\in U_X^*} \quad \Phi_X(q,x,u_x^{t'}) \in \Gamma_q^k \quad \text{for all } t' \in]0,t] \tag{4.44}$$

$$\wedge\ \phi_D|_{\Gamma_q^j} = p_j \ \wedge\ \phi_{qp_j}(x) = y_j \quad \forall_{j\in J} \tag{4.45}$$

- $R_\sigma \subseteq A_X \times A_Y$, capturing all discrete jumps of H_X:

$$\big((q,x,\sigma),(\phi_D(q,x),\phi_{q\phi_D(q,x)}(x),\varphi_\Sigma(q,x,\sigma))\big) \in R_\sigma \quad \text{iff} \quad \sigma \in \Sigma_X^* \ \wedge\ (q,x,\sigma) \in A_X \tag{4.46}$$

These relations capture different aspects of an hybrid control system dynamics. We now show that there is a unique way of combining these different relations to determine a unique fibering monoid respecting relation with domain A_X.

PROPOSITION 4.34. *Under the assumptions of Definition 4.33 we have that $A_c^j = dom(R_c^j)$, $A_\varepsilon = dom(R_\varepsilon)$ and $A_\sigma = dom(R_\sigma)$ are fibering submonoids of A_X. Furthermore, given fibering monoid preserving relations $f_c^j \subseteq A_c^j \times A_Y$, $f_\varepsilon \subseteq A_\varepsilon \times A_Y$ and $f_\sigma \subseteq A_\sigma \times A_Y$ with domains A_c^j, A_ε and A_σ,* *respectively, there is one and only one fibering monoid preserving relation* $\eta \subseteq A_X \times A_Y$ *with domain* A_X *such that the following diagrams commute:*

$$
\begin{tikzcd}
A_c^j \arrow[r, "i_c^j"] \arrow[dr, "f_c^j"'] & A_X \arrow[d, "\eta"] \\
& A_Y
\end{tikzcd}
\qquad
\begin{tikzcd}
A_\varepsilon \arrow[r, "i_\varepsilon"] \arrow[dr, "f_\varepsilon"'] & A_X \arrow[d, "\eta"] \\
& A_Y
\end{tikzcd}
\qquad
\begin{tikzcd}
A_\sigma \arrow[r, "i_\sigma"] \arrow[dr, "f_\sigma"'] & A_X \arrow[d, "\eta"] \\
& A_Y
\end{tikzcd}
\tag{4.47}
$$

for $i_c^j \subseteq A_c^j \times A_X$*,* $i_\varepsilon \subseteq A_\varepsilon \times A_X$ *and* $i_\sigma \subseteq A_\sigma \times A_X$ *the inclusion relations and any fibering monoid* A_Y*.*

PROOF. We start by showing that A_c^j, A_ε and A_σ are fibering submonoids of A_X. Consider A_c^j first. If $(q,x) \in \cap_{j\in J} cl(\Gamma_q^j)$ then $\Phi_X(q,x,u_x^0)$ satisfies (4.41) and consequently $(q,x,\varepsilon) \in A_c^j$. Consider now any $(x,u_x^t) \in A_c^j$. By definition of R_c^j, u_x^t satisfies:

$$\Phi_X(q,x,u_x^{t'}) \in \Gamma_q^j \text{ for all } t' \in]0,t] \tag{4.48}$$

but this implies that $(x,u_x^{t'})$ also belongs to A_c^j for any $t' \in]0,t]$, that is any prefix of u_x^t also belongs to A_c^j since for $t' = 0$ we have $u_x^0 = \varepsilon$. A_c^j is therefore a fibering monoid since its fibers contain the identity and are prefix closed. The inclusion relation $i_c^j \subseteq A_c^j \times A_X$ taking $(q,x,a) \in A_c^j$ to $i_c^j(q,x,a) = (q,x,a) \in A_X$ renders A_c^j a fibering submonoid of A_X.

Consider now A_ε by definition of R_ε we have that for any $(q,x) \in dom(R_{\varepsilon B})$, the triple (q,x,ε) belongs to $dom(R_\varepsilon) = A_\varepsilon$. Consider now any $(q,x,a) \in A_\varepsilon$. Then $a \in \Sigma_X$ and any prefix of a is a it self or ε which both belong to A_ε making A_ε a fibering monoid and a fibering submonoid of A_X by the inclusion relation $i_\varepsilon \subseteq A_\varepsilon \times A_X$.

Finally $(q,x,\varepsilon) \in A_\sigma$ by (4.46) and the fact that $\varepsilon \in \Sigma_X^*$. If (q,x,σ) belongs to A_σ then any prefix σ' of σ also satisfies $(q,x,\sigma') \in A_\sigma$ since $\sigma' \in \Sigma_X^*$ and A_X has prefix closed fibers. Once again the inclusion relation makes A_σ a fibering submonoid of A_X.

We now show the existence of the relation $\eta \subseteq A_X \times A_Y$ with domain A_X by defining it. Let $(q,x,a) \in A_X$, then $a = a^1 a^2 \ldots a^n$ where the elements a^i belong to U_X^* and Σ_X^* in a alternate fashion. Without loss of generality we can assume that $a^1 \in U_X^*$ and therefore every a^{2i-1} for $i = 1,2,\ldots,n$ can be decomposed as a finite concatenation of elements of the form:

$$(q,x,a^{2i-1}) = (q,x,a_1^{2i-1})(q_2,x_2,\varepsilon)(q_2,x_2,a_2^{2i-1})(q_3,x_3,\varepsilon)\ldots(q_m,x_m,a_m^{2i-1}) \tag{4.49}$$

where each $(q_j,x_j,a_j^{2i-1}) \in A_c^j$ and $(q_{j+1},x_{j+1}) = \Phi_X(q_j,x_j,a_j^{2i-1})$. Replacing each element a^{2i-1} in $a^1 a^2 \ldots a^n$ by its string (4.49) still results in a finite string which we denote by:

$$(q_1,x_1,\alpha_1)(q_2,x_2,\alpha_2)\ldots(q_k,x_k,\alpha_k) \tag{4.50}$$

Note that this decomposition is unique and will allow to define η as follows:

$$
\begin{aligned}
((q,x,a),(p,y,a')) \in \eta \quad \text{iff} \quad & ((q,x,a),(p,y,a')) \in A_c^j \\
\vee \quad & ((q,x,a),(p,y,a')) \in A_\varepsilon \\
\vee \quad & ((q,x,a),(p,y,a')) \in A_\sigma \\
\vee \quad & (q,x,a) \in A_X \ \wedge \ a = (q_1,x_1,\alpha_1)(q_2,x_2,\alpha_2)\ldots(q_k,x_k,\alpha_k) \\
& \wedge \ a' = (p_1,y_1,\beta_1)(p_2,y_2,\eta_2)\ldots(p_k,y_k,\beta_k) \\
& \wedge \ \Big[((q_r,x_r,\alpha_r),(p_r,y_r,\beta_r)) \in R_c^j \ \vee \ ((q_r,x_r,\alpha_r),(p_r,y_r,\beta_r)) \in R_\varepsilon \\
& \vee \ ((q_r,x_r,\alpha_r),(p_r,y_r,\beta_r)) \in R_\sigma \text{ for } r = 1,\ldots,k\Big]
\end{aligned}
\tag{4.51}
$$

We now show that η is fibering monoid preserving. Let $((q,x),(p,y)) \in \eta_B$ then $((q,x,\varepsilon),(p,y,\varepsilon)) \in A_\sigma$ so that $((q,x,\varepsilon),(p,y,\varepsilon)) \in \eta$. Consider now the triples $(q,x,a),(q',x',a') \in A_X$ such that $(q,x,aa') \in A_X$ and let $((q,x,a),(p,y,b)),((q',x',a'),(p',y',b')) \in \eta$. Since $(q,x,aa') \in A_X$ and η is defined for every element in A_X we know that $(q,x,aa') \in dom(\eta)$. Decomposing aa' in its unique string described in (4.50) we get:

$$((q,x,\alpha_1\alpha_2\ldots\alpha_n\alpha_1'\alpha_2'\ldots\alpha_{n'}'),(p,y,\beta_1\beta_2\ldots\beta_n\beta_1'\beta_2'\ldots\beta_{n'}')) \in \eta \tag{4.52}$$

However, by definition of η we conclude:

$$((q,x,\alpha_1\alpha_2\ldots\alpha_n\alpha_1'\alpha_2'\ldots\alpha_{n'}'),(p,y,\beta_1\beta_2\ldots\beta_n\beta_1'\beta_2'\ldots\beta_{n'}')) = ((q,x,aa'),(p,y,bb')) \tag{4.53}$$

which shows that η is fibering monoid preserving.

To show uniqueness assume the existence of another relation η' satisfying all the proposition conditions. Then for any $(q,x,a) \in A_X$ we have $((q,x,a),(p,y,b)) \in \eta'$. If $(q,x,a) \in dom(A_c^j \cup A_\varepsilon \cup A_\sigma)$ then $\eta'(q,x,a) = \eta(q,x,a)$ by commutativity of diagrams (4.47). If $(q,x,a) \notin dom(A_c^j \cup A_\varepsilon \cup A_\sigma)$ then we can write a and b in its unique decompositions and since η' is fibering monoid respecting we have that $\eta'(q,x,a) = \eta'(q,x,\alpha_1)\eta'(q_2,x_2,\alpha_2)\ldots\eta'(q_k,x_k,\alpha_k)$ where each $(q_i,x_i,\alpha_i) \in dom(A_c^j \cup A_\varepsilon \cup A_\sigma)$ and consequently $\eta'(q_i,x_i,\alpha_i) = \eta(q_i,x_i,\alpha_i)$ so that we conclude equality between η' and η and the proof is finished. □

The unique relation induced by the relations R_c^j, R_ε and R_σ will be denoted by $\overline{R}$ and called an admissible relation for the remaining of this paper.

The reason why relations are necessary, and in particular the relation R_ε, can now be explained through an example.

Example 4.35. Consider a smooth control system (an hybrid control system with a single discrete state q) with state space covered by Γ_q^1 and Γ_q^2 and assume that the abstracting maps are given by $\phi_D|_{\Gamma_q^1} = p_1$, $\phi_D|_{\Gamma_q^2} = p_2$, $\phi_{qp_1} = id_{\Gamma_q^1}$ and $\phi_{qp_2} = id_{\Gamma_q^2}$. Suppose now that Γ_q^1 is open. Then a continuous flow controlled by $u_x^t = u_x^{t_1} u_x^{t_2}$ leaving Γ_q^1 and entering Γ_q^2 should be simulated by the abstraction as displayed in Figure 6, where continuous flows are represented by straight arrows and discrete jumps by an arc of circle arrows. The evolution on the abstraction is controlled by continuous flow $u_y^{t_1}$ on p_1 followed by a discrete jump from p_1 to p_2 and followed by another continuous flow $u_y^{t_2}$ on p_2. But since Γ_q^1 is open we cannot specify the point in $Y_{p^1} = \Gamma_q^1$ where the jump will take place. If one would attempt to define ϕ_C so as to send $cl(\Gamma_q^1) \cap cl(\Gamma_q^2)$ to Y_{p_1} and not to Y_{p_2} then the same problem would occur to a flow leaving Γ_q^2 and entering Γ_q^1. The natural way of overcoming these difficulties is by using a relation which sends $cl(\Gamma_q^1) \cap cl(\Gamma_q^2)$ to both Y_{p_1} and Y_{p_2}. Associated with this "nondeterminism" on the boundary points we also introduce "nondeterminism" at the level of control inputs. The relation R_ε sends ε at the boundary points to ε, but also sends ε to the discrete input $\sigma_{p_1 p_2}$ controlling a jump from p_1 to p_2. This allows to simulate the continuous flow on X controlled by u_x^t by the evolution on Y controlled by $u_y^{t_1} \sigma_{p_1 p_2} u_y^{t_2}$.

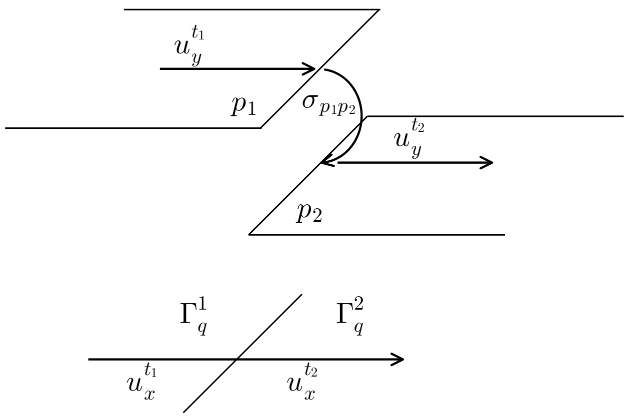

FIGURE 6. A continuous flow simulated by an hybrid abstraction.

Admissible relations allow us to effectively compute abstractions of hybrid control systems. A conceptual algorithm may be formulated as follows:

ALGORITHM 4.36 (Abstracting Algorithm).
Input data: $H_X = (X_0, X, \Sigma_X, U_X, \delta_X, F_X)$, $\overline{R} \subseteq A_X \times A_Y$
Body:

1. $Y := \overline{R}_B(X)$
2. $Y_0 := \overline{R}_B(X_0)$
3. $\Sigma_Y := \varphi_\Sigma(X \times \Sigma_X) \cup \{\sigma \ : \ \exists\, ((q,x,\varepsilon),(p,y,\sigma)) \in \overline{R}\}$
4. $U_Y := \{U_Y^p\}_{p \in P}$, $U_Y^p = \varphi_{pq}(U_X^q)$
5. $J = \{(p, y, \sigma_{pp'}, p', y') \ : \ \exists (q,x) \in \cap_{k \in K} cl(\Gamma_q^k) \quad \exists u \in U_X^q(x)$ such that $F_X^q(u)$ is transversal to the boundary of Γ_q^i, points to Γ_q^i, $\big((q,x),(p,y)\big) \in \overline{R}_B$, $p \neq \phi_D|_{\Gamma_q^i}$ and $\big((q,x),(p',y')\big) \in \overline{R}_B$, $p' = \phi_D|_{\Gamma_q^i}\}$
6. $\delta_Y := (\phi_D, \phi_{q\phi_D}, \varphi_\Sigma, \phi_D, \phi_{q\phi_D})(\delta_X) \cup J$ where δ_X is regarded as the set $\delta_X \subseteq X \times \Sigma_X \times X$.
7. $F_Y^p :=$ is the φ_{qp}-abstraction of F_X^q for every $(q,p) \in \Pi$.

Output data: $H_Y = (Y_0, Y, \Sigma_Y, U_Y, \delta_Y, F_Y)$

Intuitively the above algorithm can be described as follows. Steps 1 and 2 simply define Y and Y_0 as the image under $\overline{R}_B$ of X and X_0, respectively. In step 3 the set of labels Σ_Y is computed as the image under φ_Σ of $X \times \Sigma_X$ and all the symbols $\sigma_{pp'}$ created when the continuous flows crosses the boundary between adjacent covering sets. In step 4 the continuous control bundle is computed as the image of U_X^q under each map φ_{qp}. In step 5 the set J is computed to be used on the next step. Step 6 determines δ_Y in a way that can be described as follows: for every transition $(q,x) \xrightarrow{\sigma} (q',x')$ defined by δ_X there will be a transition $(\phi_D(q,x), \phi_{q\phi_D(q,x)}(x)) \xrightarrow{\varphi_\Sigma(q,x,\sigma)} (\phi_D(q',x'), \phi_{q'\phi_D(q',x')}(x'))$ expressed by the set $(\phi_D, \phi_{q\phi_D}, \varphi_\Sigma, \phi_D, \phi_{q\phi_D})(\delta_X)$, where δ_X is regarded as a subset of $X \times \Sigma_X \times X$. Furthermore, every time a continuous flow crosses the boundary between adjacent covering sets, the required discrete transitions are captured by the set J. Finally in the last step the continuous generator of H_Y is obtained from the continuous generator of H_X by the methods described in [60, 64] and reviewed in Chapter 3.

The above algorithm does compute a simulation of H_X as asserted in the next theorem:

THEOREM 4.37. *Let H_X be an hybrid control system over X and $\overline{R} \subseteq A_X \times A_Y$ an admissible relation. Then hybrid control system H_Y obtained through Algorithm 4.36 is a $\overline{R}$-abstraction of H_X.*

PROOF. We will split the proof into four distinct parts. We start by showing that H_Y simulates every discrete jump of H_X, next we show that H_Y also simulates every continuous flow of H_X that remains inside a single covering set. On the third part we show that continuous flows crossing adjacent covering sets are also simulated by H_Y and finally we will use the preceding results to show that any finite sequence of continuous flows and discrete jumps is also simulated by H_Y.

Discrete Jumps

By construction, δ_Y simulates δ_X so that every discrete jump of H_X is simulated by H_Y.

Continuous flows inside a single covering set and starting on a interior point

If the flow of F_X^q remains inside a single covering set and starts on a interior point, then the smooth abstraction results in [60, 64] show that F_Y^p generates a continuous flow that simulates the flow generated by F_X^q.

Continuous flows inside a single covering set and starting on a boundary point

Let $(q,x) \in \cap_{k\in K} cl(\Gamma_q^k)$ and assume that $\Phi_X((q,x),u_x^{t'}) \in \Gamma_q^i$ for all $0 < t' < t$. This implies that there exists a $u \in U_X^q(x)$ such that $F_X^q(u)$ is transversal to the boundary of Γ_q^i and points to Γ_q^i. Consequently, steps 5 and 6 of Algorithm 4.36 ensure that for any point (p,y) $\overline{R}_B$-related to (q,x), there is a $a_{(p,y)} \in A_{(p,y)}$ such that $(p,y) \xrightarrow{a_{(p,y)}} (p_i,y_i)$, where $((q,x),(p_i,y_i)) \in \mathcal{R}_B$ and $p_i = \phi_D|_{\Gamma_q^i}$. If $x \in \Gamma_q^i$, then by the previous paragraph $\Phi_X(q,x,u_x^t)$ is simulated by $\Phi_Y(p_i,y_i,u_y^t)$ with $u_y^t = \varphi_{qp_i}(u_x^t)$. If $x \notin \Gamma_q^i$, then $x \in \Gamma_q^j$ for some $j \neq i$ and $j \in K$. Also by the previous paragraph we have that $\Phi_X(q,x,u_x^t)$ is simulated by $\Phi_Y(p_j,y_j,a_{(p_j,y_j)}u_y^t)$.

Continuous flows crossing adjacent covering sets

Let u_x^t be a continuous input such that $\Phi_X(q,x,u_x^t)$ crosses the boundary between adjacent covering sets once at $t = t_1$. We decompose u_x^t into $u_x^t = u_x^{t_1}u_x^{t_2}$ with $t_2 = t - t_1$. Since $\Phi_X(q,x,u_x^{t_1})$ remains on the interior of a single covering set we have $(\Phi_X(q,x,u_x^{t_1}),\Phi_Y(p,y,u_y^{t_1})) \in \overline{R}_B$. Now let $(q',x') = \Phi_X(q,x,u_x^{t_1})$. It is not difficult to see that (q',x') belongs to the boundary between adjacent covering sets. By the previous paragraph $\Phi_X(q',x',u_x^{t_2})$ is simulated by $\Phi_Y(\phi_D(q',x'),\phi_{q'\phi_D(q',x')},a_{(\phi_D(q',x'),\phi_{q'\phi_D(q',x')})}u_y^{t_2})$ so that $\Phi_X(q,x,u_x^t) = \Phi_X(q,x,u_x^{t_1}u_x^{t_2})$ is simulated by $\Phi_Y(\phi_D(q,x),\phi_{q\phi_D(q,x)},u_y^{t_1}a_{(\phi_D(q',x'),\phi_{q'\phi_D(q',x')})}u_y^{t_2})$.

Since a continuous input making Φ_X cross adjacent covering sets several times can be decomposed into a finite product of several continuous inputs making Φ_X cross adjacent covering sets only once, the previous argument extends to all continuous inputs by induction.

Any finite sequence of discrete jumps and continuous flows

Consider a $a \in A_X$. This element can be decomposed into a finite concatenation of elements belonging to Σ_X^* and U_X^*. Since every such element can be simulated by H_Y we can extend in a unique way Φ_Y defined for $U_X^* \cup \Sigma_X^*$ to finite length sequences, since $\mathcal{M}$ is the monoid freely generated by $U_X^* \cup \Sigma_X^*$ as asserted in Proposition 4.3. $\square$

EXAMPLE 4.38. As an illustration of the construction given by Algorithm 4.36 we present a simple example adapted from [37, 23]. Consider a simple model of a six legged mechanical insect as displayed in Figure 7.

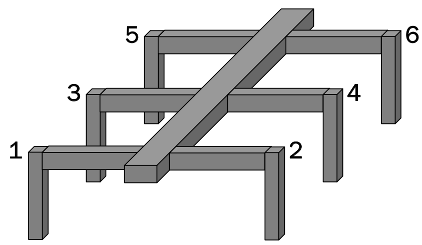

FIGURE 7. Six legged mechanical insect.

The control system associated with this mechanical system can be described by:

$$\begin{aligned}
\dot{x}_1 &= \cos\theta(\alpha(h_1)u_1 + \beta(h_2)u_2) \\
\dot{x}_2 &= \sin\theta(\alpha(h_1)u_1 + \beta(h_2)u_2) \\
\dot{\theta} &= l\alpha(h_1)u_1 - l\beta(h_2)u_2 \\
\dot{\xi}_1 &= u_1 \\
\dot{\xi}_2 &= u_2 \\
\dot{h}_1 &= u_3 \\
\dot{h}_2 &= u_4
\end{aligned}$$

where the functions α and β are defined as:

$$\alpha(h_1) = \begin{cases} 1 & \Leftarrow \quad h_1 = 0 \\ 0 & \Leftarrow \quad h_1 > 0 \end{cases} \qquad \beta(h_2) = \begin{cases} 1 & \Leftarrow \quad h_2 = 0 \\ 0 & \Leftarrow \quad h_2 > 0 \end{cases} \tag{4.54}$$

The variables in the above control system have the following interpretation:

x_1 and x_2	position of the insect center of mass.
θ	insect orientation with respect to some fixed reference frame.
ξ_1	angle of the legs 1, 4 and 5 with respect to the insect central body.
ξ_1	angle of the legs 2, 3 and 6 with respect to the insect central body.
h_1	height of the legs 1, 4 and 5 with respect to the floor.
h_2	height of the legs 2, 3 and 6 with respect to the floor.
u_1, u_2, u_3 and u_4	control inputs.

It is assumed that the robot moves the legs in a alternate fashion, that is legs 1,4 and 5 move together and then legs 2,3 and 6 move together and this pattern is repeated to achieve insect motion. It is furthermore assumed that the legs execute synchronous motions so that they can be described by their equal height h_i and angle ξ_i. When all legs are in contact with the floor, that is $h_1 = 0 = h_2$, all contribute to the motion of the insect through inputs u_1 and u_2. If $h_1 > 0$ and $h_2 = 0$ only the legs 2,3 and 6 are on the floor influencing the insect motion. On the other hand, when only legs 1,4 and 5 are on the floor only input u_2 influences the insect motion. Finally there is still an uninteresting case which corresponds to all the legs being on the air which we shall not consider. If we denote by q_1 the state where all legs are on the floor and by $f_X^{q_1}$ the corresponding control system in local coordinates, q_2 the state where only legs 1,4 and 5 are in contact with the floor and $f_X^{q_2}$ the corresponding control system and q_3 the state where legs 2,3 and 6 are on the floor and by $f_X^{q_3}$ the associated control system we can model the insect controlled kinematics by the hybrid control system displayed in Figure 8.

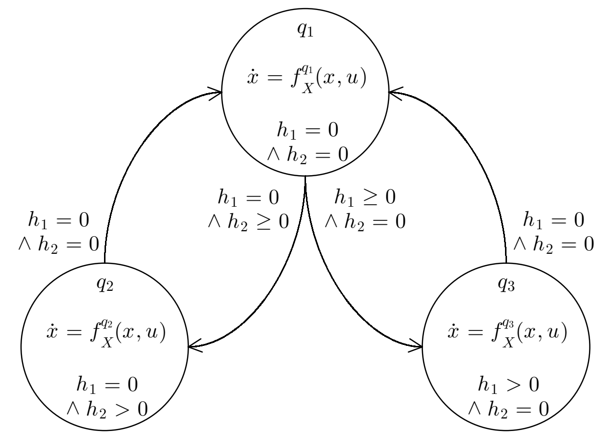

FIGURE 8. Hybrid control system model of the mechanical insect displayed in Figure 7.

Suppose now that there is a team of several mechanical insects that needs to be collectively controlled to perform some task. If the number of insects is large it becomes unfeasible to coordinate the motion of all the legs among the whole team. The advocated solution to overcome the complexity of such a problem is to perform an abstraction of the insect model so as to design the coordination in a more efficient way. A natural choice is to retain on the abstracted model only information about the insect position and to abstract away the switching policy necessary for the insect motion. This leads to the following choice for the state aggregation maps where by x we denote a point in X_q:

$$\phi_D(q_1, x) = p \quad \phi_D(q_2, x) = p \quad \phi_D(q_3, x) = p \tag{4.55}$$

and

$$\phi_{q_1 p}(x) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \phi_{q_2 p}(x) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad \phi_{q_3 p}(x) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{4.56}$$

This choice implies that the abstraction will be a hybrid control system with a single discrete state p and only two continuous variables x_1 and x_2 modeling the insect position. Assuming that the initial state of the hybrid control system is X since the insect can start in any discrete and continuous location, we follow the steps of Algorithm 4.36 to obtain:

$$\begin{aligned} Y &:= \{p\} \times \mathbb{R}^2 = \overline{R}_B(X) \\ Y_0 &:= Y = \overline{R}_B(X) = \overline{R}_B(X_0) \\ \Sigma_Y &:= \{\varepsilon\} = \varphi_\Sigma(X \times \{\varepsilon, \sigma_{q_1 q_2}, \sigma_{q_2 q_1}, \sigma_{q_1 q_3}, \sigma_{q_3 q_1}\}) = \varphi_\Sigma(X \times \Sigma_X) \cup \varnothing \end{aligned}$$

The control bundle U_Y^p is computed by the methods in [78] and Chapter 3 and equals $Y \times \mathbb{R}^2$. On step 5 J is computed to be the empty set since there is only one covering set for each set $\pi_X^q(dom(F_X^q))$. Step 6 determines the map δ_Y which is simply given by:

$$\delta_Y(y, \varepsilon) = y \tag{4.57}$$

since $\Sigma_Y = \{\varepsilon\}$. Finally the continuous abstraction of each F_X^q is computed by the methods described in [64] and is given by:

$$\begin{aligned} \dot{y}_1 &= v_1 \\ \dot{y}_2 &= v_2 \end{aligned} \tag{4.58}$$

where v_1 and v_2 are control inputs. This simple example shows the power of the abstraction methodology by reducing the hybrid automaton in Figure 8 to two integrators. The abstraction is clearly a much simpler and useful model to design the coordinated motion of a team of such robotic insects.

4.3. From hybrid abstractions to hybrid bisimulations. In this section we try to determine when can we use Algorithm 4.36 to compute a bisimulation. By taking advantage of the special structure of admissible relations we will be able to provide checkable sufficient conditions for bisimilarity. We start by relating simulation with respect to relations defined only for A_c^j, A_ε and A_σ with relations defined for A_X.

PROPOSITION 4.39. *Let H_X and H_Y be hybrid control systems and assume that H_Y is a R_c^j-simulation, a R_ε-simulation and a R_σ-simulation of H_X. Then H_Y is also a $\overline{R}$-simulation of H_X, where $\overline{R}$ is the unique relation with domain A_X defined by R_c^j, R_ε and R_σ. Furthermore, if one replaces each relation with its inverse relation the result still holds.*

PROOF. We only need to show that for any $(q,x,a) \in A_X$ such that $(q,x,a) \notin dom(R_c^j \cup R_\varepsilon \cup R_\sigma)$ there is a pair $((q,x,a),(p,y,b)) \in \overline{R}$ such that $((q,x),(p,y)) \in \overline{R}_B$ and $(\Phi_X(q,x,a), \Phi_Y(p,y,b)) \in \overline{R}_B$. Decompose a in the unique sequence $(q,x,a) = (q_1,x_1,\alpha_1)(q_2,x_2,\alpha_2)\ldots(q_n,x_n,\alpha_n)$ with $(q_{i+1},p_{i+1}) = \Phi_X(q_i,p_i,\alpha_i)$ and $\alpha_i \in dom(R_c^j \cup R_\varepsilon \cup R_\sigma)$ for $i = 1,\ldots,n$ as described in the proof of Proposition 4.34. Since each (q_i,p_i,α_i) belongs to $dom(R_c^j \cup R_\varepsilon \cup R_\sigma)$ we have that $(\Phi_X(q_1,x_1,\alpha_1), \Phi_Y(p_1,y_1,\beta_1)) \in \overline{R}$, $(\Phi_X(q_2,x_2,\alpha_2), \Phi_Y(p_2,y_2,\beta_2)) \in \overline{R}$ but $\Phi_X(q_1,p_i,\alpha_1) = (q_2,x_2)$ so that by the semi group property of abstract control systems we have:

$$(\Phi_X(q_1,x_1,\alpha_1\alpha_2), \Phi_Y(p_1,y_1,\beta_1\beta_2)) \in \overline{R}_B \tag{4.59}$$

By induction we conclude that $(\Phi_X(q_1,x_1,\alpha_1\alpha_2\ldots\alpha_n), \Phi_Y(p_1,y_1,\beta_1\beta_2\ldots\beta_n)) \in \overline{R}_B$ showing that for any $(q,x,a) \in A_X$ there is a $((q,x,a),(p,y,b)) \in \overline{R}$ such that $((\Phi_X(q,x,a), \Phi_Y(p,y,b)) \in \overline{R}_B$ and concluding that H_Y is a $\overline{R}$-simulation of H_X.

The same argument also shows that the result still holds if the relations are replaced by the corresponding inverse relations. □

The previous result allows to give a sufficient condition for bisimilarity which is based on the conditions given for abstract control systems:

PROPOSITION 4.40. *Let H_X be an hybrid control system, $\overline{R}$ an admissible relation and H_Y a $\overline{R}$-abstraction obtained through Algorithm 4.36. If the equality:*

$$\overline{R}_B(\mathcal{R}_{(q,x)}H_X) = \overline{R}_B(\mathcal{R}_{\overline{R}_B^{-1}\circ\overline{R}_B(q,x)}H_X) \tag{4.60}$$

holds then H_Y is $\overline{R}$-bisimilar to H_X.

PROOF. We recall that H_Y is a $\overline{R}$-simulation of H_X by Theorem 4.37 so that we need only to show that H_X $\overline{R}^{-1}$-simulates H_Y. The proof will be done by showing that under the proposition hypotheses H_X is a R_c^{j-1}-simulation and a R_ε^{-1}-simulation and a R_σ^{-1}- simulation of H_Y so that by Proposition 4.39 H_X will also be a $\overline{R}^{-1}$-simulation of H_Y.

We start by analyzing R_c^j using Proposition 4.16 with the restriction of H_X to A_c^j denoted by $H_X|_{A_c^j}$. This is accomplished by noting that H_X is a i_c^j-simulation of $H_X|_{A_c^j}$ where i_c^j is the inclusion morphism from $H_X|_{A_c^j}$ to H_X. The set $R_c^j(s)$ is a singleton for every $s \in dom(R_c^j)$ so that the relation R_c^j induces the fibering monoid preserving map $f_{R_c^j} : A_c^j \longrightarrow A_Y$. This map is in fact induced by the base map f_{cB}^j (defined by the base relation R_{cB}^j) through the methods described in Section 3.6 and we can apply Proposition 4.16 to $H_X|_{A_c^j}$ to conclude that if:

$$f_c^j(\mathcal{R}_{(q,x)}H_X|_{A_c^j}) = f_c^j(\mathcal{R}_{f_{cB^j}\circ f_{cB}^{j}{}^{-1}(q,x)}H_X|_{A_c^j}) \tag{4.61}$$

holds then $H_X|_{A_c^j}$ is a $f_c^{j^{-1}}$-simulation of H_Y. However, the assumptions of the theorem imply (4.61) therefore $H_X|_{A_c^j}$ is in fact a $f_c^{j^{-1}}$-simulation of H_Y. By composing $f_c^{j^{-1}}$ with the inclusion morphism i_c^j, we conclude that $i_c^j \circ f_{s_j}^{-1} = R_c^{j^{-1}}$ is a morphism from H_X to H_Y showing that H_X is a $R_c^{j^{-1}}$-simulation of H_Y.

The argument for the relation R_σ is similar to the one for the relations R_c^j.

Finally we need to show that H_X is a R_ε^{-1}-simulation of H_Y. We recall that the relation R_ε captures the discrete jumps on H_Y introduced to model the switching between discrete states caused by the crossing of adjacent covering sets on $\pi_X^q(dom(F_X^q))$ by continuous flows. Let $((q,x),(p,y)) \in R_{\varepsilon B}$ and let $(p,y,a) \in Range(R_\varepsilon)$. Then $a = \sigma_{pp'}$, $((q,x),(\Phi_Y(p,y,a)) \in R_{\varepsilon B}$ by construction of $\overline{R}$ and definition of R_ε. Furthermore $((q,x,\varepsilon),(p,y,\sigma_{pp'})) \in R_\varepsilon$ also by construction of R_ε, but then for every $((q,x),(p,y)) \in R_{\varepsilon B}$ we have $\Phi_X(q,x,\varepsilon) = (q,x)$ showing $(\Phi_X(q,x,\varepsilon),\Phi_Y(p,y,\sigma_{pp'})) \in R_{\varepsilon B}$ and implying that H_X is a R_ε^{-1}-simulation of H_Y. The proof is now finished. □

We now replace the condition of the previous result by conditions that are checkable in concrete examples.

THEOREM 4.41. *Let H_X be an hybrid control system, $\overline{R}$ an admissible relation and H_Y a $\overline{R}$-abstraction obtained through Algorithm 4.36. If:*

- *the guards intersecting $\pi_X^q(dom(F_X^q))$ are invariant for $Ker(T\phi_{qp})$;*
- *the reset maps satisfy $\phi_{q'p'}(Reset_{qq'}(\phi_{qp}^{-1} \circ \phi_{qp}(x))) = \phi_{q'p'}(Reset_{qq'}(x))$ for all $q,q' \in Q$ and $(q,p),(q',p') \in \Pi$.*
- *F_X^q is controlled invariant for $Ker(T\phi_{qp})$*
- *There is only one covering set for each set $\pi_X^q(dom(F_X^q))$.*

then H_Y is a $\overline{R}$-bisimulation of H_X

PROOF. The first condition ensures that every point belonging to preimage of $y \in Y_p$ by ϕ_{qp} has the same jumping capabilities since the guards are enabled or disabled for all those points. This ensures that the discrete part of the states reachable by the system H_X, when controlled by an element in Σ_X, is the same for every point in $\overline{R}_B^{-1} \circ \overline{R}_B(q,x)$. To ensure that the continuous part is also the same, we invoke the second condition that ensures $\overline{R}_B(Reset_{qq'}(x)) = \overline{R}_B(Reset_{qq'}(\overline{R}_B^{-1} \circ \overline{R}_B(x)))$. We have thus shown that we have:

$$\overline{R}_B(H_X(q,x,\sigma)) = \overline{R}_B(H_X(R_B^{-1} \circ \overline{R}_B(q,x),\sigma)) \quad \forall_{\sigma \in \Sigma_Y} \tag{4.62}$$

Since Σ_X^* is freely generated by Σ_X we only need to show that for every $u^t \in U_X^*$ we also have $\overline{R}_B(H_X(q,x,u^t)) = \overline{R}_B(H_X(\overline{R}_B^{-1} \circ \overline{R}_B(q,x),u^t))$. From Theorem 3.16 in Chapter 3 we know that controlled invariance is equivalent to projectabilty of the control section and this implies that for every

$(q,x) \in \overline{R}_B^{-1} \circ \overline{R}_B(q,x)$ the control section is the same modulo $Ker(T\phi_{qp})$. This is simply the infinitesimal statement of:

$$\phi_{qp} \circ \pi_{X_q}(H_X(q,x,u^t)) = \phi_{qp} \circ \pi_{X_q}(H_X(\overline{R}_B^{-1} \circ \overline{R}_B(q,x),u^t)) \quad \forall_{u^t \in U_X^{q*}} \tag{4.63}$$

where we have denoted by π_{X_q} the natural projection from X to X_q taking $(q,x) \in X$ to $x \in X_q$. By an argument similar to Theorem 3.7 in [60] it can be shown that controlled invariance implies (4.63).

We now use the last assumption of the theorem to ensure that:

$$\phi_D(H_X(q,x,u^t)) = \phi_D(H_X(\overline{R}_B^{-1} \circ \overline{R}_B(q,x),u^t)) \quad \forall_{u^t \in U_X^{q*}} \tag{4.64}$$

which follows from the fact that all the states $(q,x) \in \overline{R}_B^{-1} \circ \overline{R}_B(q,x)$ are mapped to the same discrete state since there is only one covering set for each set $\pi_X^q(dom(F_X^q))$. Equation (4.63) together with (4.64) in turn imply that:

$$\overline{R}_B(H_X(q,x,u^t)) = \overline{R}_B(H_X(\overline{R}_B^{-1} \circ \overline{R}_B(q,x),u^t)) \quad \forall_{u^t \in U_X^{q*}} \tag{4.65}$$

The desired equality:

$$\overline{R}_B(\mathcal{R}_{(q,x)}H_X) = \overline{R}_B(\mathcal{R}_{\overline{R}_B^{-1} \circ \overline{R}_B(q,x)}H_X) \tag{4.66}$$

now follows from the fact that $\mathcal{M}$ is freely generated by Σ_X and U_X^* and the result is a consequence of Proposition 4.40. $\square$

This result provides easily checkable conditions for bisimilarity, however controlled invariance is a strong requirement. Weaker conditions for bisimilarity between hybrid control systems can be achieved if one uses weaker notions of bisimulation such as weak bisimulation [52], however those results rely on a complete and thorough understanding of bisimilarity for continuous control systems which is still an area of current research.

4.4. Preservation and Reflection of Properties. In this section we will specialize the results of Subsection 3.5 to hybrid control systems and consider properties that are specific of hybrid systems such as the Zeno phenomena.

4.4.1. *Blocking.* Blocking was already discussed in Subsection 3.5 where a necessary and sufficient result for preservation of non-blocking was given. We now provide a sufficient condition that is easier to check:

PROPOSITION 4.42. *Let H_X be an hybrid control system, $\overline{R}$ an admissible relation and H_Y a $\overline{R}$-abstraction of H_X. If H_X is non-blocking and*

- *For all $p \in P$, N_p satisfies $dim(N_p) > 0$.*
- *Proposition 4.14 holds for the finite automaton underlying H_X*

then H_Y is non-blocking.

PROOF. The first condition ensures that for any $y \in \pi_Y^p(dom(F_Y^p))$, $A_y \neq \{\varepsilon\}$ by definition of hybrid control system and the continuous abstracting methodology [60, 64]. This means that blocking can only occur by removing discrete transitions. However the second assumption implies that blocking is not created on the abstracting process by removing discrete transitions. □

This result reveals that while we have continuous dynamics we only need to check what happens to the finite automaton underlying the hybrid control system to infer non-blocking. This is in principle a simple task since the number of discrete states is finite and Proposition 4.42 can be checked algorithmically

One could also attempt to determine when non-blocking is reflected by $\overline{R}$. However checking the conditions to determine if the reflection holds would be as expensive as determining if the original system is non-blocking.

4.4.2. *Zeno.* Next we examine a phenomena that has no counterpart in the discrete neither in the continuous world, the Zeno phenomena. Intuitively we say that a trajectory of an hybrid system is Zeno if there is an infinite number of jumps in finite time. This is in fact a modeling problem since no physical system is able of generating such a trajectory. On a more mathematical level existence of Zeno trajectories is equally a problem. First, one needs to deal with cardinals greater than the cardinal of the natural numbers if one attempts to define or even to refer to the states visited by the trajectory after the occurrence of infinitely many jumps in finite time. Second, Zeno trajectories make impossible to prove results using finite induction. We will have to slightly extend our setting to be able to talk about Zeno since the elements of $\mathcal{M}$ are finite length strings, therefore not capturing an infinite number of jumps. We thus need to move from finite monoids to ω-monoids. We will just briefly explain how one can extend

$$\mathcal{M} = \coprod_{t \in \mathbb{N}_0} (U^* \cup \Sigma^*)^t \tag{4.67}$$

to accommodate infinite strings without entering the technical definitions. The interested reader is deferred to [65] for more details regarding automata, infinite strings and semigroups. First we add to $\mathcal{M}$ the set of infinite strings of elements in $U^* \cup \Sigma^*$ defined as:

$$\mathcal{M}_\omega = (U^* \cup \Sigma^*)^{\mathbb{N}} \tag{4.68}$$

to get $\mathcal{M}_\infty = \mathcal{M} \cup \mathcal{M}_\omega$. Then we extend the product operation (concatenation in this case) to the following situations:

$$(a, b) \mapsto ab \text{ for } (a, b) \in \mathcal{M} \times \mathcal{M}_\omega \text{ and } ab \in \mathcal{M}_\omega \tag{4.69}$$

$$(a_1, a_2, \ldots, a_n, \ldots)_{n\in\mathbb{N}} \mapsto (a_1 a_2 \ldots a_n \ldots)_{n\in\mathbb{N}} \text{ for } a_n \in \mathcal{M} \text{ and } (a_1 a_2 \ldots a_n \ldots)_{n\in\mathbb{N}} \in \mathcal{M}_\omega \tag{4.70}$$

In this setting we can talk about Zeno sequences. Let us denote by $|m|_t$ the time duration of an element of $\mathcal{M}_\infty$. This duration is defined as the sum of the durations of all the elements of U^* that were concatenated to obtain m. A Zeno sequence is therefore defined as follows:

DEFINITION 4.43 (Zeno Sequence). Let $m \in \mathcal{M}_\omega$ be an input sequence. We say that m is a Zeno sequence iff we have $|m|_t < \infty$.

A Zeno hybrid control system is an hybrid control system such that its action is defined for Zeno input sequences:

DEFINITION 4.44 (Zeno Hybrid Control Systems). Let H_X be an hybrid control system. H_X is a Zeno hybrid control system iff Φ_X is defined for Zeno input sequences.

First we will show how one can ensure that non-Zeno trajectories are abstracted to non-Zeno trajectories. This will ensure that these non-physically meaningful sequences are not created by the abstraction process.

PROPOSITION 4.45 (Preservation of Non-Zeno). *Let H_X be an hybrid control system over X, $\overline{R}$ an admissible relation and H_Y a $\overline{R}$-abstraction of H_X. If there is only one covering set for each set $\pi_X^q(dom(F_X^q))$ or if the covering Γ_q is finitely compatible with F_X^q for every $q \in Q$ then non-Zeno input sequences are abstracted to non-Zeno input sequences.*

PROOF. Let $a_{(q,x)}$ be an input sequence of H_X and $a_{(p,y)}$ the corresponding abstracted input sequence. If $a_{(q,x)}$ is non-Zeno then the abstracted input sequence ξ_Y will be Zeno only if additional jumps are introduced by the abstracting process, that is, only if the continuous state space is abstracted into discrete components. We have therefore that if each set $\pi_X^q(dom(F_X^q))$ is covered by a single set no jumps are created and the input sequence remains non-Zeno. When there are several covering sets, the jumps created by crossing these sets will not induce Zeno sequences since the covering and the flow of F_X^q define a Zeno-free transition system. In details, we have that the number of elements from Σ_Y in $a_{(p,y)}$ is given by the sum of number of elements of Σ_X in $a_{(q,x)}$ not abstracted to ε plus the number of jumps induce by the crossing of adjacent covering set by the trajectories of F_X^q. Since $a_{(q,x)}$ is non-Zeno and the trajectories of F_X^q cross the boundaries of adjacent covering sets a finite number of times in finite time we have that the total number of elements of Σ_Y in $a_{(p,y)}$ is finite for finite time. This implies that every input sequence of H_Y is non-Zeno by surjectivity of $\overline{R}$. □

Note that a sufficient condition to ensure that the partition defines a Zeno-free transition system is given by the use of sub-analytic stratifications as described in [41].

The previous result formally shows that Zeno phenomena is introduced in hybrid models of physical systems by incorrect abstractions. When one models by discrete jumps, continuous evolutions that occur in a time scale much faster then the remaining process one may introduce non-physically meaningful trajectories such as Zeno sequences. This calls for the need to understand approximate abstractions where the abstracting systems need only to simulate the original systems approximately.

Having shown that it is not difficult to guarantee that non-Zeno trajectories propagate up in the hierarchy we came to a more interesting question. When can we ensure that a non-Zeno trajectory has non-Zeno refinements? We will only partially answer this question by determining when every refinement of a non-Zeno trajectory is non-Zeno. This amounts to ensuring that Zeno trajectories are abstracted to Zeno trajectories so that the trajectories are always divided into disjoint classes and the abstraction does not mix these classes.

PROPOSITION 4.46 (Preservation of Zeno). *Let H_X be an hybrid control system over X, $\overline{R}$ an admissible relation and H_Y a $\overline{R}$-abstraction of H_X. Every refinement in H_X of a non-Zeno input sequence of H_Y is non-Zeno if $\overline{R}$ preserves non-Zeno and for any state $(q,x)\in X$ and any discrete input $\sigma\in A_{(q,x)}$ such that $\overline{R}_B(q,x)\cap\overline{R}_B(\Phi_X((q,x),\sigma))\neq\varnothing$ we have $\pi_X(\overline{R}(q,x,a))=\{(p,y)\}$ and $\overline{R}(q,x,a)\neq\{(p,y,\varepsilon)\}$.*

PROOF. We want to that non-Zeno sequences are abstracted to non-Zeno sequences and that Zeno sequences are abstracted to Zeno sequences. The first part is ensured if $\overline{R}$ propagates non-Zeno while the second part will now be proved. If $a_{(q,x)}$ is a Zeno input sequence of H_X and $a_{(p,y)}$ (the corresponding abstracted input sequence of H_Y) is non-Zeno, then an infinite number of jumps has been removed from $a_{(q,x)}$. This can only be accomplished if the discrete inputs associated with these jumps are abstracted to ε. However (4.8) implies that if $\sigma\in A_{(q,x)}$ is abstracted to ε then $\overline{R}_B(q,x)\cap\overline{R}_B(\Phi_X((q,x),\sigma))\neq\varnothing$ but by assumption all such events σ are not abstracted to ε. □

We have only provided a superficial treatment of the Zeno phenomena which is however enough to provide some guarantees in real applications. We believe that a full understanding of this kind of behavior can only be achieved through the mathematical formalization of the operation that takes a discrete and a continuous control system and combines them into an hybrid system. We are, in fact, convinced that Zeno phenomena will be the result of that operation on singular (in some sense) cases.

4.5. Compositional Hybrid Abstractions. The results presented for compositionality of abstract control systems in Subsection 3.7 also carry over to hybrid control systems. In this subsection we present two examples of how modularity can be exploited to simplify abstraction tasks.

EXAMPLE 4.47. Consider a rubber ball bouncing on the floor under the action of gravity. Its dynamics can be described by the automaton displayed on the left of Figure 9. The state of the ball is described by the variables x and y modeling the ball position and v_x and v_y the velocity. The ball hits the floor at $y=0$ triggering a jump which resets the velocity on y with the new value $-e\,v_y$, where $e\in]0,1[$ is a parameter modeling the elasticity of the ball. To model two balls synchronized on the x position we start

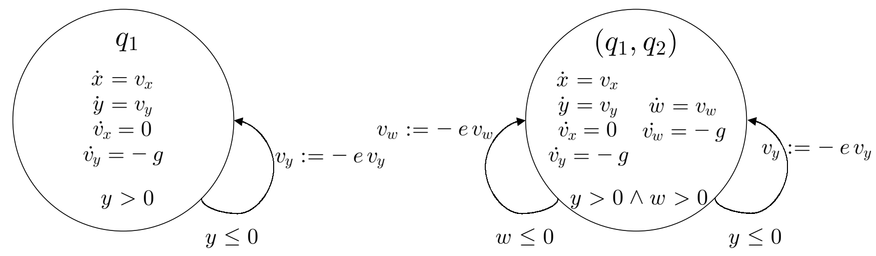

FIGURE 9. Hybrid automaton modeling a bouncing ball on the left and the composition with synchronization of two automata modeling a bouncing ball.

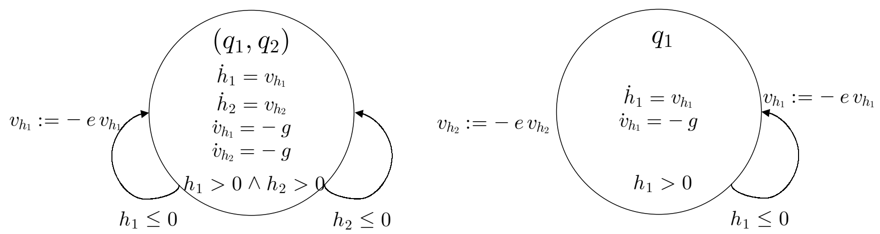

FIGURE 10. Left: abstraction of the hybrid automaton displayed on the right of Figure 9; Right: abstraction of the hybrid automaton on the left of Figure 9.

by computing the product automaton which is restricted to the set $L = \{((x, y, v_x, v_y), (z, w, v_z, v_w)) \in \mathbb{R}^4 \times \mathbb{R}^4 \;:\; x = z \wedge v_x = v_z\}$ resulting in the automaton displayed on the right of Figure 9. An abstraction can now be performed to retain only height information. The new state coordinates are naturally given by $h_1 = y, h_2 = w, v_{h_1} = v_y$ and $v_{h_2} = v_w$ and the abstraction computed by Algorithm 4.36 is displayed on the left of Figure 10. However, the abstraction process can be simplified by making use of Theorem 4.24. This is achieved by first abstracting the hybrid automaton modeling each individual ball which results in the hybrid automaton displayed on the right of Figure 10. The next step is to perform the parallel composition with synchronization of these hybrid automata. Note that this product is already simpler to perform than the product of the unabstracted systems. Furthermore the synchronizing set given by $(\phi_1, \phi_2)(L)$ equals the state space of the product system since $\phi_1(x, y, v_x, v_y) = (h_1, v_{h_1})$, $\phi_1(z, w, v_z, v_w) = (h_2, v_{h_2})$ and $L = \{((x, y, v_x, v_y), (z, w, v_z, v_w)) \;:\; y = w \;\wedge\; v_y = v_w\}$. We then see that no synchronization step needs to be performed and the resulting hybrid automaton is simply the product of two copies of the automaton displayed on the right of Figure 10. As expected the final hybrid automaton is the same as in the previous case, but the complexity of the process was considerably reduced by taking advantage to compositionality.

EXAMPLE 4.48. In this example we illustrate the use of Theorem 4.24 with the celebrated water tank system from [2]. Consider two water tanks that can be filled by water coming from a pipe as displayed on the left of Figure 11. The water level at tank A is measured by x_1 while the water level at tank B is

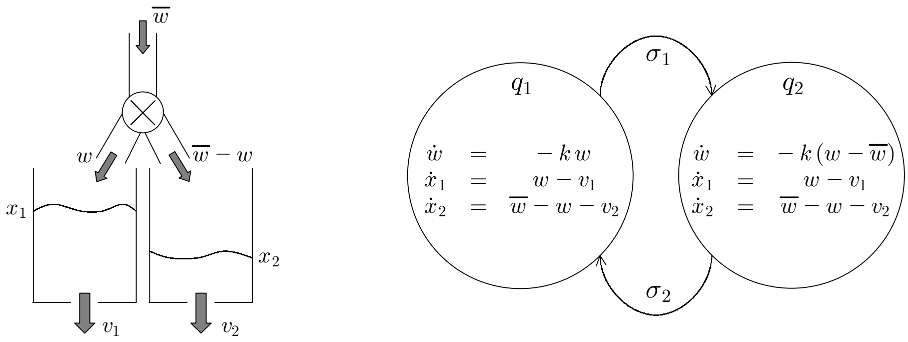

FIGURE 11. Water tank system: Physical setup on the left and hybrid model on the right.

measured by x_2. Each tank has also an outflow that causes a decrease in the water level. The outflow rate at tank A is v_1 while at tank B is v_2. This outflow can be compensated by a water inflow coming from the pipe on top of the tanks. This pipe has an inflow rate of $\overline{w}$ which can be directed to tank A or to tank B by means of a valve located in the pipe. Contrary to [2] we explicitly incorporate a first order model of the pump in the hybrid automaton describing this hybrid control system, displayed on the right of Figure 11. We now seek to abstract away the pump dynamics to obtain the usual model that considers the commutation of the inflow from one tank to the other instantaneous[4]. Instead of computing an abstraction directly from this hybrid automaton we start by realizing that this automaton can be obtained by parallel composition of hybrid control systems H_X and H_Y modeling the pipe and the tanks, respectively, as shown in Figure 12. This composition is synchronized on the fibering submonoid $A_L \subseteq A_X \times A_Y$ defined by the

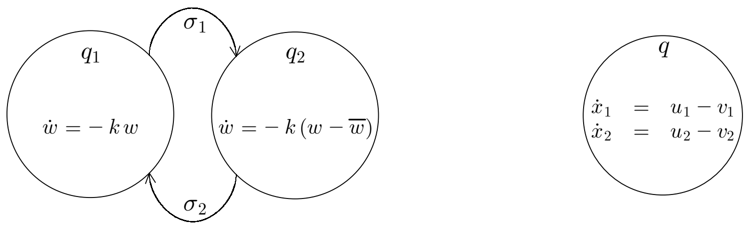

FIGURE 12. Hybrid model of the pipe and water tanks on the left and right, respectively.

points of the form $\big(((q_1,w),(x_1,x_2)),(\varepsilon,u^t)\big)$, $\big(((q_1,w),(x_1,x_2)),(\sigma_1,\varepsilon)\big)$, $\big(((q_2,w),(x_1,x_2)),(\varepsilon,u^t)\big)$ and $\big(((q_2,w),(x_1,x_2)),(\sigma_2,\varepsilon)\big)$, where the continuous inputs satisfy $u^t = (w(t), \overline{w} - w(t))$. We now abstract the

[4]We remark that considering the water commutation instantaneous leads to Zeno trajectories [35]. However, in our perspective, the hybrid model of the water tank system already allows infinite switches between discrete states q_1 and q_2 in finite time.

pipe model by aggregating all the continuous states in discrete state q_1 to 0 and all the continuous states in discrete state q_2 to $\overline{w}$. Theorem 4.24 ensures that composing H_Y with this abstraction will result in an abstraction of hybrid control system $H_X \|_{A_L} H_Y$. The new synchronizing fibering monoid is obtained from A_L by replacing w by 0 on the continuous inputs in state q_1, replacing w by $\overline{w}$ in the continuous inputs at discrete state q_2 and identifying (q_1, w) and (q_2, w) with q_1 and q_2, respectively. The resulting hybrid control system is displayed in Figure 13. This example illustrates the clear advantage of exploring

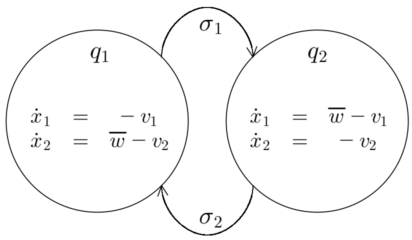

FIGURE 13. Abstracted hybrid model of the water tank system.

compositionality in computing hybrid abstractions. We have only computed continuous abstractions of one-dimensional control systems (for the pipe automaton), whereas if one would have proceeded directly from hybrid control system $H_X \|_{A_L} H_Y$ without exploring the compositional structure, one would have computed continuous abstractions of the three-dimensional continuous control systems at each discrete location.

CHAPTER 5

Formations and Abstractions of Multi-Agent Systems

1. Introduction

Advances in communication and computation have enabled the distributed control of multi-agent systems. This philosophy has resulted in next generation automated highway systems [86], coordination of aircraft in future air traffic management systems [82], as well as formation flying aircraft, satellites, and multiple mobile robots [7, 10, 80, 19]. The control of multi-agent systems is greatly simplified when the agent's mission can be executed by means of a *formation*. In several applications, maintaining a formation is even fundamental as in multiple aircraft where the formation is used to explore aerodynamic effects [51, 11] or in robotic exploration of large areas with restricted sensor capabilities [17].

The several approaches to *formation control* of a group of agents can roughly be divided into three categories: Behavior-based, Leader-Follower and Virtual Structures or Rigid-Body type formations. Behavior based approaches [7, 42, 90, 47] start by designing simple and intuitive behaviors or motion primitives for each individual agent. Then, by a weighted sum of these simple primitives more complex motion patterns are generated through the interaction of several agents. These motion patterns are usually called the group behavior that is said to emerge from the individual ones. Although this approach is characterized by being difficult to analyze in a rigorous and formal way, there have been some attempts to formally define and model behavior-based control schemes [20]. In leader-follower approaches [87, 19] one agent is designated the leader and is responsible for guiding the formation. The remaining agents are required to follow the leader with a predefined offset. This approach contrasts with rigid-body type formations [80] where rigidity allows to specify a trajectory for each agent requiring a centralized control architecture. See also [75] for a different centralized approach.

Despite the large activity in the area of formation control there are still fundamental questions unanswered. The control of a formation requires individual agents to satisfy their kinematics while constantly satisfying inter-agent constraints. In typical leader-follower formations, the leader has the responsibility of guiding the group, while the followers have the responsibility of maintaining the inter-agent formation. Distributing the group control tasks to individual agents must be compatible with the control and sensing capabilities of the individual agents. As the inter-agent dependencies get more complicated, a systematic framework for controlling formations is vital.

In this chapter, we propose a framework to study formation feasibility of multi-agent systems. Formations are modeled using *formation graphs* which are graphs whose nodes capture the individual agent kinematics, and whose edges represent inter-agent constraints that must be satisfied. A similar approach has been proposed in [19]. In [21] graphs are also used in the context of formation control, but the emphasis in on the communication flow and not on formation constraints. We assume kinematic models for each agent described by drift free control systems. This class of systems is rich enough to capture holonomic, nonholonomic, or underactuated agents. Two distinct types of formations are considered : *undirected formations* and *directed formations.*

In undirected formations each agent is equally responsible for maintaining the formation. For each edge constraining two agents of the formation graph, both agents cooperate in order to satisfy the constraint. Undirected formations therefore present a more centralized approach to the formation control problem as communication between agents is, in general, necessary. In directed formations, for each edge constraining two agents, only one of the agents (the follower) is responsible for maintaining the constraint. Directed formations, therefore, represent a more decentralized solution to the formation control problem.

In this chapter, we focus on the *feasibility* problem: *Given the kinematics of several agents along with the inter-agent constraints, determine whether there exist agent trajectories that maintain the constraints.* For both directed and undirected formations we obtain differential-geometric conditions that determine formation feasibility. When such conditions are verified, the *group abstraction* problem is then considered: *Given a feasible formation, extract a smaller control system that maintains formations along its trajectories.* The extracted control system allows to control the formation as a single entity, therefore being well suited for higher levels of control. In the case of undirected formations, the centralized nature of the problem allows us to determine feasibility using a single mathematical object. An unified approach that captures both the agent kinematics as well as the formation constraints is offered by differential forms and exterior differential systems [61]. In both the undirected and the directed cases the proposed framework allows for the extraction of a formation control abstraction. Since the abstraction can also be represented by differential forms, non-holonomic motion generation techniques based on exterior differential systems [81, 46] can readily be used to plan paths for the abstraction. The construction of these abstractions can be seen as a purely continuous example of the notion of parallel composition with synchronization introduced in Chapter 4. A preliminary version of the results presented in this chapter appeared in [79].

2. Formation Graphs

Consider n heterogeneous agents with states $x_i(t) \in M_i$, $i = 1, \ldots, n$ whose kinematics are defined by drift free controlled distributions on manifolds M_i as:

$$\begin{aligned} \Delta_i &: M_i \times U_i \to TM_i \\ \Delta_i &= \sum_j X_j u_j \end{aligned} \tag{5.1}$$

where U_i is the control space, and the vector fields X_j form a basis for the distribution. The controlled distributions are general enough to model nonholonomy and underactuation.

The formation of a set of agents is defined by the *formation graph* which completely describes individual agent kinematics and global inter-agent constrains.

Definition 5.1 (Formation Graph). A formation graph $F = (V, E, C)$ consists of:

- A finite set V of vertices whose cardinality is equal to the number of agents. Each vertex $v_i : M_i \times U_i \to TM_i$ is a distribution Δ_i modeling the kinematics of each individual agent as described in (5.1).
- A binary relation $E \subset V \times V$ representing a link between agents.
- A family of constraints C indexed by the set E, $C = \{c_e\}_{e \in E}$. For each edge $e = (v_i, v_j)$, c_e is a possibly time varying function $c_e(x_i, x_j, t) = 0$ describing the $\phi(e)$ independent constraints between vertices v_i and v_j. For a generic edge $e = (v_i, v_j)$, c_e is mathematically defined as $c_e : M_i \times M_j \times \mathbb{R} \to \mathbb{R}^{\phi(e)}$, $\phi(e) \in \mathbb{N}\ \ \forall_{e \in E}$.

Two different types of formation graphs will be considered: undirected formations where (V, E) will be an undirected graph and directed formations where (V, E) will be a directed graph. In undirected formations, for each edge $e = (v_i, v_j)$ both agents are equally responsible for maintaining the associated constraint c_e. Undirected formations are represented by the underlying undirected graph (V, E) as displayed in Figure 1 for a formation with two agents and an edge between them. In directed formations the constraint c_e

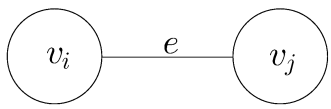

Figure 1. Undirected graph representing an undirected formation consisting of 2 agents and a constraint between them.

associated with the edge e must only be guaranteed by agent i. Directed formations are represented by the underlying directed graph as in Figure 2. At this point no further structure is assumed on the set E. Additional structure will be explicitly mentioned when needed.

We focus on the formation feasibility problem, more precisely:

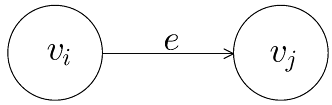

FIGURE 2. Directed graph representing a directed formation consisting of 2 agents and a constraint between them

PROBLEM 5.2. *Feasibility Given a formation graph $F = (V, E, C)$ determine whether there are solutions $x_i(t)$ of all agent kinematics (5.1) that maintain the constraints c_e for all $e \in E$ and for all $t \in \mathbb{R}$.*

We will solve Problem 5.2 for both undirected and directed formations. In case the formation is feasible, a new problem immediately emerges, the extraction of a formation control abstraction which characterizes the solution space of Problem 5.2 :

PROBLEM 5.3 (Group Abstraction). *Given a feasible formation graph $F = (V, E, C)$, extract a smaller control system that maintains formation for all values of its control inputs.*

Problem 5.3 will also be solved for both the undirected and the directed cases.

3. Undirected Formations

3.1. Feasibility. In undirected formations each agent is equally responsible for maintaining constraints. Because of this property it will be useful to collect all agent kinematics and constraints on a single manifold:

$$M = \prod_{i=1}^{n} M_i \tag{5.2}$$

. Given an element x of M the canonical projection on the ith agent,

$$\pi_i : M \longrightarrow M_i \tag{5.3}$$

allows us to denote the state of the individual agents by $x_i = \pi_i(x)$. The formation kinematics is obtained by appending individual kinematics through direct sum, that is:

$$\begin{aligned} \Delta &: M \times U \longrightarrow TM \\ \Delta &= \oplus_{i=1}^{n} \Delta_i \end{aligned} \tag{5.4}$$

where U is taken to be $U = \prod_{i=1}^{n} U_i$. This new control system Δ on M describes the kinematics of all formation agents, however it does not model any interaction between them. This interaction will be induced by the formation constraints that we now lift to the group manifold M. Each constraint c_e

linking agent i to agent j induces a constraint $\mathcal{C}_e$ on $M \times \mathbb{R}$ defined by:

$$\begin{aligned} \mathcal{C}_e &: M \times \mathbb{R} \longrightarrow \mathbb{R}^{\phi(e)} \\ \mathcal{C}_e(x,t) &= c_e(\pi_i(x), \pi_j(x), t) \end{aligned} \tag{5.5}$$

All of these constraints can now be grouped in a single map from $M \times \mathbb{R}$ to $\mathbb{R}^d$ with $d = \sum_{e \in E} \phi(e)$. This constraint map $\mathcal{C}$ is obtained by stacking all individual constraints as follows:

$$\mathcal{C} = \begin{bmatrix} \mathcal{C}_1 \\ \mathcal{C}_2 \\ \vdots \\ \mathcal{C}_m \end{bmatrix} \tag{5.6}$$

where we have considered an enumeration $\{1, 2, \ldots, m\}$ of the edges set E. Since the constraints are independent the set $\mathcal{C}^{-1}(\mathbf{0}) = \{(x,t) \in M \times \mathbb{R} \quad | \quad \mathcal{C}(x,t) = \mathbf{0}\}$ defines a submanifold[1] P of $M \times \mathbb{R}$. The projection of P on M (which is also a submanifold of M), denoted by N, characterizes the interaction between the agents since the state variables of each agent are restricted to live on this submanifold. Formation feasibility requires that the constraints are satisfied along the formation trajectories, that is, that the submanifold N is invariant under Δ trajectories:

$$\frac{d}{dt}\mathcal{C} = \mathcal{L}_X \mathcal{C} + \frac{\partial \mathcal{C}}{\partial t} = 0 \quad \forall X \in \Delta \tag{5.7}$$

Note that since $\mathcal{C}$ is vector valued we consider that the Lie derivate of $\mathcal{C}$ along X is given by:

$$\mathcal{L}_X \mathcal{C} = \begin{bmatrix} \mathcal{L}_X \mathcal{C}_1 \\ \mathcal{L}_X \mathcal{C}_2 \\ \vdots \\ \mathcal{L}_X \mathcal{C}_m \end{bmatrix} \tag{5.8}$$

To develop a single mathematical object that will allow us to check for feasibility we will adopt a differential forms approach instead of working directly with the vector fields. By defining the exterior derivative of $\mathcal{C}$ as:

$$\mathrm{d}\mathcal{C} = \begin{bmatrix} \mathrm{d}\mathcal{C}_1 \\ \mathrm{d}\mathcal{C}_2 \\ \vdots \\ \mathrm{d}\mathcal{C}_m \end{bmatrix} \tag{5.9}$$

[1]Although the map $\mathcal{C}$ depends on the chosen enumeration, the submanifold it defines does not.

equation (5.7) can be written as $\mathrm{d}\mathcal{C}|_t(X) = -\frac{\partial}{\partial t}\mathcal{C}$, where we have denoted by $\mathrm{d}\mathcal{C}|_t$ the exterior derivative of $\mathcal{C}$ for fixed t. If we now define the following vector valued forms:

$$\omega_F = \begin{bmatrix} \mathrm{d}\mathcal{C}_1|_t \\ \mathrm{d}\mathcal{C}_2|_t \\ \vdots \\ \mathrm{d}\mathcal{C}_m|_t \end{bmatrix} \qquad T_F = -\begin{bmatrix} \frac{\partial \mathcal{C}_1}{\partial t} \\ \frac{\partial \mathcal{C}_2}{\partial t} \\ \vdots \\ \frac{\partial \mathcal{C}_m}{\partial t} \end{bmatrix} \tag{5.10}$$

we can express equation (5.7) as:

$$\omega_F(X) = T_F \tag{5.11}$$

The kinematics can also be modeled as differential forms by using the annihilating codistributions. This lead us to define a single codistribution ω_K modeling the kinematics of all formation agents as:

$$\omega_K(X) = \begin{bmatrix} \omega_{K_1}(X_1) \\ \omega_{K_2}(X_2) \\ \vdots \\ \omega_{K_n}(X_n) \end{bmatrix} = \mathbf{0} \tag{5.12}$$

Solutions of equation (5.11) represent vector fields that maintain formation while solutions of equation (5.12) satisfy the kinematics. Therefore by merging both objects into:

$$\Omega = \begin{bmatrix} \omega_F \\ \omega_K \end{bmatrix} \qquad T = \begin{bmatrix} T_F \\ \mathbf{0} \end{bmatrix} \tag{5.13}$$

we can check for formation feasibility in a single equation:

$$\Omega_x(X) = T \quad \forall x \in N \tag{5.14}$$

Note that this equation only needs to hold for points belonging to N, since outside N the agents are no longer in formation. The previous discussion leads to the following solution of Problem 5.2:

PROPOSITION 5.4. *An undirected formation is feasible iff equation (5.14) has solutions, equivalently iff T belongs to the range of Ω for all $x \in N$.*

COROLLARY 5.5 (Time-Invariant Case). *If the formation constraints C are time-invariant then the undirected formation is feasible iff Ω_x is not of full rank at every $x \in N$*

A solution of equation $\Omega_x(X) = T$ specifies the infinitesimal motion of each individual agent. When more than one independent solution exists, a change in the direction of a single agent may require that all other agents also change their actions to maintain formation. This shows that, in general, solutions for undirected formations are centralized and require inter-agent communication for their implementation.

EXAMPLE 5.6. As an example of the methodology developed so far we consider an undirected formation consisting of three mobile robots of the unicycle type as displayed in Figure 3. The kinematics of each

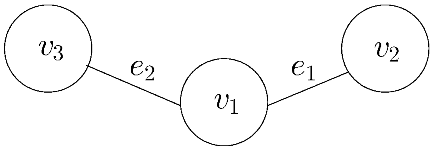

FIGURE 3. Undirected 3 agents formation.

agent is given by codistributions of the form (2.22). To completely specify the formation graph we need to define the constraints between the agents. Denoting by e_1 the edge between agent 1 and 2 we define the associated constraint as:

$$c_{e_1} = \begin{bmatrix} x_1 - x_2 - \delta_x \\ y_1 - y_2 - \delta_y \\ \theta_1 - \theta_2 \end{bmatrix} \tag{5.15}$$

where δ_x and δ_y are positive constants. The edge between agents 1 and 3 is denoted by e_2 and the associated constraint is given by:

$$c_{e_2} = \left[\tfrac{1}{2}(x_1 - x_3)^2 + \tfrac{1}{2}(y_1 - y_3)^2 - \tfrac{1}{2}(\theta_1 - \theta_3)^2 - \delta\right] \tag{5.16}$$

where δ is a positive constant. The constraint between agents 1 and 2 requires them to perform the same trajectories with an offset between their position coordinates given by δ_x and δ_y. It is intuitive that it is always possible to do so. However the constraint between agents 1 and 3 states that the distance between their positions should always equal $\delta + \frac{1}{2}(\theta_1 - \theta_3)^2$. This is clearly a non-intuitive constraint and no a priori answer can be given regarding feasibility. We will now study feasibility of this formation according to the methods developed so far. First we compute ω_K which is given by:

$$\omega_K = \begin{bmatrix} -\sin\theta_1 \mathrm{d}x_1 + \cos\theta_1 \mathrm{d}y_1 \\ -\sin\theta_2 \mathrm{d}x_2 + \cos\theta_2 \mathrm{d}y_2 \\ -\sin\theta_3 \mathrm{d}x_3 + \cos\theta_3 \mathrm{d}y_3 \end{bmatrix} \tag{5.17}$$

Since $\mathcal{C}$ is given by:

$$\mathcal{C} = \begin{bmatrix} x_1 - x_2 - \delta_x \\ y_1 - y_2 - \delta_y \\ \theta_1 - \theta_2 \\ \frac{1}{2}(x_1 - x_3)^2 + \frac{1}{2}(y_1 - y_3)^2 - \frac{1}{2}(\theta_1 - \theta_3)^2 - \delta \end{bmatrix} \tag{5.18}$$

the form ω_F will be given by:

(5.19)

$$\omega_F = \begin{bmatrix} \mathrm{d}x_1 - \mathrm{d}x_2 \\ \mathrm{d}y_1 - \mathrm{d}y_2 \\ \mathrm{d}\theta_1 - \mathrm{d}\theta_2 \\ (x_1 - x_3)\mathrm{d}x_1 + (y_1 - y_3)\mathrm{d}y_1 + (\theta_3 - \theta_1)\mathrm{d}\theta_1 + (x_3 - x_1)\mathrm{d}x_3 + (y_3 - y_1)\mathrm{d}y_3 + (\theta_1 - \theta_3)\mathrm{d}\theta_3 \end{bmatrix}$$

Combining ω_F and ω_K into Ω one easily verifies that Ω is not of full rank for every $x \in N$. This means that the formation is indeed feasible, that is, there are trajectories for each agent satisfying the formation constraints as well as its kinematics.

In the next section we will see how one can control the individual agents while maintaining the formation and gain some insight into the group trajectories.

3.2. Group Abstraction. Whenever more than one independent solution exists, the solution space of equation $\Omega(X) = T$ can be used to extract a smaller control system that will preserve the formation along its trajectories. This new control system is an abstraction that hides away low-level control necessary to maintain the formation and can be used in higher levels of control. Since the solution space is in general an affine space the new control system will also be affine in the control. If K_P is a particular solution of equation (5.14), we can solve Problem 5.3 with the new control system:

$$\Delta_G = K_P + Ker(\Omega) \tag{5.20}$$

By making use of a basis $\{K_1, K_2, \ldots, K_k\}$ for the kernel of Ω, we can rewrite (5.20) in a more usual form as:

$$\Delta_G = K_P + \sum_{j=1}^{k} K_j u_j \tag{5.21}$$

In the time-independent case we recover linearity of the abstracted control system since we can chose $K_P = 0$. The centralized nature of the problem is also reflected on the control abstraction. When one or more of the control inputs u_i are used, inter-agent cooperation is necessary to implement the new direction of motion since each vector K_j specifies the motion for all formation agents.

Example 5.7. Continuing with the previous example we will extract an abstraction representing the formation as a whole. Straightforward computations provide the following basis for the kernel of Ω:

$$
\begin{aligned}
K_1 &= (\theta_3-\theta_1)\cos\theta_1\frac{\partial}{\partial x_1}+(\theta_3-\theta_1)\sin\theta_1\frac{\partial}{\partial y_1}+(\theta_3-\theta_1)\cos\theta_1\frac{\partial}{\partial x_2}+(\theta_3-\theta_1)\cos\theta_1\frac{\partial}{\partial y_2}\\
&+ \Big((x_1-x_3)\cos\theta_1+(y_1-y_3)\sin\theta_1\Big)\frac{\partial}{\partial\theta_3}\\
K_2 &= \Big((x_1-x_3)\cos\theta_3+(y_1-y_3)\sin\theta_3\Big)\cos\theta_1\frac{\partial}{\partial x_1}+\Big((x_1-x_3)\cos\theta_3+(y_1-y_3)\sin\theta_3\Big)\sin\theta_1\frac{\partial}{\partial y_1}\\
&+ \Big((x_1-x_3)\cos\theta_3+(y_1-y_3)\sin\theta_3\Big)\cos\theta_1\frac{\partial}{\partial x_2}+\Big((x_1-x_3)\cos\theta_3+(y_1-y_3)\sin\theta_3\Big)\sin\theta_1\frac{\partial}{\partial y_2}\\
&+ \Big((x_1-x_3)\cos\theta_1+(y_1-y_3)\sin\theta_1\Big)\cos\theta_3\frac{\partial}{\partial x_3}+\Big((x_1-x_3)\cos\theta_1+(y_1-y_3)\sin\theta_1\Big)\sin\theta_3\frac{\partial}{\partial y_3}\\
K_3 &= (\theta_1-\theta_3)\cos\theta_1\frac{\partial}{\partial x_1}+(\theta_1-\theta_3)\sin\theta_1\frac{\partial}{\partial y_1}+\Big((x_1-x_3)\cos\theta_1+(y_1-y_3)\sin\theta_1\Big)\frac{\partial}{\partial\theta_1}\\
&+ (\theta_1-\theta_3)\cos\theta_1\frac{\partial}{\partial x_2}+(\theta_1-\theta_3)\sin\theta_1\frac{\partial}{\partial y_2}+\Big((x_1-x_3)\cos\theta_1+(y_1-y_3)\sin\theta_1\Big)\frac{\partial}{\partial\theta_2}
\end{aligned}
$$

These vector fields define the abstraction through the control system:

$$\Delta_G = K_1u_1 + K_2u_2 + K_3u_3 \tag{5.22}$$

To gain some insight on the abstraction control system and the formation trajectories we display in Figure 4 the formation evolution when the open loop control $u_1 = 1$, $u_2 = 0$ and $u_3 = 0$ is used. Agent

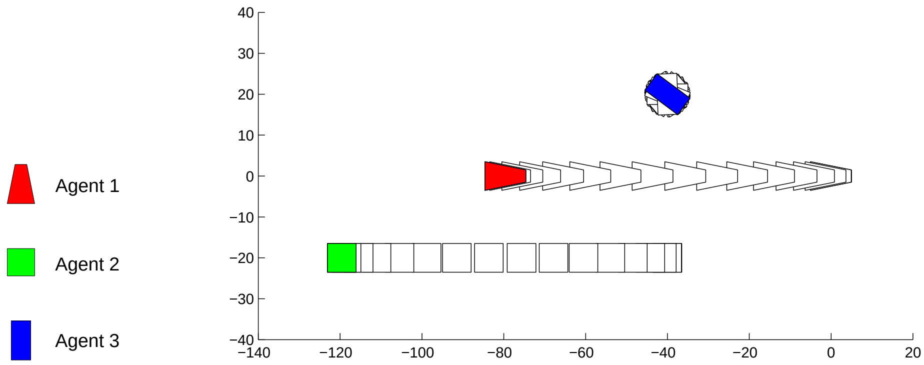

FIGURE 4. Formation flow along vector field K_1 corresponding to $u_1 = 1$, $u_2 = 0$ and $u_3 = 0$.

1 is represented by a trapezoid, agent 2 by a square and agent 3 by rectangle. The formation evolution is characterized by agent 3 rotating around the same point while agent 1 and 2 perform straight line motions. When the formation flows along vector field K_2 corresponding to the open loop control $u_1 = 0$, $u_2 = 1$ and $u_3 = 0$ all the agents in the formation move along parallel trajectories as displayed in Figure 5. This was achieved since their initial orientations where identical. When this is not the case, more complex motions characterize the flow along K_2. However it is always possible to achieve identical orientations by flowing along K_1 or K_3. The formation flow along basis vector K_3 is somewhat dual to K_1. Instead of agent 1 rotating around itself to achieve different configuration errors regarding agent 1, agent 3 is

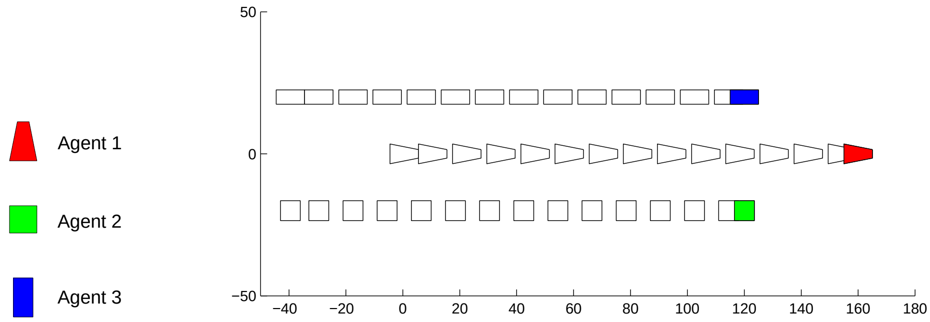

FIGURE 5. Formation flow along vector field K_2 corresponding to $u_1 = 0$, $u_2 = 1$ and $u_3 = 0$.

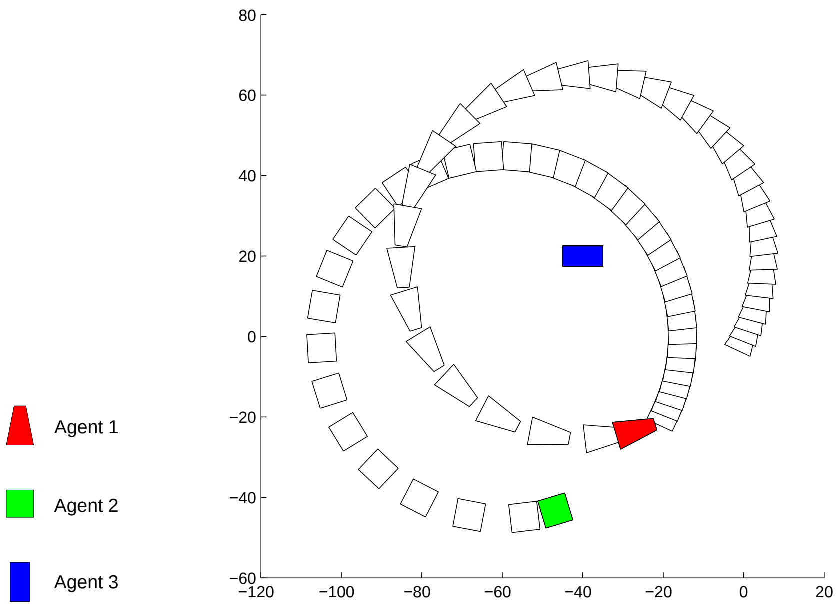

FIGURE 6. Formation flow along vector field K_3 corresponding to $u_1 = 0$, $u_2 = 0$ and $u_3 = 1$.

now stopped and the remaining agents revolve around it as suggested in Figure 6. To generate more complex motions for the formation other open or closed loops control laws can be used with the group abstraction (5.22).

3.3. Formation Guidance. In addition to using the above abstracted system to control the formation, one can also guide the formation by appending a *virtual* vertex v_0 defining the reference trajectory and several edges specifying how the reference should be followed by the formation. In particular consider a feasible formation graph $F = (V, E, C)$ and let V' be a singleton containing the vertex $v_0 : \mathbb{R} \to TM_0$, $v_0 = \frac{d}{dt}x_0(t)$. This vertex is connected to the remaining formation by the additional edge set $E' = \cup_{i \in I}\{(v_0, v_i)\}$, where $I \subseteq V$ is a subset of all the vertices indices. Associated with each vertex we have the constraints $C' = \{c'_e\}_{e' \in E'}$ and we can define a new formation graph given by

$F' = (V' \cup V, E' \cup E, C' \cup C)$. Once again it is necessary to ensure that the feasible formation is capable of maintaining the reference constraints by applying Proposition 5.4 to formation graph F'.

Note that this construction is general enough to encompass traditional formations such as: leader-follower by superimposing the virtual vertex onto an existing vertex or placing references on the formation centroid. It also allows some other interesting possibilities such as connecting a disconnected feasible formation graph by the reference constraints, *i.e.* , several independent formations following a single reference.

4. Directed Formations

Another important class of formations can be modeled by directed graphs. A directed graph assigns responsibilities to the formation members in an asymmetric way. For each edge $e = (v_i, v_j)$ agent i is responsible for maintaining the constraints c_e, while agent j is not affected by the constraint of the edge. Once agent j determines its motion, agent i is always capable of *locally* computing a control strategy enforcing the formation constraint. From an implementation point of view directed formations simplify the synthesis of the low level control laws responsible for maintaining the agents in formation. These control laws require only local information and are therefore easier to synthesize. The information flow is also simplified since each agent determines its motion without the need of coordination/cooperation with other agents.

We will assume through the remaining section that a directed formation graph is a directed acyclic graph. As a consequence all directed formations will have at least one leader. This assumption will allow recursive procedures to start on the leaders and to terminate since there are no cycles. Cyclic formation graphs, although important, will not be considered in this thesis. We will also consider that the formation constraints are time independent for simplicity of presentation although the results can easily be extended to time-varying constraints.

4.1. Feasibility.

Although in the undirected case we were able to lift the constraints and individual agents kinematics to a larger manifold M, we will adopt a different approach for the directed case. Given an edge $e = (v_i, v_j)$ the time derivative of its associated constraints c_e can be decomposed as:

$$\frac{dc_e}{dt} = \mathcal{L}_{X_i} c_e + \mathcal{L}_{X_j} c_e \tag{5.23}$$

Feasibility requires that $\frac{dc_e}{dt} = 0$, however only X_i can be chosen to ensure feasibility. In view of this we will follow a similar approach to the undirected case, but in a recursive formulation. This requires the following operators:

DEFINITION 5.8 (Post and Pre). Let $F = (V, E, C)$ be a directed formation graph. The *Post* operator is defined by:

$$\begin{aligned} Post : V &\longrightarrow 2^V \\ v_i &\mapsto \{v_j \in V \;:\; (v_i, v_j) \in E\} \end{aligned} \tag{5.24}$$

Similarly, the *Pre* operator is defined as:

$$\begin{aligned} Pre : V &\longrightarrow 2^V \\ v_i &\mapsto \{v_j \in V \;:\; (v_j, v_i) \in E\} \end{aligned} \tag{5.25}$$

Intuitively, $Post(v_i)$ will return the agents that are leading agent i, while $Pre(v_i)$ will return all the agents that are following agent i. *Post* and *Pre* extend to sets of vertices in the natural way, $Post(P) = \cup_{p \in P} Post(p)$ and $Pre(P) = \cup_{p \in P} Pre(p)$. A vertex v_i is called a *leader* iff $Post(v_i) = \varnothing$. By assumption the graph underlying the formation is acyclic implying that there will be at least a leader in the formation graph.

We shall abuse notation by representing the distribution Δ_i defining the kinematics of agent v_i as $\Delta(v_i)$ and for the set of agents $Post(v_i)$, $\Delta(Post(v_i)) = \oplus_{v \in Post(v_i)} \Delta(v)$ defined over $\Pi_{v \in Post(v_i)} M_v$. Similarly to the undirected case we define the following objects for each agent i:

$$\omega_F^i = \begin{bmatrix} \mathrm{d}c_1|_{x_j} \\ \mathrm{d}c_2|_{x_j} \\ \vdots \\ \mathrm{d}c_m|_{x_j} \end{bmatrix} \quad \omega_F^j = -\begin{bmatrix} \mathrm{d}c_1|_{x_i} \\ \mathrm{d}c_2|_{x_i} \\ \vdots \\ \mathrm{d}c_m|_{x_i} \end{bmatrix} \quad i \neq j \tag{5.26}$$

where $\{1, 2, \ldots m\}$ is an enumeration of the edges set between agent i and its leaders ($Post(v_i)$). These vector valued differential forms allow us to write equation (5.23) as:

$$\omega_F^i(X_i) = \omega_F^j(X_J) \tag{5.27}$$

which is to be considered only for $X_i \in \Delta(v_i)$ and $X_J \in \Delta(Post(v_i))$. Instead of restricting the X_i's to $\Delta(v_i)$ we can incorporate the kinematic restrictions directly into equation (5.27) by defining:

$$\Omega^i = \begin{bmatrix} \omega_F^i \\ \omega_K^i \end{bmatrix} \qquad \Omega^j = \begin{bmatrix} \omega_F^j \\ \mathbf{0} \end{bmatrix} \tag{5.28}$$

where ω_K^i is the vector valued form annihilating agent i kinematic distribution $\Delta(v_i)$. The equality $\frac{d}{dt} c_e = 0$ can now be further modified to the following form:

$$\Omega^i(X_i) = \Omega^j(X_J) \qquad \forall X_J \in \Delta(Post(v_i)) \tag{5.29}$$

This motivates the following result analogous to the undirected case:

PROPOSITION 5.9. *A directed formation is feasible iff equation (5.29) has solutions for each agent i in the formation. Equivalently iff the range of $\Omega^j|_{\Delta(Post(v_i))}$ is contained in the range of Ω^i for each agent i.*

Since Proposition 5.9 must be true for each agent, an algorithm can be constructed to determine feasibility. Let $L \subset V$ be a set of leaders and denote by F the operator returning the feasible directions of an agent i and defined by $F(v_i) = (\Omega^i)^{-1}(\mathcal{R}(\Omega^j|_{\Delta(Post(v_i))}))$, where $(\Omega^i)^{-1}(S)$ denotes the set of preimages of each $s \in S$ under Ω^i.

Algorithm 1 (Directed Feasibility)

initialization: $V := L$
while $Pre(V) \neq \varnothing$ **do**
    $V := Pre(V)$
    for all $v_i \in V$ **do**
        $\Delta(v_i) := \mathbf{0}$
        if $\mathcal{R}(\Omega^j|_{\Delta(Post(v_i))}) \nsubseteq \mathcal{R}(\Omega^i)$
            return UNFEASIBLE
            STOP
        else
            $\Delta(v_i) := \Delta(v_i) + F(v_i)$
        end if
    end
end

All the computations in the algorithm can be performed using basis vector fields for the distributions, in particular the inclusion $\mathcal{R}(\Omega^j|_{\Delta(Post(v_i))}) \subseteq \mathcal{R}(\Omega^i)$ needs to be tested only for the basis vectors and the inverse $(\Omega^i)^{-1}$ can be computed using pseudo-inverse techniques. The acyclic nature of the graph ensure us that the algorithm will terminate so that the following result naturally follows:

THEOREM 5.10 (Directed Formation Feasibility). *Let $F = (V, E, C)$ be an acyclic, directed formation graph. Algorithm 1 terminates in a finite number of steps and returns:*

- *Unfeasible if the formation is not feasible.*
- *A distribution per agent specifying the available directions to maintain formation if the formation is feasible.*

EXAMPLE 5.11. An example of directed feasibility motivated by the transportation of an hazardous load by a group of robots, escorted by another group of robots can be given by a 6 agent formation as depicted in Figure 7. Agents 1,2 and 3 move as a rigid body to collective transport the load. The remaining

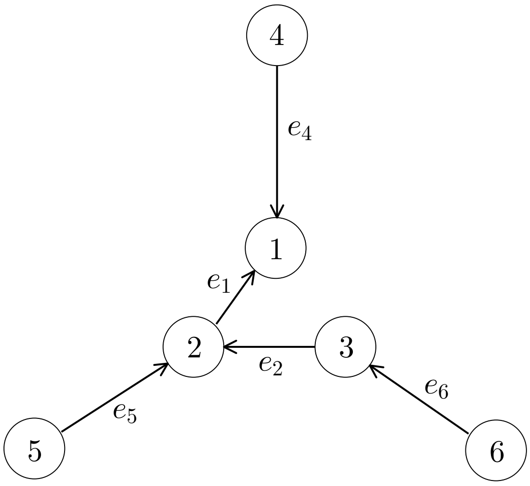

FIGURE 7. Directed graph representing a 6 agents formation.

agents serve as an escort to avoid attempts from external agents to approach the load, simultaneously protecting them from the possible hazards induced by the load. We will consider that agent 2 is of unicycle type being modeled by a distribution of the form (2.22) and all the remaining agents have no kinematic constraints, being therefore modeled by:

$$\Delta_i = X_1^i u_1^i + X_2^i u_2^i + X_3^i u_3^i \quad i = 1, 3, 4, 5, 6 \tag{5.30}$$

The constraints associated with edges e_4, e_5 and e_6 are simply given by:

$$c_{e_i} = (x_{i-3} - x_i)^2 + (y_{i-3} - y_i)^2 - \delta^2 \quad i = 4, 5, 6 \tag{5.31}$$

Intuitively the constraints model the fact that each agent belonging to the escort should keep a fixed distance of δ to a given robot transporting the load. The remaining constraints model in a directed way a rigid-body type formation with respect to the agents positions and are given by:

$$c_{e_1} = \begin{bmatrix} x_1 - x_2 - \delta_x \\ y_1 - y_2 - \delta_y \\ \theta_1 - \theta_2 \end{bmatrix} \quad c_{e_2} = \begin{bmatrix} x_2 - x_3 + \delta_x \\ y_2 - y_3 \\ \theta_2 - \theta_3 \end{bmatrix} \tag{5.32}$$

Following the steps of feasibility algorithm we start by analyzing the edge between agent 1 and 2. This requires the computation of:

$$\omega_F^1 = \begin{bmatrix} \mathrm{d}x_1 \\ \mathrm{d}y_1 \\ \mathrm{d}\theta_1 \end{bmatrix} \quad \omega_F^2 = \begin{bmatrix} \mathrm{d}x_2 \\ \mathrm{d}y_2 \\ \mathrm{d}\theta_2 \end{bmatrix} \tag{5.33}$$

and:

$$\Omega^1 = \begin{bmatrix} \mathrm{d}x_1 \\ \mathrm{d}y_1 \\ \mathrm{d}\theta_1 \\ 0 \end{bmatrix} \quad \Omega^2 = \begin{bmatrix} \mathrm{d}x_2 \\ \mathrm{d}y_2 \\ \mathrm{d}\theta_2 \\ \sin\theta_2 \mathrm{d}x_2 - \cos\theta_2 \mathrm{d}y_2 \end{bmatrix} \tag{5.34}$$

From these expressions we immediately see that $\mathcal{R}(\Omega^1) \nsubseteq \mathcal{R}(\Omega^2)$ since $[\sin\theta \;\; -\cos\theta \;\; 0 \;\; 0]^T$ belongs to $\mathcal{R}(\Omega^1)$ but it does not belong to $\mathcal{R}(\Omega^2)$. The formation is therefore not feasible. However if the edge e_1 is replaced by a new edge with the same associated constraint but with a reversed direction as displayed in Figure 8 feasibility is ensured. In this case we have that:

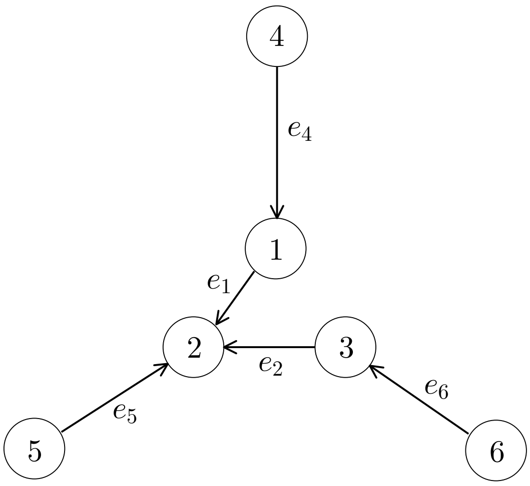

FIGURE 8. Directed graph representing a 6 agents formation with a new edge e_1.

$$\Omega^1 = \begin{bmatrix} \mathrm{d}x_1 \\ \mathrm{d}y_1 \\ \mathrm{d}\theta_1 \end{bmatrix} \quad \Omega^2 = \begin{bmatrix} \mathrm{d}x_2 \\ \mathrm{d}y_2 \\ \mathrm{d}\theta_2 \end{bmatrix} \tag{5.35}$$

and inclusion $\mathcal{R}(\Omega^1) \subseteq \mathcal{R}(\Omega^2)$ is true. The next vertex to analyze is v_3, but $\mathcal{R}(\Omega^2) \subseteq \mathcal{R}(\Omega^3)$ since the kinematics of agents 1 and 3 are equal as well as the exterior derivative of the constraints linking them to agent 2. To analyze edge e_4 one computes:

$$\begin{aligned} \Omega^1 &= [2(x_1 - x_4)\mathrm{d}x_1 + 2(y_1 - y_4)\mathrm{d}y_1] \\ \Omega^4 &= [2(x_1 - x_4)\mathrm{d}x_4 + 2(y_1 - y_4)\mathrm{d}y_4] \end{aligned} \tag{5.36}$$

and since agent 4 has no kinematic constraints the inclusion $\mathcal{R}(\Omega^1|_{\Delta(Post(v_4))}) \subseteq \mathcal{R}(\Omega^4)$ holds independently of $\Delta(Post(v_4))$. A similar reasoning shows that the corresponding inclusions also hold for agents 5 and 6. We conclude that the formations is feasible meaning that independently of agent 2 motion the remaining agents are always capable of locally determine a control strategy that will enforce the formation constraints.

4.2. Group Abstraction. When a directed formation is feasible the formation control abstraction is trivially taken as the control systems of the leaders. In the previous example the abstraction is simply given by the control system of agent 2. Contrary to the undirected case this abstraction does not allow direct control over each individual agent. Control is exerted on the leaders that indirectly control the formation through inter-agents links. Note that any attempt to control a non-leader agent in a abstraction would violate the semantics of a directed edge. On the other hand regarding the leaders as an abstraction of the formation is already implicit when the formation is specified by placing only ingoing arrows in these agents.

CHAPTER 6

Conclusions

Hybrid systems have been used to model multi-agent, networked and embedded systems among other kinds of complex large-scale systems. The increasing complexity of nowadays applications ask for analysis and synthesis methods that scale well with dimension and complexity. One approach is to adopt a hierarchical perspective by modeling hybrid systems through a hierarchy of different layers of abstraction representing different aspects of the same system. Analysis tasks are then performed on simpler, abstracted models that are equivalent with respect to the relevant properties. Synthesis tasks also benefit from this approach since the design starts as the top of the hierarchy on a simple model and is then successively refined by incorporating the modeling details of each layer of abstraction. A complementary approach to hierarchies of abstractions is to take advantage of the compositional structure of embedded systems. These systems are usually constructed through the interconnection of smaller components or subsystems. This should be regarded as structure that must be exploited to deal with the inherent complexity of these systems. One possible approach is to take advantage of this compositional structure of hybrid systems to simplify the computation of abstractions. This simplification comes from the fact that it is, usually, much simpler to abstract subsystems individually and then interconnect them to obtain an abstraction, than to extract the abstraction of the system as a whole. In order to accomplish this, compositional operators need to be compatible with abstraction operators.

In this thesis we introduced a general methodology for compositional abstractions of hybrid control systems. To accomplish this goal we also made several contributions to related areas such as abstractions of smooth control systems and formation control of multi-agent systems. In Chapter 3 we extended the continuous abstraction methodology proposed in [60, 63, 64] to model explicitly control inputs. We have characterized the structure of the abstracted control bundles in a hierarchy of abstractions induced by equivalence relations on the state space. These results were obtained by resorting to simple ideas from category theory that allowed to expose and understand the structure of smooth control systems. In Chapter 4 we proposed a methodology for compositional abstractions of hybrid control systems. An abstract framework capturing discrete, continuous and hybrid control systems was proposed as a category. In this category we introduced a notion of abstraction based on simulations and also the notion of bisimulation. We also introduced a composition operator modeling the interconnection and synchronization of subsystems. This operator was shown to be compatible with simulations and, under certain conditions on the synchronization, with bisimulations. All of these results were then specialized for hybrid control systems where an algorithm was proposed for the computations of abstractions. It was also shown that this algorithm also computes bisimulations under certain assumptions. All of these results constitute important tools to effectively deal with the complexity or large-scale, complex, embedded systems. Finally, in Chapter 5 we addressed and solved the formation feasibility problem for both directed and undirected formations. Furthermore we also provided a way of obtaining a group abstraction that maintains the formation along its trajectories. This abstraction can be regarded as a purely continuous example of the compositional abstraction methodology introduced in Chapter 4.

The research carried out under this Ph.D. program also lead to many interesting open questions that we mention only a few:

- In the case of purely continuous control systems it is not yet well understood when an abstraction is in fact a bisimulation. Determining checkable conditions for bisimilarity of smooth control systems is an extremely important problem not only from the applications perspective as well as from a theoretical point of view. Bisimilar control systems allow to design controllers hierarchically since we are assured that any specification for an abstract model has a feasible implementation or refinement in a more detailed level. Besides this perspective of a hierarchical control theory bisimilarity is also provides a major contribution to the classification of control systems. In this respect it would also be very rewarding to understand the relation between the symmetries of control systems and its bisimulations. It is clear that partial-symmetries as described in [57] lead to bisimilar quotient systems but is this always the case?
- With respect to hybrid control systems, it is fundamental to render the results developed in this work computational. In this respect it matters to identify special classes of hybrid control systems for which the proposed abstracting algorithm can be fully automated. Also some of the given results may be difficult to check in real examples, and again the identification of special classes of hybrid systems could be extremely helpful to overcome these difficulties. It is also important to stress that since large-scale, embedded systems are becoming increasingly distributed and networked an extension of the proposed methodology toward the explicit modeling of communication channels would be another valuable tool for the analysis and synthesis or real world applications.
- Finally, although we are able of determining if a given directed formation is feasible or not, it is important to consider the problem of determining if there are other inter-agents constraints defining a formation with the same trajectories as an unfeasible directed formation. A related problem is to extract the largest feasible directed formation from an unfeasible directed formation, since this would have direct impact in control and communication design.

Bibliography

[1] R. Abraham, J. Marsden, and T. Ratiu. *Manifolds, Tensor Analysis and Applications.* Applied Mathematical Sciences. Springer-Verlag, 1988.

[2] R. Alur and T.A. Henzinger. Modularity for timed and hybrid systems. In *Proceedings of the 9th International Conference on Concurency Theory*, volume 1243 of *Lecture Notes in Computer Science*, pages 74–88. Springer-Verlag, 1997.

[3] Rajeev Alur, Calin Belta, Franjo Ivancic, Vijay Kumar, Max Mintz, George J. Pappas, Harvey Rubin, and Jonathan Schug. Hybrid modeling and simulation of biomolecular networks. In Maria Domenica Di Benedetto and Alberto Sangiovanni-Vicentelli, editors, *Hybrid Systems: Computation and Control*, volume 2034 of *Lecture Notes in Computer Science*, pages 19–32. Springer Verlag, 2001.

[4] Rajeev Alur, Tom Henzinger, Gerardo Lafferriere, and George J. Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88(7):971–984, July 2000.

[5] M.A. Arbib and E.G. Manes. *Arrows, Structures and Functors - The Categorical Imperative.* Academic Press Inc, 1975.

[6] Michael Arbib and Ernest G. Manes. Machines in a category: An expository introduction. *SIAM Review*, 16(2):163–192, 1974.

[7] T. Balch and R. Arkin. Behavior-based formation control for multirobot systems. *IEEE Transactions on Robotics and Automation*, 14(12), 1998.

[8] A. Balluchi, L. Benvenuti, M. D. Di Benedetto, C. Pinello, and A. L. Sangiovanni-Vicentelli. Automotive engine control and hybrid systems: Challenges and opportunities. *Proceedings of the IEEE*, 88(7):888–912, July 2000.

[9] Andrea Balluchi, Luca Benvenuti, Guido M. Miconi, Ugo Pozzi, Tiziano Villa, Maria D. Di Benedetto, Howard Wong-Toi, and Alberto L. Sangiovanni-Vincentelli. Maximal safe set computation for idle speed control of an automotive engine. In Nancy Lynch and Bruce H. Krogh, editors, *Hybrid Systems: Computation and Control*, volume 1790 of *Lecture Notes in Computer Science*, pages 32–44. Springer Verlag, 2000.

[10] R.W. Beard, J.Lawton, and F.Y. Hadaegh. A coordination architecture for spacecraft formation control. *IEEE Transactions on Control Systems Technology*, 2000. To appear.

[11] Willian Blake and Dieter Multhopp. Design, performance, and modeling considerations for close formation flight. *American Institute of Aeronautics and Astronautics Journal*, 1998.

[12] W. M. Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry.* Academic Press, 1975.

[13] R.W. Brockett. Control theory and analytical mechanics. In C. Martin and R. Hermann, editors, *Geometric Control Theory*, volume VII of *Lie Groups: History, Frontiers and Applications*, pages 1–46. Math Sci Press, 1977.

[14] R.W. Brockett. Feedback invariants for nonlinear systems. In *Preprints of the 6th IFAC Congress*, pages 1115–1120, Helsinki, 1978.

[15] P.E. Caines and Y.J. Wei. Hierarchical hybrid control systems: A lattice theoretic formulation. *IEEE Transactions on Automatic Control : Special Issue on Hybrid Systems*, 43(4):501–508, April 1998.

[16] C. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems.* Kluwer Academic Publishers, Boston, MA, 1999.

[17] D. J. Cook, P. Gmytrasiewicz, and L. B. Holder. Decision-theoretic cooperative sensor planning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(10):1013–1023, 1996.

[18] J.E.R. Cury, B.H. Krogh, and T. Niinomi. Synthesis of supervisory controllers for hybrid systems based on approximating automata. *IEEE Transactions on Automatic Control : Special Issue on Hybrid Systems*, 43(4):564–568, April 1998.

[19] J. Desai, V. Kumar, and J. Ostrowski. Control of changes in formation of multi-robot teams. In *Proceedings of the 1999 International Conference on Robotics and Automation*, Detroit 1998.

[20] Magnus Egerstedt. Behavior based robotics using hybrid automata. In *Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science. Springer-Verlag, 2000.

[21] Fabrizio Giulietti, Lorenzo Pollini, and Mario Innocenti. Autonomous formation flight. *IEEE Control Systems Magazine*, 20(6):34–44, 2001.

[22] Kagan Gokbayrak and Christos G. Cassandras. Hybrid controller for hierarchically decomposed systems. In Nancy Lynch and Bruce H. Krogh, editors, *Hybrid Systems: Computation and Control*, volume 1790 of *Lecture Notes in Computer Science*, pages 117–129. Springer Verlag, 2000.

[23] Bill Goodwine and Joel Burdick. Controllability of kinematics control systems on stratified configuration spaces. *IEEE Transactions on Automatic Control*, 46(3):358–368, March 2001.

[24] K.A. Grasse. On controlled invariance for fully nonlinear control systems. *International Journal of Control*, 56:1121–1137, 1992.

[25] Jessy W. Grizzle and Steven I. Marcus. The structure of nonlinear control systems possessing symmetries. *IEEE Transactions on Automatic Control*, 30(3):248–258, March 1985.

[26] T.A. Henzinger. The theory of hybrid automata. In *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science*, pages 278–292. IEEE Computer Society Press, 1996.

[27] T.A. Henzinger. The theory of hybrid automata. In *Proceedings of the 11th Annual Symposium on Logic in Computer Science*, pages 278–292. IEEE Computer Society Press, 1996.

[28] Joao Hespanha, Stephan Bohacek, Katia Obraczka, and Junsoo Lee. Hybrid modeling of tcp congestion control. In Maria Domenica Di Benedetto and Alberto Sangiovanni-Vicentelli, editors, *Hybrid Systems: Computation and Control*, volume 2034 of *Lecture Notes in Computer Science*, pages 291–304. Springer Verlag, 2001.

[29] John E. Hopcroft and Jeffery D. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley Publishing Company, USA, 1979.

[30] John M. Howie. *Fundamentals of semigroup theory.* Oxford Science Publications, 1995.

[31] L.R. Hunt, R. Su, and G. Meyer. Design for multi-input nonlinear systems. In R.W. Brocket, R.S. Millman, and H.J.Sussmann, editors, *Differential Geometric Control Theory*, pages 268–298. Birkauser, Boston, 1983.

[32] A. Isidori, A.J. Krener, C. Gori Giorgi, and S. Monaco. Nonlinear decoupling via feedback: a differential geometrical approach. *IEEE Transactions on Automatic Control*, 26:331–345, 1981.

[33] Alberto Isidori. *Nonlinear Control Systems.* Springer-Verlag, third edition, 1996.

[34] B. Jakubczyk and W. Respondek. On linearization of control systems. *Bull. Acad. Pol. Sci. Ser. Sci. Math.*, 28:517–522, 1980.

[35] Karl Henrick Johansson, Magnus Egersted, John Lygeros, and S. Sastry. On the regularization of hybrid automata. *Systems and Control Letters*, 38:141–150, 1999.

[36] A. Joyal, M. Nielsen, and G. Winskel. Bisimulation from open maps. *LICS'93 special issue of Information and Computation*, 127(2):164–185, 1996.

[37] Scott Kelly and Richard Murray. Geometric phases and robotic locomotion. *Journal of Robotic Systems*, 12(6):417–431, 1995.

[38] Ivan Kolar, Peter W. Michor, and Jan Slovak. *Natural Operations in Differential Geometry.* Springer-Verlag, New York-Heidelberg-Berlin, 1993.

[39] A.J. Krener. On the equivalence of control systems and linearization of nonlinear systems. *SIAM Journal on Control and Optimization*, 11:670–676, 1973.

[40] Arthur J. Krener. A decomposition theory for differentiable systems. *SIAM Journal on Control and Optimization*, 15(5):813–829, August 1977.

[41] Gerardo Lafferriere, George J. Pappas, and Shankar Sastry. Subanalytic stratifications and bisimulations. In T. Henzinger and S. Sastry, editors, *Hybrid Systems : Computation and Control*, volume 1386 of *Lecture Notes in Computer Science*, pages 205–220. Springer Verlag, Berlin, 1998.

[42] D. Lager, J. Rosenblatt, and M. Hebert. A behavior-based systems for off-road navigation. *IEEE Transaction on Robotics and Automation*, 10(6):776–783, 1994.

[43] Saunders Mac Lane. *Categories for the Working Mathematician.* Springer-Verlag, 1971.

[44] F. William Lawvere and Stephen H. Schannel. *Conceptual Mathematics: A First Introduction to Categories.* Cambridge University Press, New York, 1997.

[45] Andrew D. Lewis. The category of affine connection control systems. In *Proceedings of the 39th IEEE conference on decision and control*, pages 1260–1265, Sydney, Australia, December 2000.

[46] Z. Li and J. Canny, editors. *Nonholonomic Motion Planning*. Kluwer Academic Publishers, 1993.

[47] Pedro Lima, Rodrigo Ventura, Pedro Aparício, and Luis Custódio. A functional architecture for a team of fully autonomous cooperative robots. In Manuela Veloso and Hiroaki Kitano, editors, *RoboCup-99: Robot Soccer World Cup III*, Lecture Notes in Computer Science. Springer-Verlag, 2000.

[48] Z. Manna and A. Pnueli. *Temporal Verification of Reactive Systems: Safety*. Springer Verlag, New York, 1995.

[49] Giuseppe Marmo, Alberto Simoni, Bruno Vitale, and Eugene J. Saletan. *Dynamical Systems*. John Wiley & Sons, 1985.

[50] L.S. Martin and P.E. Crouch. Controllability on principal fiber bundles with compact group structure. *Systems and Control Letters*, 5(1):35–40, 1984.

[51] Brian Maskew. Formation flying benefits based on vortex lattice calculations. Technical report, NASA-CR-151974.4, 1977.

[52] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[53] T. Moor and J. M. Davoren. Robust controller synthesis for hybrid systems using modal logic. In *Hybrid Systems : Computation and Control*, volume 2034 of *Lecture Notes in Computer Science*, pages 433–446. Springer Verlag, 2001.

[54] Peter Niebert and Sergio Yovine. Computing optimal operation schemes for chemical plants in multi-batch mode. In Nancy Lynch and Bruce H. Krogh, editors, *Hybrid Systems: Computation and Control*, volume 1790 of *Lecture Notes in Computer Science*, pages 338–351. Springer Verlag, 2000.

[55] H. Nijmeijer. Controlled invariance for affine control systems. *International Journal of Control*, 24:825–833, 1981.

[56] H. Nijmeijer. Feedback decomposition of nonlinear control systems. *IEEE Transactions on Automatic Control*, 28:861–862, 1983.

[57] H. Nijmeijer and A.J. van der Schaft. Partial symmetries for nonlinear systems. *Mathematical Systems Theory*, 18(1):79–96, 1985.

[58] H. Nijmeijer and A.J. van der Schaft. *Nonlinear Dynamical Control Systems*. Springer-Verlag, 1995.

[59] George J. Pappas. Bisimilar linear systems. 2001. Submitted.

[60] George J. Pappas, Gerardo Lafferriere, and Shankar Sastry. Hierarchically consistent control systems. *IEEE Transactions on Automatic Control*, 45(6):1144–1160, June 2000.

[61] George J. Pappas, John Lygeros, Dawn Tilbury, and Shankar Sastry. Exterior differential systems in control and robotics. In J. Baillieul, S. Sastry, and H. Sussmann, editors, *Essays on Mathematical Robotics*, volume 104 of *IMA Volumes in Mathematics and its Applications*, pages 271–372. Springer Verlag, New York, 1998.

[62] George J. Pappas and Shankar Sastry. Towards continuous abstractions of dynamical and control systems. In P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, editors, *Hybrid Systems IV*, volume 1273 of *Lecture Notes in Computer Science*, pages 329–341. Springer Verlag, Berlin, Germany, 1997.

[63] George J. Pappas and Slobodan Simic. Consistent hierarchies of nonlinear abstractions. In *Proceedings of the 39th IEEE Conference in Decision and Control*. Sydney, Australia, December 2000.

[64] George J. Pappas and Slobodan Simic. Consistent hierarchies of affine nonlinear systems. *IEEE Transactions on Automatic Control*, 2001. To appear.

[65] D. Perrin and J.-E. Pin. Semigroups and automata on infinite words. In J. Fountain, editor, *NATO Advanced Study Institute Semigroups, Formal Languages and Groups*, pages 49–72. Kluwer academic publishers, 1995.

[66] J. W. Polderman and J.C. Willems. *Introduction to Mathematical Systems Theory: A Behavioral Approach*. Springer-Verlag, New York, 1997.

[67] N. Halbwachs T.A. Henzinger P.H. Ho X. Nicollin A.Olivero J. Sifakis S. Yovine R. Alur, C.Courcoubetis. Hybrid automata: An algorithmic approach to specification and verification of hybrid systems. *Theoretical Computer Science*, 138:3–34, 1995.

[68] J. Raisch, E. Klein, S. O'Young, C. Meder, and A. Itigin. Approximating automata and discrete control for continuous systems - two examples from process control. In Panos Antsaklis, Wolf Kohn, Michael Lemmon, Anil Nerode, and Shankar Sastry, editors, *Hybrid Systems V*, volume 1567 of *Lecture Notes in Computer Science*, pages 279–303. Springer Verlag, 1999.

[69] J. Raisch and S.D. O'Young. Discrete approximations and supervisory control of continuous systems. *IEEE Transactions on Automatic Control : Special Issue on Hybrid Systems*, 43(4):569–573, April 1998.

[70] P.J. Ramadage and W.M. Wonham. Supervisory control of a class of discrete event systems. *SIAM Journal on Optimization and Control*, 25(1):206–230, 1987.

[71] P.J. Ramadage and W.M. Wonham. The control of discrete event systems. *Proceedings of IEEE*, 77(1):81–98, 1989.

[72] W. Respondek. On decomposition of nonlinear control systems. *Systems and Control Letters*, 1:301–308, 1982.

[73] Markus Roggenbach and Mila Majster-Cederbaum. Towards a unified view of bisimulation: a comparative study. *Theoretical Computer Science*, (238):81–130, 2000.

[74] J.J.M.M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, 249(1):3–80, 2000.

[75] Daniel J. Stilwell and Bradley E. Bishop. Platoons of underwater vehicles. *IEEE Control Systems Magazine*, 2001.

[76] Hector J. Sussmann. Orbits of families of vector fields and integrability of distributions. *Transactions of the American Mathematical Society*, 180:171–188, 1973.

[77] Paulo Tabuada and George J. Pappas. Hybrid abstractions that preserve timed languages. In *Hybrid Systems : Computation and Control*, volume 2034 of *Lecture Notes in Computer Science*. Springer Verlag, 2001.

[78] Paulo Tabuada and George J. Pappas. Quotients of fully nonlinear control systems. *SIAM Journal on Control and Optimization*, 2001. Submitted.

[79] Paulo Tabuada, George J. Pappas, and Pedro Lima. Feasible formations of multi-agent systems. In *Proceedings of the 2001 American Control Conference*, Arlington, VA, June 2001.

[80] Kar-Han Tan and M. Anthony Lewis. Virtual structures for high-precision cooperative mobile robot control. *Autonomous Robots*, 4(4):387–403, October 1997.

[81] D. Tilbury, R. M. Murray, and S. S. Sastry. Trajectory generation for the n-trailer problem using goursat normal form. *IEEE Transaction on Automatic Control*, 40(5):802–819, 1995.

[82] C. Tomlin, G. J. Pappas, and S. Sastry. Conflict resolution for air traffic management : A study in muti-agent hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):509–521, April 1998.

[83] A. van der Schaft. Symmetries and conservation laws for hamiltonian systems with inputs and outputs: A generalization of noether's theorem. *Systems and Control Letters*, 1:773–779, 1981.

[84] A. van der Schaft and B.Maschke. Interconnected mechanical systems, part i: geometry of interconnection and implicit hamiltonian systems. In A. Astolfi, D.J.N. Limebeer, C. Melchiorri, A. Tornambe, and R.B. Vinter, editors, *Modelling and Control of Mechanical Systems*. Imperial College Press, 1997.

[85] Arjan van der Schaft and Hans Schumacher. *An Introduction to Hybrid Dynamical Systems*, volume 251 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, 2000.

[86] P. Varaiya. Smart cars on smart roads: problems of control. *IEEE Transactions on Automatic Control*, 38(2):195–207, 1993.

[87] P. K. C. Wang. Navigation strategies for multiple autonomous mobile robots moving in formation. *Journal of Robotic Systems*, 8(2):177–195, 1991.

[88] J.C. Willems. System theoretic models for the analysis of physical systems. *Ricerche di Automatica*, 10:71–106, 1979.

[89] Glynn Winskel and Mogens Nielsen. Models for concurrency. In Abramsky, Gabbay, and Maibaum, editors, *Handbook of Logic and Foundations of Theoretical Computer Science*, volume 4. Oxford University Press, London, 1994.

[90] Gokhan Alptekin nd Okay Albayrak Xiaoping Yun. Line and circle formations of distributed physical mobile robots. *Journal of Robotic Systems*, 14(2):63–76, 1997.