

Copyright © 1998, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

**HYBRID SYSTEMS: COMPUTATION
AND ABSTRACTION**

by

George James Pappas

Memorandum No. UCB/ERL M98/78

16 December 1998

**HYBRID SYSTEMS: COMPUTATION
AND ABSTRACTION**

by

George James Pappas


Memorandum No. UCB/ERL M98/78

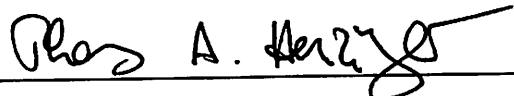
16 December 1998

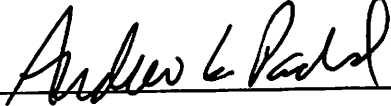
ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

The dissertation of George James Pappas is approved:

  Dec 11, 1998
Chair Date

  Dec. 16, 1998
Date

  Dec 10, 1998
Date

  10 Dec '98
Date

University of California at Berkeley

Fall 1998

HYBRID SYSTEMS : COMPUTATION AND ABSTRACTION

by

George James Pappas

B.S. (Rensselaer Polytechnic Institute) 1991

M.S. (Rensselaer Polytechnic Institute) 1992

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Sciences

in the

GRADUATE DIVISION

of the

UNIVERSITY of CALIFORNIA at BERKELEY

Committee in charge:

Professor Shankar Sastry, Chair

Professor Thomas Henzinger

Professor Andrew Packard

Professor Pravin Varaiya

Fall 1998

HYBRID SYSTEMS : COMPUTATION AND ABSTRACTION

Copyright Fall 1998

by

George James Pappas

Abstract

HYBRID SYSTEMS : COMPUTATION AND ABSTRACTION

by

George James Pappas

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California at Berkeley

Professor Shankar Sastry, Chair

Recent advances in computation and communication have enabled the control of large scale, multi-agent, distributed, and hierarchical systems such as automated highway and air traffic management systems. Improving the performance, safety, and reliability of such systems is extremely challenging as it requires new modeling formalisms accompanied by novel analysis and design techniques.

Hybrid systems combine discrete event and continuous time dynamics in a manner that can capture decision logic, agent dynamics, and inter-agent communication in a unified modeling framework. The expressive power of hybrid systems has been successfully applied in both automated highway and air traffic management systems. Given the safety criticality of such systems, one of the most important problems in the area of hybrid systems is the computation of the reachable space of a hybrid system in order to verify that no undesirable states are feasible.

The first part of this dissertation focuses on algorithmic methods for exactly computing the reachable states of hybrid systems. State of the art methods from theoretical computer science perform reachability computation for timed, multirate, and rectangular automata before reaching undecidability barriers. Using the very recent notion of o-minimality from mathematical logic, the first class of hybrid systems with linear differential equations having a decidable reachability problem is obtained. This result is important given the wide applicability of linear systems in control theory and applications.

The second part of this dissertation focuses on reducing the complexity of reachability calculations for continuous systems. In particular, a notion of abstraction for control

systems is introduced. In addition to complexity reduction, the notion of control system abstraction is useful in hierarchical system design. Conditions are derived under which one control system is a consistent abstraction of another, in the sense that checking reachability on the abstracted model is equivalent to the detailed model. For linear systems, this leads to a hierarchical controllability algorithm, whose computational advantages are verified by recovering the best known controllability algorithm from numerical linear algebra.



Professor Shankar Sastry
Dissertation Committee Chair

The author would like to thank the following people for their help and support during the preparation of this book: ...

To my family

Contents

List of Figures	vi
List of Tables	vii
1 Introduction	1
1.1 Research Areas and State of the Art	2
1.2 Issues Addressed and Dissertation Outline	9
2 Mathematical Background	13
2.1 Differential Geometry	13
2.1.1 Differentiable Manifolds	13
2.1.2 Tangent Spaces	15
2.1.3 Vector Fields	17
2.2 Subanalytic Geometry	19
2.2.1 Semianalytic and Subanalytic Sets	20
2.2.2 Subanalytic Stratifications	22
2.3 Mathematical Logic	25
2.3.1 Languages and Formulas	26
2.3.2 Model Theory	27
2.3.3 Decidability and Quantifier Elimination	28
3 Straightening Out Differential Inclusions	31
3.1 Straightening Out Differential Equations	33
3.2 Straightening Out Differential Inclusions	35
3.3 Conclusions	42
4 Computable Hybrid Systems	43
4.1 Bisimulations of Transition Systems	44
4.2 Bisimulations of Hybrid Systems	49
4.3 O-Minimal Theories	55
4.4 O-Minimal Hybrid Systems	59
4.5 Classes of O-Minimal Hybrid Systems	65
4.6 Linear Hybrid Systems	68
4.6.1 Nilpotent matrices	70

4.6.2	Diagonalizable matrices with rational eigenvalues	71
4.6.3	Diagonalizable matrices with imaginary eigenvalues	74
4.7	Conclusions	79
5	Abstractions of Control Systems	80
5.1	Abstractions of Vector Fields	83
5.2	Control System Abstractions	86
5.3	Consistent Control Abstractions	92
5.4	Consistent Linear Abstractions	97
5.5	Hierarchical Controllability Algorithm	110
5.6	Conclusions	116
6	Conclusions	117
	Bibliography	120
A	Appendix	132
A.1	Implemetation of Algorithms 5.37 and 5.38	132

List of Figures

1.1	Example of a timed automaton	3
1.2	Supervisory control of continuous systems	4
1.3	System analysis using abstractions	8
1.4	Two layer control hierarchy	9
2.1	Example of a partition but not a stratification	22
2.2	Example of subanalytic stratification	23
2.3	Infinite crossings on a bounded interval	24
3.1	Rectangular hybrid automaton	32
3.2	Straightening the flow of a vector field	33
4.1	A typical hybrid automaton	51
4.2	Bisimulation algorithm does not terminate	55
4.3	Inductive definition of cells	58
5.1	Two layer control hierarchy	81
5.2	Comparison of Algorithm 5.38 and the Kalman rank condition	115
5.3	Comparison of Algorithm 5.38 and the Popov-Belevitch-Hautus test	115
5.4	Comparison of Algorithm 5.38 and Algorithm 5.37 with $k = 1$	116

List of Tables

4.1	Definable sets and flows in o-minimal theories	57
-----	--	----

Acknowledgements

Being a graduate student at Berkeley was one of the most tremendous experiences in my life. The guiding light through this amazing journey has been my advisor, Professor Shankar Sastry, who has created a highly charged and scientifically stimulating atmosphere in his group while respecting the academic freedom of his students. His vision, advice, support, and continuous inspiration are comparable only to his wine tasting abilities and highly cultured taste of life. I will always be grateful for everything he has taught me about science and life.

I would like to thank Professors Tom Henzinger, Andy Packard, and Pravin Varaiya for serving on my Qualifying and Dissertation Committees. Tom's course on Computer Aided Verification has inspired much of the work in this dissertation, as he motivated me to transfer computer science concepts to control theory. A special note goes to Professor Gerardo Lafferriere with whom I had an excellent collaboration during his sabbatical stay at Berkeley.

I have greatly benefited from being surrounded by such a talented group of students and postdocs. Datta N. Godbole, John-Morten Godhavn, T. John Koo, Jana Košecká, John Lygeros, Yi Ma, Bruno Sinopoli, Claire Tomlin, and Sergio Yovine have not only collaborated with me, but have also made my every day life in the office extremely pleasant and enjoyable.

Many of you may know that I have spend an enormous amount of time for coffee breaks at Nefeli Cafe. This would not be possible without the help of Kostas Adam, Nick Bizouras, Jana Košecká, John Lygeros, Anna Papafragou, Jason Vassiliou, and the rest of my friends and local Greek Mafia. Our daily discussions on life, science, art, sports, stocks, and politics, will never be forgotten...

Last but not least, this dissertation is dedicated to my family. My parents have given me complete autonomy from a very young age, and have supported my every decision. To them, I owe more than I could possibly have...

Chapter 1

Introduction

In the past few decades, advances in computation and communication have enabled the development and control of large scale, highly complex systems. Air traffic management systems, automated highway systems, flight management systems, communication networks, and power distribution networks are a few examples that are important not only from an engineering perspective but also from their prevalence in our everyday lives and the well being of the economy.

The nature of the above systems is *distributed* as various subsystems or *agents* either compete or cooperate to satisfy individual or common objectives. As the computational ability of individual agents and the communication between agents increase rapidly, next generation systems attempt to balance centralized and decentralized designs by allowing individual agents to self optimize their own objectives but coordinate with other agents when conflicts arise. This naturally leads to multi-agent, multi-objective systems which are also multi-modal in the sense that the system functions in various modes of operation.

In addition, large scale systems such as air traffic management systems and automated highway systems, are systems of very high complexity. Complexity is typically reduced by imposing a *hierarchical* structure on the system architecture, where systems at higher levels utilize coarser system models than lower levels. Hierarchical structures also arise as a reflection of a hierarchy of system objectives.

In order to improve the performance, safety, and reliability of such systems, engineers are currently faced with the challenge of developing appropriate models, analysis, and design methods. The candidate modeling frameworks must have the expressive power to describe both agent dynamics, typically described by differential equations, as well as

decision logic and communication protocols, usually modeled by discrete event systems. In addition, they must be equipped with *composition* and *abstraction* operators in order to capture the distributed and hierarchical nature of such systems. Composition operators perform the proper interconnection and synchronization of subsystems whereas abstraction operators allow macromodeling, or the ability to hide unnecessary details at the higher level.

A solution to this modeling challenge is offered by *hybrid systems*. Hybrid systems are discrete event systems with possibly different differential equations in each discrete location. One modeling approach is to extend the formalisms and tools of finite state machines to incorporate continuous dynamics. This approach has been pursued by theoretical computer science, and has been applied in modeling real time hardware and software systems. In the control community, a variety of hybrid system models extend differential equations and control systems to capture discrete decision logic, and switching behavior. Naturally, computer science models focus hybrid systems with sophisticated discrete dynamics but simple continuous dynamics, whereas control theory models include complicated continuous dynamics but relatively rudimentary discrete behavior. It is becoming apparent from applications that a synergy of techniques from these fields is needed in order to analyze hybrid systems with complicated continuous dynamics and sophisticated discrete behavior.

1.1 Research Areas and State of the Art

The new modeling paradigm of hybrid systems has generated numerous issues that need to be resolved from both a theoretical and applied perspective. We list a few of them below while reviewing state of the art techniques in each area of hybrid systems research. More thorough and detailed starting points in the area of hybrid systems include [4, 6, 7, 41, 44, 74].

Modeling

Various hybrid system models have been proposed in the literature. Computer scientists consider hybrid automata which extend finite state machines to include simple continuous dynamics. This has resulted in timed automata [3], where clock dynamics of the form $\dot{x} = 1$ where inserted within each state. The automaton would instantaneously jump from one discrete location to another depending on the comparison of the clock value with integer constants. An example of a timed automaton is shown in Figure 1.1. Timed au-

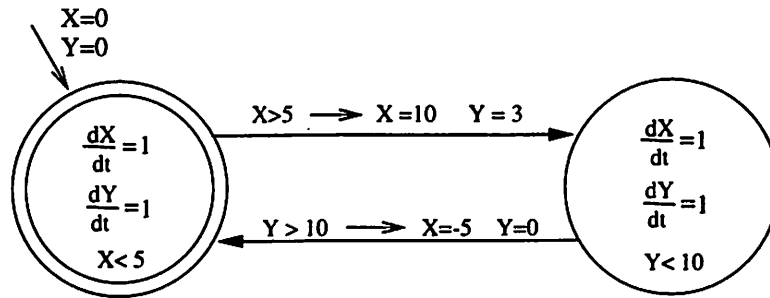


Figure 1.1: Example of a timed automaton

timed automata are useful for modeling and analyzing properties of real time hardware and software systems, such as real time communication protocols. Timed automata have been extended to fixed slope and multirate automata [2] which capture dynamics of the form $\dot{x} = c$, and rectangular automata [48, 91], where continuous dynamics are modeled by differential inclusions of the form $\dot{x} \in [a, b]$. Finally, the most expressive but still computable class of hybrid systems is linear hybrid automata [46], which capture dynamics of the form $A\dot{x} \leq b$. Hybrid input/output automata extend the input/output framework of [72], which has been used to analyze distributed algorithms and protocols. All the above models are equipped with composition operators which allow the parallel composition of various subsystems. This is an important modeling feature for the so called *concurrent* systems where different system components interact and synchronize with their environment across well defined boundaries.

On the other side, the control community starts with complicated differential equations and control systems, and starts adding discrete behavior [13, 23, 99, 108, 112]. Naturally, switched systems [80], and systems with discontinuous dynamics [40] are hybrid systems with special structure. Another way in which hybrid systems arise in control theory is the framework of supervisory control of continuous systems, shown in Figure 1.2. In this framework, a purely continuous plant is interconnected with a discrete event controller by generalized analog to digital and digital to analog converters. The outputs of the plant are quantized by the analog to digital converter, which generates discrete events as inputs to a discrete event supervisor. The supervisor encodes some computer program or decision logic, and outputs its control decision to the generalized digital to analog converter which parses the discrete command to a continuous control law. This framework is a special type of hybrid system that has been considered in [8, 26, 32, 50, 81, 93]. More unified modeling

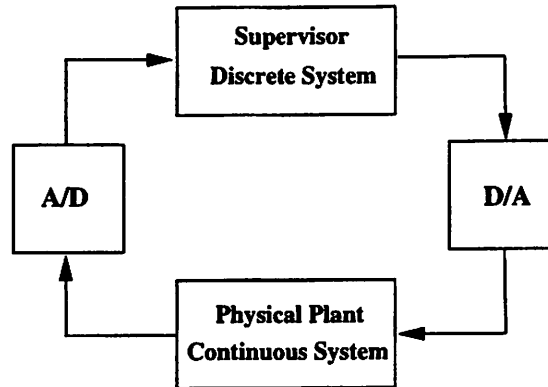


Figure 1.2: Supervisory control of continuous systems

approaches which capture general continuous and discrete dynamics include [18, 67]. The fundamental modeling components of these formalisms is truly hybrid, as state, inputs, and outputs can have both a discrete and continuous component. In addition, modeling formalisms accompanied by the first simulation tools for hybrid systems include SHIFT [37], OMOLA/OMSIM [5], and HYBRID CC [27].

With a few exceptions [99, 108], most of the above models lack qualitative results which guarantee their well posedness. As a result, there are very few results on existence, uniqueness, or robustness of system trajectories. Even though it is becoming apparent that hybrid systems, in general, are not robust, theorists are searching for the correct notion of continuity for hybrid systems [33]. Another manifestation of the complicated nature of hybrid system models is the *zeno* property. A hybrid system is zeno when there exist system trajectories with infinite switching in finite time. Zeno hybrid systems are not valid mathematical abstractions of physical processes. Much more work is needed in this area in order to understand the complex behavior that hybrid systems are capable of generating.

Analysis and Verification

Since hybrid models are used to describe *safety critical systems*, like air traffic management systems, it is important to have guarantees of safe operation. This is the goal of *formal verification methods* which, given a mathematical abstraction of the system, attempt to prove that the actual system satisfies the desired properties. Computational approaches to system verification is typically a three step methodology:

1. Modeling : The system is modeled in some hybrid system formalism
2. Specification : Desired properties are expressed as temporal logic formulas
3. Analysis : The system is analyzed using algorithmic or deductive methods

In the computer science community there are essentially two approaches to hybrid system verification. *Model checking approaches* essentially completely explore the whole state space to check whether the system satisfies the desired specification. The advantage of model checking approaches is that they can be completely automated, resulting in meaningful computer aided verification tools. In addition, the lack of structure of purely discrete systems makes computational approaches to model checking inevitable. Even though model checking methods are very successful in verifying properties of discrete systems with finite state spaces, their application to hybrid systems with infinite state spaces makes the issue of decidability extremely important.

The main tool for obtaining decidable classes of hybrid systems is given by the concept of *bisimulation* [79]. Bisimulations are simply quotient systems which preserve the properties of the original system. For purely discrete systems, bisimulations are used in order to reduce the complexity of verifying properties of very large scale systems. If an infinite state hybrid system has a finite state bisimulation, then checking properties for the hybrid system can be equivalently performed on the finite, discrete, quotient graph. Since the quotient graph is finite, the algorithm will terminate. If in addition, each step of the algorithm can be encoded and implemented by a computer program, then the problem is decidable.

The first decidability result of this kind for hybrid systems was obtained in [3] for timed automata, which are finite state machines with clock dynamics. This was extended to multirate automata [2], as well as initialized rectangular automata [48, 91] which at each discrete location contain constant rectangular differential inclusions of the form $\dot{x} \in [a, b]$. Based on these results, computational tools have been developed for automatic verification of timed (KRONOS [34] and UPPAAL [15]) and linear hybrid automata (HYTECH [47]). In [48], various relaxations of these models were shown to be undecidable.

The use of *deductive methods* is the second approach in the computer science for hybrid system verification. Deductive methods try to prove properties using formal deduction based on a set of inference rules [72, 76]. Even though deductive methods are

not constrained by any decidability frontiers, their application requires significant human intervention. This makes their application to large scale systems difficult. However, semi-automated tools like STEP [17], automate part of the verification procedure, thus reducing the human workload.

The response of the control community to hybrid system verification uses game theoretic methods [67]. Instead of verifying all system trajectories, a game is solved resulting in the worst possible system trajectory. If the worst trajectory satisfies the specification then so does every other system trajectory. The difficulty with this approach is in solving the game and obtaining the worst system trajectories. Other analysis results have focused on the stability of various classes of switched and hybrid systems. The first extension of Lyapunov type theorems to hybrid systems used multiple Lyapunov functions [18, 19]. Other Lyapunov type results include [117] as well as more constructive stability results for switched linear systems [52]. A hierarchical stabilization method for systems with changing dynamics can be found in [118].

Controller Synthesis

Whereas verification ensures properties of completed designs, controller synthesis attempts to design systems so that they are guaranteed to satisfy the desired specifications. For hybrid systems, however, the notion of control is much broader than the classical open loop or feedback control found in continuous control theory. Controlled variables exist not only in the continuous domain, but also in the discrete domain. Therefore the controller synthesis problem ask us to design not only continuous control laws but also discrete strategies in order to satisfy the system specification.

In the computer science community, one approach to controller synthesis is the parametric verification problem. As long as one has a decidable class of models, then model checking algorithms can determine ranges of parameter values for which the system will satisfy the specification. A more standard method synthesizes a control automaton which, when composed with the plant automaton, results in the system meeting the desired specification. This has been explored in [11, 75] for timed systems, and [115] for linear hybrid automata.

The control community has naturally generated a variety of control methodologies for hybrid systems, as various frameworks that apply to purely continuous systems have

been extended to capture hybrid systems. Optimal control approaches have been used in [20, 18, 71] to formulate and solve an optimal control problem for a class of hybrid systems, while providing existence of optimal and near-optimal control policies. Relaxed optimal control problems is also the fundamental machinery in the control framework advocated in [54, 81]. There is considerable research effort in the direction of supervisory control of continuous systems [8, 26, 32, 50, 81, 93], which is trying to formalize the control of continuous systems by computer programs. Control methods in the spirit of viability theory [12] have been used by [36, 55]. Control of hybrid systems with integrator dynamics in each location is considered in [101]. Game theoretic approaches were first proposed in [70, 67], and have been applied in automated highway systems [68], and air traffic management systems [102]. Games are very common in the synthesis problem of purely discrete systems [29, 100], but they also have very natural applications in continuous systems [14]. As a result, the game theoretic approach is a uniform controller synthesis platform for purely continuous, purely discrete, and true hybrid systems [69, 103].

Computational Complexity

The above analysis and control techniques face serious challenges in the near future. With the exception of simulation tools, control theoretic approaches are not currently accompanied by meaningful computational tools. Computer scientists do provide computational tools, but it has been recognized, mainly in the control community that uses more sophisticated dynamical models, that the expressive power offered by decidable hybrid systems is limited. However, the main obstacle that significantly limits the application of both computer science and control theoretic approaches is *computational complexity*.

There are two main techniques to deal with complexity: *compositional methods* and *abstraction*. As the main cause of state space explosion is parallel composition, compositional methods attempt to decompose the analysis of a large scale system into a set of smaller problems for individual system components. Very often, however, compositional verification of hybrid systems is difficult due to the strong coupling of system components.

On the other hand, abstraction techniques tackle the complexity involved in verifying that a given large scale system satisfies certain properties, by extracting a simpler but qualitatively equivalent abstracted system, shown in Figure 1.3. Checking the desired property on the abstracted system should be *equivalent* or *sufficient* to checking the prop-

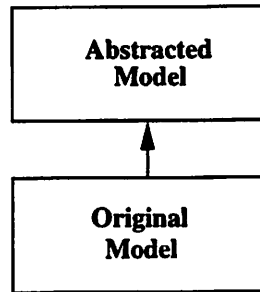


Figure 1.3: System analysis using abstractions

erty on the original system. Abstraction techniques have been rather successful in facing problems of exponential complexity for purely discrete systems [31, 66]. Depending on the property, special graph quotients which preserve the property of interest are constructed.

In addition to complexity reduction, abstraction techniques have been the main tool in expanding the applicability of current decidability results. In particular, [43, 49, 83, 92] have abstracted hybrid systems with complicated dynamics by overapproximating their trajectories by decidable hybrid models. In [90], the reachable set of Lipschitz differential inclusions is overapproximated using rectangular hybrid automata. Such conservative overapproximations are sufficient abstractions, in the sense that verifying the abstraction is a sufficient but not necessary condition. If the abstracted system satisfies the property, then so does the original system. If, however, the abstracted system does not satisfy the required specification, then this may be attributed to the redundant trajectories feasible in the abstracted system but not by the original system.

In addition to the analysis of large scale systems, abstractions are also extremely useful in hierarchical system design. The main classes of hierarchical structures are nicely described and classified in [78]. Figure 1.4 shows a typical two-layer control hierarchy which is frequently used in the quite common planning and control hierarchical systems. In this layered control paradigm, each layer has different objectives. In performing their tasks, the higher level uses an abstracted model of the lower level. One of the main challenges in hierarchical systems is the extraction of a hierarchy of models at various levels of abstraction which are compatible with the functionality and objectives of each layer. A theory of abstraction would be critical in designing valid hierarchies.

Hierarchical systems for discrete event systems have been formally considered in [25, 113, 114, 119]. The supervisory control framework of Figure 1.2 is another example

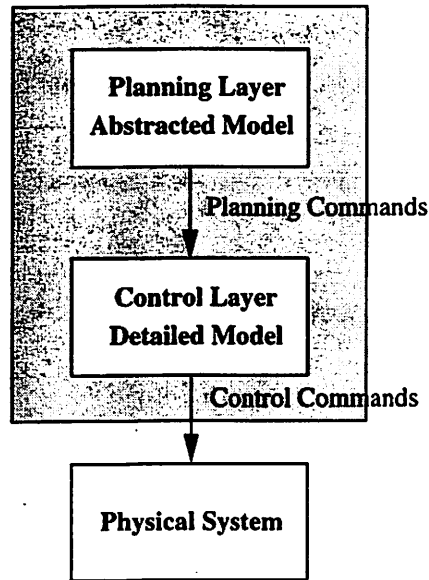


Figure 1.4: Two layer control hierarchy

of a hierarchical system consisting of discrete abstractions of continuous systems. This has been considered in [24, 26] as well as [8, 32, 93]. For purely continuous systems, the only existing notions of aggregation are in the spirit of model reduction [9, 57, 58, 59, 60]. These methods perform only state aggregation as opposed to both state and input aggregation. The above approaches, however, are quite a distance away from being applicable to truly large scale, hierarchical systems, like Automated Highway Systems [109], and Air Traffic Management Systems [88, 95].

1.2 Issues Addressed and Dissertation Outline

Despite the initial progress in developing a theoretical basis for modeling, analyzing, and designing hybrid systems, the majority of the successful developments have been either too discrete or too continuous. The main reason is that there are fundamental limitations to the techniques of both theoretical computer science and control theory. Theoretical computer science techniques are usually of a combinatorial nature due to the lack of structure on the discrete dynamics, and rely on powerful computational tools. However, they are not well suited to handling complicated continuous dynamics. The structure of differential equations naturally suggests the use of deductive techniques for their analysis by control

theorists. Deductive techniques, even if they are conceptually appealing and insightful, will require algorithmic methods in order to be applicable to complex hybrid systems with large numbers of discrete states. It is becoming clear that in order to successfully handle true hybrid systems with many discrete states and complicated dynamics, a synergy of concepts and methods from computer science and control theory is needed.

In Chapter 2, we present the necessary mathematical tools from differential geometry and mathematical logic that will be used in this dissertation. Differential geometry is the natural framework for studying differential equations and control systems, whereas mathematical logic is the heart of theoretical computer science. Even though these two mathematical areas seem disconnected, an amazing bridge between them has been recently built by geometric model theory. This connection is fully exploited in subsequent chapters.

Chapter 3 describes the first attempt to enlarge the modeling frontier of decidable hybrid automata, namely rectangular hybrid automata. In rectangular hybrid automata, continuous variables must satisfy constant, decoupled, rectangular differential inclusions of the form $\dot{x} \in [a, b]$. A natural problem is the characterization of general rectangular differential inclusions which can be transformed to constant, decoupled inclusions by state transformation. The resulting conditions are quite restrictive and presented a serious barrier to extending the decidable classes of hybrid systems. The results of this chapter can also be found in [87].

The goal of Chapter 4 is to extend the known decidable classes of hybrid systems. The main tool for obtaining decidability results for hybrid systems is the concept of bisimulation. If a hybrid system has a *finite* bisimulation, then reachability properties of the original hybrid system can be equivalently checked on a finite, discrete graph. In the search of new classes of hybrid systems with finite bisimulations, the very recent notion of *o-minimal* theories from geometric model theory is used. O-minimal theories connect the seemingly disjoint worlds of geometry and logic presented in Chapter 2. Using this powerful mathematical machinery, the notion of o-minimal hybrid systems is introduced as hybrid systems with all relevant sets and flows definable in an o-minimal theory. It is shown that all o-minimal hybrid systems admit finite bisimulations. This is followed by a list of o-minimal hybrid systems which captures versions of most hybrid systems known to admit finite bisimulations. Furthermore, it includes new classes of hybrid systems with linear dynamics in each discrete location. This result is the evolution of previous attempts which were more geometric in nature [61, 65], but were restricted to planar dynamics. Showing

that o-minimal hybrid systems admit finite bisimulations, must be followed by methods to construct them, in order to obtain new classes of hybrid systems with a decidable reachability problem. The constructive methods that are used come from mathematical logic. Sets are symbolically represented as formulas in first order logic, and reachability calculations are performed using *quantifier elimination* techniques. Since quantifier elimination is possible for the theory of reals with addition, we either find or transform subclasses of o-minimal hybrid systems which are definable in this theory. This procedure results in the first class of hybrid system with linear dynamics in each discrete location with a decidable reachability problem. Chapter 4 is a review of the results in [62, 63].

Whereas Chapter 4 is concerned with extracting discrete abstractions from hybrid systems, Chapter 5 focuses on continuous abstractions of continuous control systems. In particular, Chapter 5 introduces a notion of control system abstraction. Given a control system, and a map which performs state aggregation, an abstracted system is any control system which overapproximates the abstracted trajectories of the original system. This notion of abstraction is formalized by generalizing the classical notion of Φ -related vector fields to control systems. Furthermore, this notion mathematically formalizes the concept of virtual inputs used in backstepping designs. In hierarchical systems, however, aggregation is not independent of the functionality of the layer at which the abstracted system will be used. Our goal is to not only extract abstractions of control systems, but to also ensure that certain properties propagate from the abstracted to the original model. Properties of interest include reachability, controllability, stabilizability, and trajectory tracking. Reachability preserving abstractions are defined as *consistent*, in the sense that controllability requests from the abstracted systems are *implementable* by the detailed original model. We focus on controllability of linear control systems and characterize consistent linear abstractions. In this spirit, a hierarchical controllability criterion is obtained for large scale, linear systems. Intuitively, instead of checking controllability of a large scale system, we construct a sequence of consistent abstractions and then check the controllability of a system which is much smaller in size. Consistency will then propagate controllability along this sequence of abstractions from the simpler quotient system to the original complex system. The computational advantages of this approach are verified by recovering the best known controllability algorithms from numerical linear algebra [39], as a special case of the hierarchical controllability criterion. Chapter 5 reviews most of the material in [86] and [85, 89].

Finally, Chapter 6 presents many directions for future research. As the field of hybrid systems is young, there are many more questions than answers...

Chapter 2

Mathematical Background

In this chapter, we review some facts from differential geometry, subanalytic geometry, and mathematical logic. Differential geometry is the natural mathematical framework for nonlinear control systems and geometric control theory. Subanalytic sets is the richest class of sets that is closed under unions, intersections, complements, forward, and inverse maps. These operations of subanalytic sets are the main ingredients of first order logic, which is reviewed along with some elementary model theory. In Chapter 4, the above notions will be directly linked, as first order predicate logic will be used to capture subanalytic sets.

2.1 Differential Geometry

Our treatment of differential geometry follows that of [51]. For a more thorough introduction to geometry, the reader may wish to consult numerous books on the subject such as [1, 96].

2.1.1 Differentiable Manifolds

Recall that a function $h : A \rightarrow B$ is a homeomorphism iff h is a bijection and both h and h^{-1} are continuous. In this case, topological spaces A and B are called homeomorphic. A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is called smooth or C^∞ if all derivatives of any order exist and are continuous. Function f is real analytic or C^ω , if it is C^∞ and for each $x \in \mathbb{R}^n$ there exists a neighborhood U of x , such that the Taylor series expansion of f at x converges to $f(x)$ for

all $x \in U$. A mapping $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a collection (f_1, \dots, f_m) of functions $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$. The mapping f is smooth (analytic) if all functions f_i are smooth (analytic).

Definition 2.1 (Manifolds). *A manifold M of dimension n is a metric space¹ which is locally homeomorphic to \mathbb{R}^n .*

A manifold, which is of great interest to us, is \mathbb{R}^n itself. A subset N of a manifold M which is itself a manifold is called a submanifold of M . Any open subset N of a manifold M is clearly a submanifold, since if M is locally homeomorphic to \mathbb{R}^n then so is N . In particular, an open interval $I \subseteq \mathbb{R}$ is also a manifold.

A coordinate chart on a manifold M is a pair (U, ϕ) where U is an open set of M and ϕ is a homeomorphism of U on an open set of \mathbb{R}^n . The function ϕ is also called a coordinate function and can also be written as (ϕ_1, \dots, ϕ_n) where $\phi_i : M \rightarrow \mathbb{R}$. If $p \in U$ then $\phi(p) = (\phi_1(p), \dots, \phi_n(p))$ is called the set of local coordinates in the chart (U, ϕ) .

When doing operations on a manifold, we must ensure that our results are consistent regardless of the particular chart we use. We must therefore impose some conditions. Two charts (U, ϕ) and (V, ψ) with $U \cap V \neq \emptyset$, are called C^∞ (C^ω) compatible if the map

$$\psi \circ \phi^{-1} : \phi(U \cap V) \subseteq \mathbb{R}^n \rightarrow \psi(U \cap V) \subseteq \mathbb{R}^n$$

is a C^∞ (C^ω) function. A C^∞ (C^ω) atlas on a manifold M is a collection of charts (U_α, ϕ_α) with $\alpha \in A$ which are C^∞ (C^ω) compatible and such that the open sets U_α cover the manifold M , so $M = \bigcup_{\alpha \in A} U_\alpha$. An atlas is called maximal if it is not contained in any other atlas.

Definition 2.2 (Differentiable Manifolds). *A differentiable (analytic) manifold is a manifold with a maximal, C^∞ (C^ω) atlas.*

Now that we have imposed this differential structure on our manifold M we can perform calculus on M . In particular let $f : M \rightarrow \mathbb{R}$ be a map. If (U, ϕ) is a chart on M then the function

$$\hat{f} = f \circ \phi^{-1} : \phi(U) \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$$

is called the local representative of f in the chart (U, ϕ) . We therefore define the map f to be smooth (analytic) if its local representative \hat{f} is smooth (analytic). Notice if f is

¹More generally, we may replace metric space with Hausdorff and second countable topological space

smooth (analytic) in one chart, then it is smooth (analytic) in every chart since we required our charts to be C^∞ (C^ω) compatible and our atlas to be maximal. Hence our results are intrinsic to the manifold and do not depend on the particular chart we use. Similarly, if we have a map $f : M \rightarrow N$, where M, N are differentiable manifolds, the local representation of f given a chart (U, ϕ) of M and (V, ψ) of N is

$$\hat{f} = \psi \circ f \circ \phi^{-1}$$

which makes sense only if $f(U) \cap V \neq \emptyset$. Again f is smooth (analytic) if \hat{f} is a smooth (analytic) map.

Let $f : M \rightarrow N$ be a map between two manifolds. The map f is called a diffeomorphism if both f and f^{-1} are smooth. In this case, manifolds M and N are called diffeomorphic.

The rank of a smooth map $f : M \rightarrow N$ at $p \in M$ is defined to be the rank of the Jacobian matrix of f expressed in local coordinates. The rank is independent of the particular choice of coordinate charts used. If the rank of f is equal to the dimension of M for all $p \in M$, then f is called an immersion. If the rank of f is equal to the dimension of N for all $p \in M$, then f is called a submersion. If $f : M \rightarrow N$ is an injective immersion, then $f(M)$ is called an immersed submanifold of N . If, in addition, the topology induced on $f(M)$ from M coincides with the topology of $f(M)$ as a subset of N , then $f(M)$ is an embedded submanifold of N .

2.1.2 Tangent Spaces

Let p be a point on a manifold M and let $C^\infty(p)$ denote the vector space of all smooth functions in a neighborhood of p . A tangent vector X_p at $p \in M$ is an operator from $C^\infty(p)$ to \mathbb{R} which satisfies for $f, g \in C^\infty(p)$ and $a, b \in \mathbb{R}$, the following properties,

1. Linearity $X_p(a \cdot f + b \cdot g) = a \cdot X_p(f) + b \cdot X_p(g)$
2. Derivation $X_p(f \cdot g) = f(p) \cdot X_p(g) + X_p(f) \cdot g(p)$

The set of all tangent vectors at $p \in M$ is called the tangent space of M at p and is denoted by $T_p M$. The tangent space $T_p M$ becomes a vector space over \mathbb{R} if for tangent vectors X_p, Y_p and real numbers c_1, c_2 we define

$$(c_1 \cdot X_p + c_2 \cdot Y_p)(f) = c_1 \cdot X_p(f) + c_2 \cdot Y_p(f)$$

for any smooth function f in the neighborhood of p . The collection of all tangent spaces of the manifold,

$$TM = \bigcup_{p \in M} T_p M$$

is called the tangent bundle. The tangent bundle has a naturally associated projection map $\pi : TM \rightarrow M$ taking a tangent vector $X_p \in T_p M \subset TM$ to the point $p \in M$. The tangent space $T_p M$ can then be thought of as $\pi^{-1}(p)$.

The tangent space can be thought of as a special case of a more general mathematical object called a fiber bundle. Loosely speaking, a fiber bundle can be thought of as gluing sets at each point of the manifold in a smooth way.

Definition 2.3 (Fiber Bundles [82]). *A fiber bundle is a tuple $(B, M, \pi, U, \{O_i\}_{i \in I})$ where B, M, U are smooth manifolds called the total space, the base space, and the standard fiber respectively. The map $\pi : B \rightarrow M$ 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 $\Phi_i : \pi^{-1}(O_i) \rightarrow O_i \times U$ satisfying*

$$\pi \circ \Phi_i = \pi$$

where $\pi \circ$ is the projection from $O_i \times U$ to O_i . The submanifold $\pi^{-1}(p)$ is called the fiber at $p \in M$. If all the fibers are vector spaces of constant dimension, then the fiber bundle is called a vector bundle.

The tangent bundle is a vector bundle and the fiber at each point $p \in M$ is the tangent space $T_p M$. From Definition 2.3 it is clear that fiber bundles are locally diffeomorphic to $O_i \times U$. Therefore, fiber bundles are manifolds of dimension $n_M + n_U$ where n_M and n_U are the dimensions of M and U respectively. In particular, the tangent bundle TM has dimension $2n$.

Now let M be a manifold and let (U, ϕ) be a chart containing the point p . In this chart we can associate the following tangent vectors

$$\frac{\partial}{\partial \phi_1}, \dots, \frac{\partial}{\partial \phi_n}$$

defined by

$$\frac{\partial}{\partial \phi_i}(f) = \frac{\partial(f \circ \phi^{-1})}{\partial x_i}$$

for any smooth function $f \in C^\infty(p)$. The tangent space T_pM is an n -dimensional vector space and if (U, ϕ) is a local chart around p then the tangent vectors

$$\frac{\partial}{\partial \phi_1}, \dots, \frac{\partial}{\partial \phi_n}$$

form a basis for T_pM . Therefore if X_p is a tangent vector at p then

$$X_p = \sum_{i=1}^n a_i \frac{\partial}{\partial \phi_i}$$

where a_1, \dots, a_n are real numbers. From the above formula we can see that $X_p(f)$ is an operator which simply takes the directional derivative of f in the direction of $[a_1, \dots, a_n]$.

Now let M and N be smooth manifolds and $f : M \rightarrow N$ be a smooth map. Let $p \in M$ and let $q = f(p) \in N$. We wish to push forward tangent vectors from T_pM to T_qN using the map f . The natural way to do this is by defining a map $f_* : T_pM \rightarrow T_qN$ by

$$(f_*(X_p))(g) = X_p(g \circ f)$$

for smooth functions g in the neighborhood of q . One can easily check that $f_*(X_p)$ is a linear operator and a derivation and thus a tangent vector. The map $f_* : T_pM \rightarrow T_{f(p)}N$ is called the push forward map of f . The push forward map $f_* : T_pM \rightarrow T_{f(p)}N$ is a linear map, and furthermore if $f : M \rightarrow N$ and $g : N \rightarrow K$ then

$$(g \circ f)_* = g_* \circ f_*$$

which is essentially the chain rule.

2.1.3 Vector Fields

A vector field on a manifold M is a smooth map X which places at each point p of M a tangent vector from T_pM . Therefore since a vector field, X , places at each point p a tangent vector $X(p)$ we have that in the chart (U, ϕ) the local expression for the vector field X is

$$X(p) = \sum_{i=1}^n a_i(p) \frac{\partial}{\partial \phi_i}$$

The vector field is smooth (analytic) if and only if $a_i(p)$ is C^∞ (C^ω).

Let $I \subseteq \mathbb{R}$ be an open interval containing the origin. An integral curve of a vector field is a curve $c : I \rightarrow M$ whose tangent at each point is identically equal to the vector field at that point. Therefore an integral curve satisfies for all $t \in I$,

$$c' = c_*(1) = X(c)$$

A vector field is called complete if the integral curve passing through every $p \in M$ can be extended for all time, that is we can choose $I = \mathbb{R}$. Integral curves of smooth (analytic) vector fields are smooth (analytic).

Now let $\Phi : M \rightarrow N$ be a smooth map between two manifolds and let X be a vector field on M . At every point $p \in M$ we can use Φ_* to push forward $X(p)$ of the vector field to $T_{f(p)}N$. If Φ is a diffeomorphism, then this procedure results in a well defined vector field on N denoted $\Phi_*(X)$. If Φ is surjective, then $\Phi_*(X)$ is a well defined vector field only when Φ and X are such that $\Phi_*(X_{p_1}) = \Phi_*(X_{p_2})$ whenever $\Phi(p_1) = \Phi(p_2)$. This is captured by the following definition.

Definition 2.4 (Φ -related Vector Fields). *Let X and Y be vector fields on manifolds M and N respectively and $\Phi : M \rightarrow N$ be a smooth map. Then X and Y are Φ -related iff the following diagram commutes*

$$\begin{array}{ccc} M & \xrightarrow{\Phi} & N \\ X \downarrow & & Y \downarrow \\ TM & \xrightarrow{\Phi_*} & TN \end{array} \quad (2.1)$$

or in other words iff $\Phi_* \circ X = Y \circ \Phi$.

If Φ is not surjective, then X may be Φ -related to many vector fields on N . If, however, Φ is surjective, then X can only be Φ -related to a unique vector field on N .

Given two vector fields X and Y on manifold M , we define their Lie bracket, denoted $[X, Y]$, by the following rule

$$[X, Y]_p(f) = X_p(Y(f)) - Y_p(X(f)) \quad (2.2)$$

for functions $f \in C^\infty(p)$. It can be easily verified that $[X, Y]_p \in T_pM$, and thus $[X, Y]$ is indeed a vector field. If X and Y are given in local coordinates as vectors $f(x)$, $g(x)$, then the expression for their Lie bracket $[f, g]$ in local coordinates is simply

$$[f, g] = \frac{\partial g}{\partial x} f - \frac{\partial f}{\partial x} g \quad (2.3)$$

Finally, there is an interesting relation between Lie brackets and Φ -related vector fields. Let $\Phi : M \rightarrow N$ be a surjection, let X_1 and X_2 be two smooth vector fields on M , and let $\Phi_*(X_1)$ and $\Phi_*(X_2)$ be Φ -related to X_1 and X_2 respectively. Then

$$\Phi_*([X_1, X_2]) = [\Phi_*(X_1), \Phi_*(X_2)] \quad (2.4)$$

The above fact is, of course, also true when Φ is a diffeomorphism.

2.2 Subanalytic Geometry

In much of the subsequent analysis, we shall be dealing with sets and operations on sets. In general, if one starts with a general class of sets, and performs on them unions, complements, closures, intersections, and projections, then either the class of sets is closed under these operations, or new, more complicated sets emerge. In this section, we present classes of sets which are closed under unions, complements as well as forward and inverse maps.

Definition 2.5 (Boolean Algebras of Sets). *A boolean algebra of a set X is a nonempty collection \mathcal{C} of subsets of X , $\mathcal{C} \subseteq 2^X$, such that if $A, B \in \mathcal{C}$, then $A \cup B$ and $X \setminus A$ belong to \mathcal{C} .*

It is immediate from the above definition that $\emptyset, X \in \mathcal{C}$, and if $A, B \in \mathcal{C}$ then $A \cap B \in \mathcal{C}$. Given a family of sets $\mathcal{A} = (A_i)_{i \in K}$, with $K = \{1, \dots, n\}$, we denote by $B(\mathcal{A})$ the boolean algebra of sets generated by $(A_i)_{i \in K}$, that is the smallest boolean algebra containing $(A_i)_{i \in K}$. It can be shown that the elements of $B(\mathcal{A})$ are exactly the finite unions of sets of the form

$$\left(\bigcap_{i \in J} A_i \right) \cap \left(\bigcap_{i \notin J} X \setminus A_i \right) \quad (2.5)$$

with $J \subseteq K$. Consider now sets of the form

$$\{x \in \mathbb{R}^n \mid f_1(x) = 0, \dots, f_p(x) = 0, g_1(x) > 0, \dots, g_q(x) > 0\}$$

where functions $f_1, \dots, f_p, g_1, \dots, g_q : \mathbb{R}^n \rightarrow \mathbb{R}$ are of the form $a_n x_n + \dots + a_1 x_1 + a_0$. Such sets are called *basic semilinear* sets. A *semilinear* set is a finite union of basic semilinear sets. If functions $f_1, \dots, f_p, g_1, \dots, g_q$ are allowed to be polynomials in x_1, \dots, x_n , then

we obtain the analogous notions of *basic semialgebraic* and *semialgebraic* sets. Clearly, semilinear sets are a special case of semialgebraic sets. The boolean algebra generated by semilinear (semialgebraic) sets is well known to be a closed family of sets with respect to linear (polynomial) maps.

2.2.1 Semianalytic and Subanalytic Sets

In our search for a rich family of well behaved sets, sets defined by smooth functions appear as the next obvious candidate. Unfortunately, given any closed set Z of the real line (for example the Cantor set), there exists a smooth function $f : \mathbb{R} \rightarrow \mathbb{R}$ such that $Z = \{x \in \mathbb{R} \mid f(x) = 0\}$. Fortunately, real analytic functions are free from such pathologies. The following classical result illustrates this point.

Proposition 2.6. *Let $I \subseteq \mathbb{R}$ be an open interval and $f : I \rightarrow \mathbb{R}$ be an analytic function. Let $Z = \{x \in I \mid f(x) = 0\}$. Then, either $Z = I$ or Z has no accumulation point in I . Also, if f is not identically zero, then every compact subset K of I contains at most a finite number of zeros of f .*

The above proposition has motivated the use of analytic functions for describing subsets of \mathbb{R}^n . Given open neighborhood $U \subseteq \mathbb{R}^n$, let $C^\omega(U, \mathbb{R})$ denote the set of analytic functions from U to \mathbb{R} . Denote by $B(C^\omega(U, \mathbb{R}))$ the boolean algebra generated by sets of the form

$$\{x \in U \mid f_1(x) = 0, \dots, f_p(x) = 0, g_1(x) > 0, \dots, g_q(x) > 0\} \quad (2.6)$$

where $f_1, \dots, f_p, g_1, \dots, g_q \in C^\omega(U, \mathbb{R})$.

Definition 2.7 (Semianalytic Sets). *A subset A of \mathbb{R}^n is semianalytic if for every $x \in \mathbb{R}^n$, there is an open neighborhood U of x such that $A \cap U \in B(C^\omega(U, \mathbb{R}))$.*

Semianalytic sets can therefore be locally described by a finite number of equalities and inequalities of analytic functions. It is clear from the definition that semianalytic sets are closed under complementation, and locally finite unions and intersections. Unfortunately, images of semianalytic sets under analytic maps are not in general semianalytic. However, semianalytic sets can be enlarged to a larger class which has this desirable property.

Definition 2.8 (Subanalytic Sets). *A subset A of \mathbb{R}^n is subanalytic if for every $x \in \mathbb{R}^n$ there is an open neighborhood U of x , and a bounded semianalytic set $Y \subset \mathbb{R}^{n+m}$, such that $A \cap U$ is the projection of Y into U .*

Therefore, subanalytic sets are projections of semianalytic sets. Even though this construction forces closure with respect to analytic maps, it endangers closure with respect to complementation. Fortunately, the following proposition summarizes the nice properties of subanalytic sets. Recall that a map f is called proper if $f^{-1}(K)$ is compact whenever K is.

Proposition 2.9 (Properties of Subanalytic Sets [16]). *The class of subanalytic sets has the following properties*

1. *Subanalytic sets are closed under locally finite unions and intersections.*
2. *If A is subanalytic, then $\mathbb{R}^n \setminus A$ is subanalytic.*
3. *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be an analytic map. If A is subanalytic, then $f^{-1}(A)$ is subanalytic.*
4. *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be an analytic, proper map. If A is subanalytic, then $f(A)$ is subanalytic*

Example 1. Points are subanalytic, and so is any locally finite union of points, for example \mathbb{Z}^n as subset of \mathbb{R}^n . Clearly \emptyset and \mathbb{R}^n are also subanalytic. Let $a, b \in \mathbb{R}$, $a < b$, then $[a, b]$, (a, b) , $[a, b)$ and $(a, b]$ are subanalytic in \mathbb{R} . Let $B(p, r)$ be the open ball centered at p of radius r in \mathbb{R}^n . Then $B(p, r)$ is subanalytic. In general, as is clear from the definition, semianalytic sets are subanalytic. In particular, any semialgebraic or semilinear subset of \mathbb{R}^n is subanalytic.

The following example shows an undesirable set that is not subanalytic.

Example 2. Consider the set $Z = \{\frac{1}{n} : n \in \mathbb{N}\}$. The set Z is not subanalytic. To see why simply consider any open neighborhood U of the origin. But then, by Proposition 2.6, $U \cap Z$ cannot be expressed as the zero set of a analytic function.

The above example suggests that graphs of analytic functions can only have locally finite intersections with subanalytic sets. Such good intersection properties may be useful in avoiding zeno hybrid systems, which exhibit infinite switching in finite time. In the next section, we describe well known results about subanalytic sets, that may be useful in such areas of research. However, the results of the next section will not be used in the next chapters.

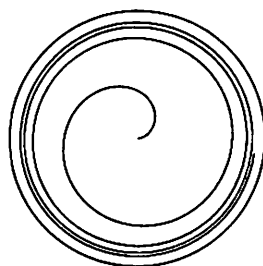


Figure 2.1: Example of a partition but not a stratification

2.2.2 Subanalytic Stratifications

Stratifications are special partitions of sets that have a very nice hierarchical structure. Roughly, the boundary of each set is a set of lower dimension. To give a formal definition, denote by \bar{S} the closure of a set S , and consider,

Definition 2.10 (Stratifications). *An analytic (C^ω) stratification of \mathbb{R}^n is a partition \mathcal{S} of \mathbb{R}^n with the following properties:*

1. *each $S \in \mathcal{S}$ is a connected analytic embedded submanifold of \mathbb{R}^n ,*
2. *\mathcal{S} is locally finite,*
3. *given two sets $S, P \in \mathcal{S}$, $P \neq S$, such that $S \cap \bar{P} \neq \emptyset$ then $S \subset \bar{P}$ and $\dim S < \dim P$*

The sets in a stratification are called strata.

Example 3. Consider the partition of the plane into the four embedded submanifolds according to Figure 2.1. The two 2-dimensional strata are the complement of the closed unit disk, and the complement of the spiral in the open unit disk. The two 1-dimensional strata are the unit circle and the spiral. Notice that the unit circle is contained in the closure of the spiral and yet it has the same dimension. Therefore, this partition is not a stratification.

The following theorem is a powerful property of subanalytic sets.

Theorem 2.11 (Subanalytic Stratification [42]). *Let \mathcal{A} be a locally finite collection of subanalytic sets of \mathbb{R}^n . Then there is a C^ω stratification \mathcal{S} of \mathbb{R}^n such that:*

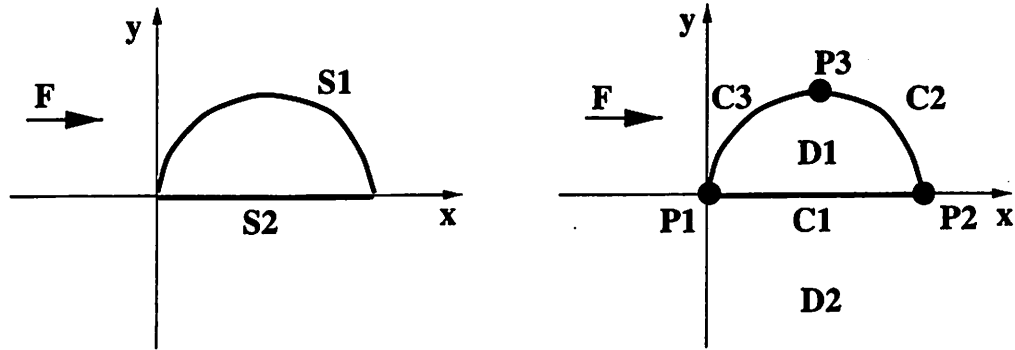


Figure 2.2: Example of subanalytic stratification

1. All strata in S are subanalytic,
2. S is compatible with \mathcal{A} . That is, every set in \mathcal{A} is a union of strata from S .

In fact, one can stratify \mathbb{R}^n in a manner that is compatible not only with a collection of subanalytic sets, but also with a finite number of analytic vector fields.

Theorem 2.12 (Subanalytic Stratifications with Vector Fields [97]). *Let \mathcal{A} be a locally finite family of nonempty subanalytic subsets of \mathbb{R}^n . For each $A \in \mathcal{A}$, let $F(A)$ be a finite set of real analytic vector fields on \mathbb{R}^n . Then there exists a subanalytic stratification S of \mathbb{R}^n , compatible with \mathcal{A} , and having the property that, whenever $S \in S$, $S \subset A$, $A \in \mathcal{A}$, $X \in F(A)$, then either (i) X is everywhere tangent to S or (ii) X is everywhere transversal to S .*

The above theorem is illustrated by the following example.

Example 4. Let F be the following analytic vector field on \mathbb{R}^2

$$\begin{aligned} \dot{x} &= x^2 + y^2 \\ \dot{y} &= 0 \end{aligned}$$

which has an isolated equilibrium at the origin and points in the positive x -direction otherwise. Consider the following two subanalytic sets

$$\begin{aligned} S_1 &= \{(x, y) \in \mathbb{R}^2 \mid y \geq 0 \text{ and } (x-1)^2 + y^2 = 1\} \\ S_2 &= \{(x, y) \in \mathbb{R}^2 \mid y = 0 \text{ and } 0 \leq x \leq 2\} \end{aligned}$$

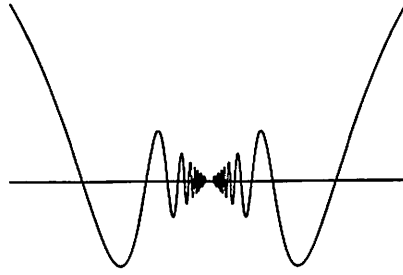


Figure 2.3: Infinite crossings on a bounded interval

shown in Figure 2.2. A subanalytic stratification of \mathbb{R}^2 which is compatible with the sets S_1 , S_2 and the vector field F is also shown in Figure 2.2. It consists of

- 0-dimensional strata

- $P_1 = (0, 0)$, $P_2 = (2, 0)$, and $P_3 = (1, 1)$

- 1-dimensional strata

- $C_1 = \{(x, y) \in \mathbb{R}^2 \mid y = 0 \text{ and } 0 < x < 2\}$

- $C_2 = \{(x, y) \in \mathbb{R}^2 \mid y > 0 \text{ and } 1 < x < 2 \text{ and } (x - 1)^2 + y^2 = 1\}$

- $C_3 = \{(x, y) \in \mathbb{R}^2 \mid y > 0 \text{ and } 0 < x < 1 \text{ and } (x - 1)^2 + y^2 = 1\}$

- 2-dimensional strata

- $D_1 = \{(x, y) \in \mathbb{R}^2 \mid y > 0 \text{ and } (x - 1)^2 + y^2 < 1\}$

- $D_2 = \mathbb{R}^2 \setminus \{P_1, P_2, P_3, C_1, C_2, C_3, D_1\}$

Notice that the vector field is tangent to P_1 since it is an equilibrium as well as to C_1 , D_1 and D_2 . The vector field is transverse to all the other strata. Moreover, $S_1 = P_1 \cup P_2 \cup P_3 \cup C_2 \cup C_3$ and $S_2 = P_1 \cup P_2 \cup C_1$.

The following proposition illustrates some of the good intersection properties that analytic curves have with subanalytic sets. The finiteness property indicated in the proposition makes it possible to define transitions between strata in a natural way.

Proposition 2.13. *Let I be an open interval, M a real analytic manifold and $\gamma : I \rightarrow M$ a real analytic function. Let \mathcal{S} be an analytic stratification of M by subanalytic sets. If*

$[a, b] \subset I$ then there exists a finite partition $\{x_1, \dots, x_n\}$ of $[a, b]$ with the property that for each $i = 1, \dots, n - 1$ there exists a stratum $S_i \in \mathcal{S}$ such that $\gamma((x_i, x_{i+1})) \subset S_i$.

Such good finiteness properties are useful in having well defined discrete abstractions of continuous systems. The following example shows that the assumption of subanalyticity in the proposition above can not be dropped.

Example 5. Consider the stratification of \mathbb{R}^2 by the following five sets (see Figure 2.3).

$$\begin{aligned} S_1 &= \{(0, 0)\} \\ S_2 &= \left\{ (x, y) : x > 0 \wedge y = x \sin \frac{1}{x} \right\} \\ S_3 &= \left\{ (x, y) : x < 0 \wedge y = x \sin \frac{1}{x} \right\} \\ S_4 &= \left\{ (x, y) : x \neq 0 \wedge y > x \sin \frac{1}{x} \right\} \cup \{(0, y) : y > 0\} \\ S_5 &= \left\{ (x, y) : x \neq 0 \wedge y < x \sin \frac{1}{x} \right\} \cup \{(0, y) : y < 0\} \end{aligned}$$

Each set is an embedded analytic submanifold of \mathbb{R}^2 and they clearly satisfy the condition on the dimension of the strata in the closure of other strata. Finally, consider the constant vector field $X = \frac{\partial}{\partial x}$. Then the integral curve γ of X through $(0, 0)$ is the x -axis (parameterized by x itself). Therefore, the image by γ of any interval containing 0 intersects both S_4 and S_5 an infinite number of times.

For other important results on subanalytic sets as well as their relevance to control theory, the reader is referred to [16], [42], and [97].

2.3 Mathematical Logic

In this section we give a brief introduction to mathematical logic and model theory. Logic will serve as the main computational tool for symbolically representing sets as well as performing boolean operations on them. The reader is referred to [104] for a more detailed introduction.

2.3.1 Languages and Formulas

A language is a set of symbols separated into three groups: relations, functions and constants. More formally, a language is $\mathcal{L} = \{R_1, \dots, R_n, f_1, \dots, f_m, c_0, \dots, c_l\}$, where R_1, \dots, R_n are the relation symbols, f_1, \dots, f_m are the function symbols, and c_0, \dots, c_l are symbols for constants. For example, the sets $\mathcal{P} = \{<, +, -, \cdot, \exp, 0, 1\}$, $\mathcal{R} = \{<, +, -, \cdot, 0, 1\}$, and $\mathcal{R}_{\text{exp}} = \{<, +, -, \cdot, \exp, 0, 1\}$ are examples of languages where $<$ (less than) is the relation, $+$ (plus), $-$ (minus), \cdot (product), and \exp (exponentiation) are the functions, and 0 (zero) and 1 (one) are the constants.

Let $\mathcal{L} = \{R_1, \dots, R_n, f_1, \dots, f_m, c_0, \dots, c_l\}$ be a language, and x_0, x_1, \dots be a countable set of variables. We define the following two syntactical categories.

Definition 2.14 (Terms). *The set of terms of \mathcal{L} are defined inductively as follows*

1. *Constants and variables are terms*
2. *If t_1, \dots, t_m are terms, and f is a function, then $f(t_1, \dots, t_m)$ is a term.*

For instance, $x - 2y + 3$ and $x + yz^2 - 1$ are terms of \mathcal{P} and \mathcal{R} , respectively. In other words, terms of \mathcal{P} are linear expressions, and terms of \mathcal{R} are polynomials with integer coefficients. Notice that integers are the only numbers allowed in expressions (they can be obtained by adding up the constant 1).

Definition 2.15 (Atomic Formulas). *The atomic formulas of a language are of the form $t_1 = t_2$, or $R(t_1, \dots, t_n)$, where $t_i, i = 1, \dots, n$ are terms and R is an n -ary relation.*

For example, $xy > 0$ and $x^2 + 1 = 0$ are atomic formulas of \mathcal{R} . Note that the equality symbol $=$ is part of our language even though it was not explicit in the set of language symbols. In general, we will assume that every language contains the equality symbol.

Definition 2.16 (First-order Formulas). *The set of first-order formulas of language \mathcal{L} is recursively defined as follows:*

1. *Atomic formulas are formulas*
2. *If ϕ, ψ are formulas, then $\phi \wedge \psi$, $\phi \vee \psi$, and $\neg\phi$ are formulas*
3. *If ϕ is a formula, then $\forall x_i : \phi$ and $\exists x_i : \phi$ are formulas*

Formulas defined in a language \mathcal{L} are called \mathcal{L} -formulas. Examples of \mathcal{R} -formulas are:

$$\forall x \forall y : xy > 0 \quad (2.7)$$

$$\exists x : x^2 - 2 = 0 \quad (2.8)$$

$$\exists w : xw^2 + yw + z = 0 \quad (2.9)$$

The occurrence of a variable in a formula is *free* if it is not inside the scope of a quantifier; otherwise, it is *bound*. For example, x , y , and z are free and w is bound in (2.9). We often write $\phi(x_1, \dots, x_n)$ to indicate that x_1, \dots, x_n are the free variables of the formula ϕ . A *sentence* of \mathcal{R} is a formula with no free variables. Formulas (2.7) and (2.8) are sentences.

2.3.2 Model Theory

Syntax would not be interesting without semantics. A *model* appropriate to a language consists of a non-empty set S and a semantic interpretation of the relations, functions and constants. For example, $(\mathbb{R}, <, +, -, \cdot, 0, 1)$ and $(\mathbb{Q}, <, +, -, \cdot, 0, 1)$, are models assigning the usual meaning to symbols of \mathcal{R} .

Every sentence of a language will be either true or false in a given model. For instance, formula (2.8) is true over \mathbb{R} , but false over \mathbb{Q} . Formulas that are not sentences may hold for some assignments of values to the free variables but not for others. For instance, formula (2.9) holds in \mathbb{R} for the assignment $(1, 1, 0)$ of (x, y, z) , but not for $(1, 0, 1)$ (there is no real number w such that $w^2 + 1 = 0$).

We say that a set $Y \subseteq S^n$ is \mathcal{L} -*definable* or simply *definable* in a language \mathcal{L} , if there exists a formula $\phi(x_1, \dots, x_n)$ such that

$$Y = \{(a_1, \dots, a_n) \in S^n \mid \phi(a_1, \dots, a_n)\} \quad (2.10)$$

For example, over \mathbb{R} , the formula $x^2 - 2 = 0$ defines the set $\{\sqrt{2}, -\sqrt{2}\}$. Two formulas $\phi(x_1, \dots, x_n)$ and $\psi(x_1, \dots, x_n)$ are *equivalent* in a model, denoted by $\phi \equiv \psi$, if for every assignment (a_1, \dots, a_n) of (x_1, \dots, x_n) , $\phi(a_1, \dots, a_n)$ is true if and only if $\psi(a_1, \dots, a_n)$ is true. Equivalent formulas define the same set.

Example 6. As an interesting example consider the vector field defined by the differential equation

$$\begin{aligned}\dot{x}_1 &= 2 \\ \dot{x}_2 &= -1\end{aligned}\tag{2.11}$$

Let $Y = \{(y_1, y_2) \in \mathbb{R}^2 \mid \phi(y_1, y_2)\}$ be a \mathcal{R} -definable set. Let $Pre(Y)$ be the set of all points $(x_1, x_2) \in \mathbb{R}^2$ that can reach a point $(y_1, y_2) \in Y$ following a trajectory satisfying (2.11). Then $Pre(Y)$ is also \mathcal{R} -definable since $Pre(Y) = \{(x_1, x_2) \in \mathbb{R}^2 \mid \psi(x_1, x_2)\}$, where

$$\psi(x_1, x_2) \triangleq \exists y_1 \exists y_2 \exists t : \phi(y_1, y_2) \wedge t \geq 0 \wedge y_1 = x_1 + 2t \wedge y_2 = x_2 - t.$$

Example 7. Consider now the linear vector field defined by

$$\begin{aligned}\dot{x}_1 &= 2x_1 \\ \dot{x}_2 &= -x_2\end{aligned}\tag{2.12}$$

The set of points $(x_1, x_2) \in \mathbb{R}^2$ that can reach a point (y_1, y_2) in an \mathcal{R} -definable set Y following a trajectory solution of (2.12) is definable in \mathcal{R}_{exp} . That is $Pre(Y) = \{(x_1, x_2) \in \mathbb{R}^2 \mid \psi(x_1, x_2)\}$ where

$$\psi(x_1, x_2) \triangleq \exists y_1 \exists y_2 \exists t : \phi(y_1, y_2) \wedge t \geq 0 \wedge y_1 = x_1 e^{2t} \wedge y_2 = x_2 e^{-t}$$

2.3.3 Decidability and Quantifier Elimination

Every model defines a *theory* as the set of all sentences which hold in the model. We denote by $\text{Lin}(\mathbb{R})$ the theory defined as the formulas of \mathcal{P} that are true over $(\mathbb{R}, <, +, -, \cdot, 0, 1)$. In other words, $\text{Lin}(\mathbb{R})$ is the theory of linear constraints (polyhedra). We denote by $\text{OF}(\mathbb{R})$ the theory obtained by interpreting \mathcal{R} over $(\mathbb{R}, <, +, -, \cdot, 0, 1)$. In other words, $\text{OF}(\mathbb{R})$ is the set of all true assertions about the set of real numbers when viewed as an *ordered field*. The theory $\text{OF}_{\text{exp}}(\mathbb{R})$ is the extension of the ordered field of real numbers with the exponentiation.

Given a theory, it is important to determine the sentences of the language that belong to the theory. Tarski [98] showed the remarkable fact that $\text{OF}(\mathbb{R})$ is *decidable*, that is a computational procedure that will decide whether any \mathcal{R} -sentence ϕ is true in the model $(\mathbb{R}, <, +, -, \cdot, 0, 1)$. The decision procedure is a two step procedure:

1. Every formula $\phi(x_1, \dots, x_n)$ is converted to an equivalent quantifier free formula $\psi(x_1, \dots, x_n)$.

2. There is an algorithm for deciding the truth of quantifier free sentences.

For example, formula (2.9) is equivalent to the quantifier free formula $4xz - y^2 \leq 0$. Then, given assignments for x, y, z , one can easily decide whether the quantifier free formula is true or false. Theories that admit quantifier elimination have the desirable property that every \mathcal{R} -definable set $Y \subseteq \mathbb{R}^n$ is definable without quantifiers. This immediately shows that every definable set in $\text{OF}(\mathbb{R})$ can be described by the boolean algebra generated by polynomial functions. Therefore, the definable sets in $\text{OF}(\mathbb{R})$ are exactly the semialgebraic sets. A similar line of reasoning shows that the definable sets in $\text{Lin}(\mathbb{R})$, which also admits quantifier elimination, are the semilinear sets.

Moreover, the decidability of a theory implies that there is a computational procedure for checking whether Y is empty. In particular, for decidable theories that admit quantifier elimination, a definable set $Y = \{(y_1, \dots, y_n) \in \mathbb{R}^n \mid \phi(y_1, \dots, y_n)\} = \emptyset$ if and only if the sentence $\exists y_1 \dots \exists y_n : \phi(y_1, \dots, y_n)$ is equivalent to the (quantifier-free) formula *false*. Furthermore, quantifier elimination allows to compute $Pre(Y)$ of Example 6. This reachability calculation is illustrated in the following example.

Example 8. Consider the vector field defined in Example 6 and let $Y = \{(y_1, y_2) \in \mathbb{R}^2 \mid y_1 = 4 \wedge y_2 = 3\}$. Then $Pre(Y) = \{(x_1, x_2) \in \mathbb{R}^2 \mid \psi(x_1, x_2)\}$, where

$$\begin{aligned} \psi(x_1, x_2) &\triangleq \exists y_1 \exists y_2 \exists t : t \geq 0 \wedge y_1 = 4 \wedge y_2 = 3 \wedge y_1 = x_1 + 2t \wedge y_2 = x_2 - t \\ &\equiv \exists t : t \geq 0 \wedge x_1 + 2t = 4 \wedge x_2 - t = 3 \\ &\equiv -(4 - x_1) = 2(3 - x_2) \\ &\equiv x_1 + 2x_2 - 10 = 0 \wedge x_2 - 3 \geq 0. \end{aligned}$$

Tarski's result, even though spectacular, is far from being efficient computationally. More recent approaches to quantifier elimination are based on cylindrical algebraic decomposition techniques [10, 110]. This has resulted in meaningful computational tools that perform quantifier elimination, like REDLOG [38] and QEPCAD [30].

Note that in Example 8, the set $Pre(Y)$ is an \mathcal{R} -definable set, and by the decidability of $\text{OF}(\mathbb{R})$, the formula ψ is equivalent to a quantifier free formula. If we are to use the same approach for Example 7, we immediately run into difficulty as the corresponding formula for $Pre(Y)$ is definable in $\text{OF}_{\text{exp}}(\mathbb{R})$. Tarski envisioned an extension of his decidability result for $\text{OF}(\mathbb{R})$ to the theory of reals with exponentiation $\text{OF}_{\text{exp}}(\mathbb{R})$. Such an extension

is of great interest to control theory, as the exponential function allows us to describe the flows of linear vector fields.

Though it is not known whether $\text{OF}_{\text{exp}}(\mathbb{R})$ is decidable, it has been shown in [105] that the following formula

$$y > 0 \wedge \exists w (wy = x \wedge z = ye^w)$$

is not equivalent to a quantifier-free \mathcal{R}_{exp} -formula. In other words, $\text{OF}_{\text{exp}}(\mathbb{R})$ does not admit quantifier elimination. Even if quantifiers could be eliminated, there is no obvious algorithm for deciding quantifier-free sentences in \mathcal{R}_{exp} , like

$$e^{e^2-2} - e^5 = e^{3+e^{-3}}$$

Deciding whether such sentences are true depends on whether there are no surprising exponential algebraic relations holding over the integers. It is known, that if the famous Schanuel's conjecture in number theory holds, then there are no unexpected exponential-algebraic relationships over the integers. In fact, it has been shown in [73], that if Schanuel's conjecture is true, then the theory $\text{OF}_{\text{exp}}(\mathbb{R})$ is decidable!

Until this issue is resolved², in Chapter 4, we identify several subsets of \mathcal{R}_{exp} where quantifiers can be eliminated. This allows us to perform reachability calculation for classes of linear vector fields.

²Hopefully this will take less than Fermat's Last Theorem!

Chapter 3

Straightening Out Differential Inclusions

Computer aided verification is one of the main, formal approaches for the analysis of hybrid systems. In the verification community, hybrid systems are modeled as hybrid automata where differential equations or inclusions exist in each discrete state of a finite state machine. Transitions from one discrete state to another are triggered by guards on the variables of the system. An example of a hybrid automaton is shown in Figure 3.1. Given a desired specification for a hybrid automaton, such as satisfying certain reachability properties, verification algorithms check whether the system indeed satisfies the desired specification by exactly computing the reachable states of the system. A very important issue in computer aided verification is the decidability of the resulting algorithms.

The state of the art in the verification of hybrid systems is that the reachability problem for *initialized, rectangular hybrid automata* is decidable [91]. Rectangular hybrid automata are automata where in each discrete location the continuous dynamics are described by *decoupled, constant, rectangular differential inclusions*. Thus, the time derivative of each continuous variable must belong to a constant interval of the form $[a, b] \subset \mathbb{R}$, as shown in Figure 3.1. Furthermore, checking properties on various relaxations of the above hybrid automaton model have been shown to be undecidable [48]. Therefore, initialized, rectangular hybrid automata lie on the boundary between decidability and undecidability. However, it has been recognized, mainly in the control community which is used to more sophisticated dynamical models, that the expressive power offered by a rectangular hybrid

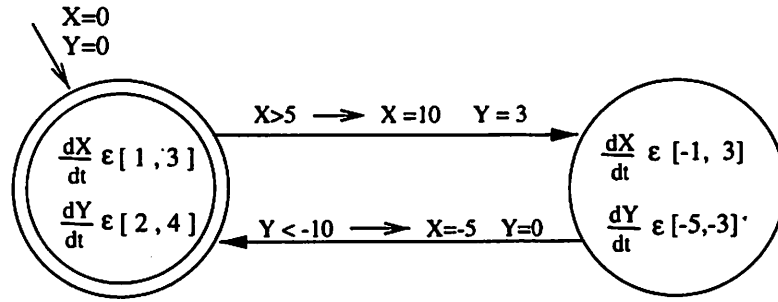


Figure 3.1: Rectangular hybrid automaton

automaton is limited.

In an effort to expand the applicability of the abovementioned decidability results, it is natural to characterize more general hybrid systems that can be transformed to initialized rectangular automata. Such a characterization would be useful as it could capture the modeling frontier of the known decidability frontier. Along this direction, in this chapter, we focus on the following, *continuous* version of this problem.

Problem 3.1 (Straightening Out Rectangular Inclusions). *Under what conditions can a coupled, rectangular differential inclusion of the form,*

$$\begin{aligned} \dot{x}_1 &\in [f_1(x_1, \dots, x_n), g_1(x_1, \dots, x_n)] \\ &\vdots \\ \dot{x}_n &\in [f_n(x_1, \dots, x_n), g_n(x_1, \dots, x_n)] \end{aligned}$$

where $x = [x_1, \dots, x_n]^T \in U \subseteq \mathbb{R}^n$, $f_1, \dots, f_n, g_1, \dots, g_n$ smooth maps from U to \mathbb{R} , and for each $1 \leq i \leq n$ and for all $x \in U$, $g_i(x) > f_i(x)$, be converted by a smooth coordinate change $z = \Phi(x)$ to a decoupled, constant, rectangular inclusion of the form

$$\begin{aligned} \dot{z}_1 &\in [a_1, b_1] \\ &\vdots \\ \dot{z}_n &\in [a_n, b_n] \end{aligned}$$

where a_i, b_i are real constants for all $1 \leq i \leq n$?

It should be noted that solving Problem 3.1 focuses only on the continuous part of transforming a general hybrid automaton to a rectangular automaton. In general, one must

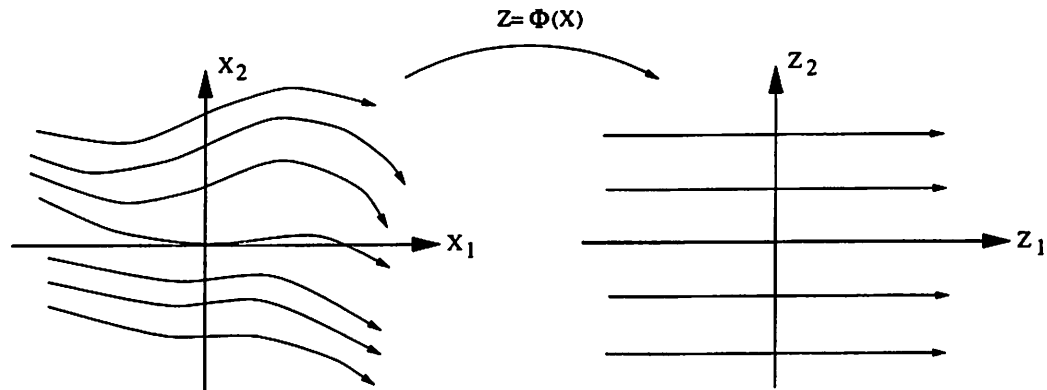


Figure 3.2: Straightening the flow of a vector field

also transform the maps associated with the discrete transitions as well. In addition, the only parameters allowed in the description of a rectangular hybrid automaton are rational numbers, as only rational numbers can be symbolically represented and manipulated by computers. Therefore, in general, one must restrict the class of coordinate changes to ensure that rational numbers are being mapped to rational numbers. Even though Problem 3.1 does not consider these issues, it will be shown that even this relaxed version of the problem gives rise to quite restrictive conditions.

In order to derive necessary and sufficient conditions for the solution of Problem 3.1, two versions of the well known straightening out theorem for differential equations are used. In the next section, these classic results are reviewed, and in Section 3.2 they are used for solving Problem 3.1.

3.1 Straightening Out Differential Equations

Given any vector field on a manifold, then away from equilibria, there exists a local change of coordinates which transforms the flow of the vector field to straight lines.

Theorem 3.2 (Straightening Out Theorem). *Let X be a smooth vector field on manifold M with $X(p) \neq 0$ at some $p \in M$. Then there exists a coordinate chart $(U, z) = (U, z_1, \dots, z_n)$ of p such that on U vector field X is expressed as*

$$X = \frac{\partial}{\partial z_1} \tag{3.1}$$

Therefore given a differential equation of the form

$$\dot{x} = f(x)$$

where $x \in \mathbb{R}^n$, and $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is smooth, then away from equilibria, $f(x) \neq 0$, there exists a local change of coordinates $z = \Phi(x)$ such that in the z coordinates the differential equation is expressed as

$$\dot{z}_1 = 1 \quad \dot{z}_2 = 0 \quad \dots \quad \dot{z}_n = 0 \quad (3.2)$$

An intuitive, planar explanation of this remarkable theorem is shown in Figure 3.2. Assume without loss of generality that x_0 is at the origin of the (x_1, x_2) coordinate system. Integrating vector field f in a neighborhood of x_0 results in foliating the state space by integral curves. Each point x in a neighborhood of x_0 can be then uniquely characterized by the unique leaf of the foliation to which it belongs, and the time it takes for the integral curve to reach the point from the x_2 axis. The derivative of the coordinate which describes the leaf of the foliation is zero since the leaf is invariant under the flow. The derivative of the coordinate which measures time is simply one. Therefore the desired diffeomorphism is simply the time parameterization of the integral curves (z_1) along with the leaves of the resulting foliation (z_2, \dots, z_n) which is induced by integrating the system. Since obtaining the desired diffeomorphism involves explicit integration of the differential equation, the straightening out theorem is a local and non-constructive result. Constructive cases are feasible if the vector field can be integrated analytically. A complete proof of this theorem can be found in most differential geometry books like [1, 96].

In the case where many vector fields must be straightened out by the same change of coordinates, then the following theorem is useful. It can be considered as a generalization of Theorem 3.2 for multiple vector fields.

Theorem 3.3 (Straightening Out Multiple Vector Fields). *Let X_1, \dots, X_k be k smooth, linearly independent vector fields in a neighborhood of $p \in M$ satisfying*

$$[X_i, X_j] = 0 \quad 1 \leq i, j \leq k \quad (3.3)$$

Then there exists a coordinate chart $(U, z) = (U, z_1, \dots, z_n)$ such that on U we have for $1 \leq i \leq k$

$$X_i = \frac{\partial}{\partial z_i} \quad (3.4)$$

Therefore given n differential equations of the form

$$\dot{x} = f_i(x)$$

where $1 \leq i \leq n$, $x \in \mathbb{R}^n$ and $f_i : \mathbb{R}^n \rightarrow \mathbb{R}^n$ are smooth, then at any $x_0 \in \mathbb{R}^n$ where the vectors $\{f_i(x_0)\}_{i=1}^n$ is a linearly independent set¹, and the Lie bracket conditions hold, there exists a local change of coordinates $z = \Phi(x)$ such that in the z coordinates the i -th differential equation is expressed as

$$\dot{z}_1 = 0 \quad \dots \quad \dot{z}_i = 1 \quad \dots \quad \dot{z}_n = 0 \quad (3.5)$$

Like the Flow Box Theorem, Theorem 3.3 is also local and non-constructive. The Lie bracket condition, which simply says that the flows of the vector fields commute, is necessary in order for the change of coordinates to be well defined. More important though, in the new coordinates, the vector fields in addition to being straightened out are also decoupled.

3.2 Straightening Out Differential Inclusions

A differential inclusion on \mathbb{R}^n is defined as

$$\dot{x} \in F(x) \quad (3.6)$$

where F is a map which at each $x \in \mathbb{R}^n$ assigns a subset of $T_x \mathbb{R}^n$. From now on, we focus on *rectangular* differential inclusions of the form

$$\begin{aligned} \dot{x}_1 &\in [f_1(x_1, \dots, x_n), g_1(x_1, \dots, x_n)] \\ &\vdots \\ \dot{x}_n &\in [f_n(x_1, \dots, x_n), g_n(x_1, \dots, x_n)] \end{aligned} \quad (3.7)$$

where the derivative of each coordinate lies in an interval. A more convenient representation of rectangular inclusion (3.7) is given by the following expression

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \end{bmatrix} \in F(x) = F_1(x) + F_2(x) + \dots + F_n(x) \quad (3.8)$$

¹Note that linear independence at x_0 requires that x_0 is not an equilibrium of any of the n vector fields. By smoothness, the linear independence condition extends to a neighborhood of x_0

with

$$F_i(x) = \text{co} \left\{ \begin{bmatrix} 0 \\ \vdots \\ f_i(x) \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \vdots \\ g_i(x) \\ \vdots \\ 0 \end{bmatrix} \right\} = \text{co} \{f_i(x)e_i, g_i(x)e_i\} \quad (3.9)$$

where $\text{co}\{p_1, p_2\}$ stands for the convex hull of vectors p_1 and p_2 , and e_1, \dots, e_n is the standard orthonormal basis for \mathbb{R}^n .

Given a smooth change of coordinates $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and differential inclusion (3.6), we can naturally push forward the differential inclusion by pointwise assigning to each $z = \Phi(x)$, the push forward of all tangent vectors belonging in $F(x)$. Thus

$$\dot{z} \in \Phi_*(F(x)) \quad (3.10)$$

is the differential inclusion resulting from the change of coordinates. We can now proceed to the main theorem.

Theorem 3.4 (Straightening Coupled Rectangular Inclusions). *Consider the coupled, rectangular differential inclusion in \mathbb{R}^n ,*

$$\begin{aligned} \dot{x}_1 &\in [f_1(x_1, \dots, x_n), g_1(x_1, \dots, x_n)] \\ &\vdots \\ \dot{x}_n &\in [f_n(x_1, \dots, x_n), g_n(x_1, \dots, x_n)] \end{aligned} \quad (3.11)$$

where $x = [x_1, \dots, x_n]^T \in U \subseteq \mathbb{R}^n$, $f_1, \dots, f_n, g_1, \dots, g_n$ are smooth maps from U to \mathbb{R} , and for each i and for all $x \in U$ we have $g_i(x) > f_i(x)$. Then there exists a local change of coordinates $z = \Phi(x)$ on U such that in the new coordinates the differential inclusion is expressed as

$$\begin{aligned} \dot{z}_1 &\in [a_1, b_1] \\ &\vdots \\ \dot{z}_n &\in [a_n, b_n] \end{aligned} \quad (3.12)$$

if and only if for all $x \in U$ and for all $1 \leq i, j \leq n$,

$$[f_i(x)e_i, g_j(x)e_j] = 0 \quad (3.13)$$

$$\left[f_i(x)e_i, f_j(x)e_j \right] = 0 \quad (3.14)$$

and for all $1 \leq i \leq n$ and for all $x \in U$ there exist $k_i \in \mathbb{R}$, such that either

$$g_i(x) = k_i f_i(x) \text{ or } f_i(x) = k_i g_i(x) \quad (3.15)$$

Proof. Before we begin with the proof, we remark that conditions (3.13,3.14,3.15) contain some redundancy. However, a minimal set of conditions would be notationally complicated.

(Necessity) Consider rectangular inclusion (3.7) along with it's useful representation (3.8,3.9). Note that for $i \neq j$, any vector in $F_i(x)$ is linearly independent from any vector in $F_j(x)$. Performing the change of coordinates $z = \Phi(x)$ results in

$$\begin{aligned} \dot{z} &\in \Phi_*(F(x)) \\ &= \Phi_*(F_1(x) + F_2(x) + \cdots + F_n(x)) \end{aligned} \quad (3.16)$$

By the linearity of Φ_* we have that

$$\dot{z} \in \Phi_*(F_1(x)) + \Phi_*(F_2(x)) + \cdots + \Phi_*(F_n(x)) \quad (3.17)$$

Since Φ_* is pointwise an isomorphism, we retain the property that any vector from $\Phi_*(F_i(x))$ is linearly independent from any vector in $\Phi_*(F_j(x))$ for $i \neq j$.

Now, by assumption, the change of coordinates results in inclusion (3.12) which is also expressed as

$$\dot{z} = \begin{bmatrix} \dot{z}_1 \\ \vdots \\ \dot{z}_n \end{bmatrix} \in Z = Z_1 + Z_2 + \cdots + Z_n \quad (3.18)$$

where Z_i is the constant set

$$Z_i = \text{co} \left\{ \begin{bmatrix} 0 \\ \vdots \\ a_i \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \vdots \\ b_i \\ \vdots \\ 0 \end{bmatrix} \right\} = \text{co} \{ a_i e_i, b_i e_i \} \quad (3.19)$$

Note again that for $i \neq j$, any vector from Z_i is linearly independent from any vector in Z_j .

By assumption we then have that

$$\dot{z} \in \Phi_*(F_1(x)) + \Phi_*(F_2(x)) + \cdots + \Phi_*(F_n(x)) = Z_1 + Z_2 + \cdots + Z_n \quad (3.20)$$

Since for all $i \neq j$, vectors in $\Phi_*(F_i(x))$ (also Z_i) are linearly independent from vectors in $\Phi_*(F_j(x))$ (respectively Z_j) then (3.20) requires that for each i there exists some j_i such that $\Phi_*(F_i(x)) = Z_{j_i}$. Therefore, up to a permutation of the indices, the sets $\Phi_*(F_i(x))$ are equal to the sets Z_i .

In general, for linear map A and vectors p_1, p_2 the following property of convex hulls

$$\text{Aco}\{p_1, p_2\} = \text{co}\{Ap_1, Ap_2\} \quad (3.21)$$

can be easily checked. By applying this property on (3.17,3.9) we obtain that

$$\begin{aligned} \Phi_*(F_i(x)) &= \Phi_*(\text{co}\{f_i(x)e_i, g_i(x)e_i\}) \\ &= \text{co}\{\Phi_*(f_i(x)e_i), \Phi_*(g_i(x)e_i)\} \end{aligned} \quad (3.22)$$

The above calculations essentially show that in order to push forward a rectangular differential inclusion, one only needs to push forward the finite number of vector fields that are needed to define the rectangular set of tangent vectors.

But since $\Phi_*(F_i(x)) = Z_{j_i}$, condition (3.22) results in

$$\text{co}\{\Phi_*(f_i(x)e_i), \Phi_*(g_i(x)e_i)\} = \text{co}\{a_{j_i}e_{j_i}, b_{j_i}e_{j_i}\} \quad (3.23)$$

which means that either

$$\Phi_*(f_i(x)e_i) = a_{j_i}e_{j_i} \text{ and } \Phi_*(g_i(x)e_i) = b_{j_i}e_{j_i} \quad (3.24)$$

or

$$\Phi_*(f_i(x)e_i) = b_{j_i}e_{j_i} \text{ and } \Phi_*(g_i(x)e_i) = a_{j_i}e_{j_i} \quad (3.25)$$

Assume without loss of generality that the first case holds (equations (3.24)). Then for all $0 \leq i, l \leq n$,

$$\begin{aligned} \Phi_*\left([f_i(x)e_i, g_l(x)e_l]\right) &= [\Phi_*(f_i(x)e_i), \Phi_*(g_l(x)e_l)] \\ &= [a_{j_i}e_{j_i}, b_{j_l}e_{j_l}] = 0 \end{aligned} \quad (3.26)$$

which results in the necessary conditions

$$[f_i(x)e_i, g_l(x)e_l] = 0 \quad \text{for all } 0 \leq i, l \leq n \quad (3.27)$$

since Φ_* is pointwise an isomorphism. In a similar manner one obtains

$$\left[f_i(x)e_i, f_l(x)e_l \right] = 0 \quad \text{for all } 0 \leq i, l \leq n \quad (3.28)$$

In addition, since $g_i(x) > f_i(x)$, if $f_i(x) \neq 0$ we can express $g_i(x)$ as a nonlinear function of $f_i(x)$ by $g_i(x) = k_i(x)f_i(x)$ (if $f_i(x) = 0$ then express $f_i(x)$ as $g_j(x)$ multiplied by zero and proceed in the same way). Then

$$\begin{aligned} b_{j_i}e_{j_i} &= \Phi_*(g_i(x)e_i) = \Phi_*(k_i(x)f_i(x)e_i) \\ &= k_i(x)\Phi_*(f_i(x)e_i) = k_i(x)a_{j_i}e_{j_i} \end{aligned} \quad (3.29)$$

must hold for all $x \in U$. Thus $k_i(x)$ must be constant and $g_i(x)$ must be a constant multiple of $f_i(x)$ for all $x \in U$. Note that for each i either $f_i(x)$ or $g_i(x)$ can be zero (but not both since $g_i(x) > f_i(x)$). However, if $f_i(x)$ or $g_i(x)$ is zero at some point x_0 , say $g_i(x_0) = 0$ and $f_i(x_0) \neq 0$, then smoothness and the fact that $g_i(x)$ must be a constant multiple of $f_i(x)$ for all $x \in U$, force $g_i(x)$ to be identically zero on U .

(Sufficiency) Consider conditions (3.13,3.14,3.15) and assume without loss of generality that for all i , $f_i(x) \neq 0$. (if $f_{i_0} = 0$ for some i_0 , then pick g_{i_0} which must be nonzero and proceed in a similar way). Then, the set of vector fields

$$\{f_i(x)e_i\}_{i=1}^n \quad (3.30)$$

satisfies the conditions of Theorem 3.3. Thus, there exists a diffeomorphism $z = \Phi(x)$ such that

$$\Phi_*(f_i(x)e_i) = e_i \quad (3.31)$$

Now pushing forward the rectangular inclusion

$$\dot{x} \in F_1(x) + F_2(x) + \cdots + F_n(x) \quad (3.32)$$

by Φ_* results in

$$\begin{aligned} \dot{z} &\in \Phi_*(F(x)) \\ &= \Phi_*(F_1(x) + F_2(x) + \cdots + F_n(x)) \\ &= \Phi_*(F_1(x)) + \Phi_*(F_2(x)) + \cdots + \Phi_*(F_n(x)) \end{aligned} \quad (3.33)$$

But since for each i and for all x we have $g_i(x) = k_i f_i(x)$ for some constant k_i (positive, negative or zero), we obtain

$$\begin{aligned}\Phi_*(F_i(x)) &= \Phi_*(\text{co}\{f_i(x)e_i, k_i f_i(x)e_i\}) \\ &= \text{co}\{\Phi_*(f_i(x)e_i), \Phi_*(k_i f_i(x)e_i)\} \\ &= \text{co}\{e_i, k_i e_i\}\end{aligned}\tag{3.34}$$

and thus in the z coordinates we obtain the inclusion

$$\begin{aligned}z_1 &\in [1, k_1] \\ &\vdots \\ z_n &\in [1, k_n]\end{aligned}$$

Note that some of the k_i may be zero or even negative in which case the corresponding intervals must be flipped. This completes the proof. \square

Note that the necessary part of the proof of Theorem 3.4, depends on the fact that $g_i(x)$ is strictly greater than $f_i(x)$ for all i . Therefore Theorem 3.4 is not a generalization of the straightening out theorem for differential equations. Even though straightening out a differential equation is always possible away from an equilibrium, straightening out a rectangular differential inclusion, requires straightening out many vector fields, while using the same change of coordinates. This places restrictions on the types of rectangular differential inclusions that can be straightened out. The following example shows how restrictive this class is.

Example 9. Consider the coupled differential inclusion

$$\begin{aligned}\dot{x}_1 &\in [f_1(x_1, x_2), g_1(x_1, x_2)] \\ \dot{x}_2 &\in [f_2(x_1, x_2), g_2(x_1, x_2)]\end{aligned}$$

where we have $f_1(x_1, x_2) \neq 0$ and $f_2(x_1, x_2) \neq 0$ on some set $U \subseteq \mathbb{R}^2$. Then conditions (3.15) require that $g_i(x_1, x_2)$ is a constant multiple of $f_i(x_1, x_2)$. Thus necessary conditions (3.13,3.14) reduce to simply checking whether

$$[f_1(x_1, x_2)e_1, f_2(x_1, x_2)e_2] = 0\tag{3.35}$$

as all other Lie brackets are guaranteed to be zero if the above one is. But

$$[f_1(x_1, x_2)e_1, f_2(x_1, x_2)e_2] = 0 \implies \begin{bmatrix} \frac{\partial f_1}{\partial x_2} f_2 \\ \frac{\partial f_2}{\partial x_1} f_1 \end{bmatrix} = 0 \quad (3.36)$$

But since $f_1 \neq 0$ and $f_2 \neq 0$ on U , this requires

$$\frac{\partial f_1}{\partial x_2} = 0 \quad \frac{\partial f_2}{\partial x_1} = 0 \quad (3.37)$$

which means that it is necessary for the rectangular inclusion to be already decoupled!

The above example suggests that the conditions of Theorem 3.4 are quite restrictive. In the case that $f_i(x)$ and $g_i(x)$ depend on x_i alone, the Lie bracket conditions (3.13,3.14) are trivially satisfied. As a corollary of Theorem 3.4, we obtain the following straightening out theorem for decoupled, rectangular inclusions.

Corollary 3.5 (Straightening Out Decoupled Inclusions). *Consider the scalar differential inclusion*

$$\dot{x} \in [f(x), g(x)] \quad (3.38)$$

with $x \in U \subseteq \mathbb{R}$, $f, g : U \rightarrow \mathbb{R}$ smooth, and assume that for all $x \in U$ we have $g(x) > f(x)$. Then there exists a local change of coordinates $z = \Phi(x)$ such that in the new coordinates the differential inclusion is expressed as

$$\dot{z} \in [a, b] \quad (3.39)$$

if and only if for all $x \in U$ either $g(x)$ is a constant multiple of $f(x) \neq 0$ or $f(x)$ is a constant multiple of $g(x) \neq 0$.

As a corollary of Corollary 3.5 we obtain

Corollary 3.6. *The following scalar inclusions can be locally transformed to constant rectangular differential inclusions:*

- *Linear Differential Inclusions : $\dot{x} \in [a, b]x$, $x \neq 0$*
- *Nonlinear Differential Inclusions : $\dot{x} \in [0, f(x)]$, $f(x) > 0$*
- *Nonlinear Differential Inclusions : $\dot{x} \in [f(x), 0]$, $f(x) < 0$*

- *Nonlinear Differential Inclusions* : $\dot{x} \in [a, b]f(x)$, $f(x) \neq 0$

Corollaries 3.5 and 3.6 show that scalar rectangular differential inclusions cannot be straightened out unless one boundary vector field, $g(x)$, is a constant multiple of the other, $f(x)$. This result is intuitively clear. By Theorem 3.2, any vector field, say $f(x)$, can be straightened out away from singularities. But if the same diffeomorphism must also straighten the flows of the other vector field, $g(x)$, then $g(x)$ must be a constant multiple of $f(x)$. But if $g(x)$ is a constant multiple of $f(x)$, then after factorization, we obtain that a differential inclusion of the form $\dot{x} \in [a, b]f(x)$ is the limiting case of an inclusion which can be straightened out.

Example 10. Consider the following simple differential inclusion

$$\dot{x} \in [3, 5]x$$

on $U = \{x \in \mathbb{R} \mid x > 0\}$. Then $z = \ln x$ satisfies

$$\dot{z} = \frac{\partial \ln x}{\partial x} \dot{x} \in \frac{1}{x}[3, 5]x = [3, 5]$$

and the inclusion is straightened out on U .

3.3 Conclusions

The goal of this chapter was to potentially expand the applicability of the known decidability results, for computationally verifying properties of hybrid systems. However, given the restrictive nature of the necessary conditions, Theorem 3.4 presents a serious modeling barrier in the battle against decidability. This leaves control theorists unsatisfied as the modeling power of decidable hybrid systems does not capture meaningful continuous dynamics.

The next two chapters present an effort to computationally analyze hybrid systems with more complicated dynamics. Even though the undecidability results in [48] in conjunction with the results of this chapter, restrict hybrid systems to very simple continuous dynamics, we shall escape this undecidability frontier by restricting the type of discrete transitions allowed in our model. This will give us a lot of room to maneuver on the continuous side, and will allow us to capture classes of *linear vector fields* in each discrete location.

Chapter 4

Computable Hybrid Systems

Verification algorithms perform reachability computations and check whether trajectories of the hybrid system can reach certain undesirable regions of the state space. When such computational algorithms are applied to systems with infinite state spaces, they are in danger of never terminating. This makes the issue of *decidability*, which guarantees termination of the algorithm, a very important one.

The main tool for obtaining classes of hybrid system for which the reachability problem is decidable, is given by the concept of *bisimulation* [79]. Bisimulations are simply reachability preserving quotient systems. If an infinite state hybrid system has a finite state bisimulation, then checking reachability for the hybrid system can be *equivalently* performed on the finite, discrete, quotient graph. Since the quotient graph is finite, the algorithm will terminate. If in addition, each step of the algorithm can be encoded and implemented by a computer program, then the problem is decidable. Therefore, in order to obtain classes of hybrid systems with a decidable reachability problem, we must answer the following two questions:

- **Step 1** : When does a hybrid system admit a *finite* bisimulation?
- **Step 2** : If a finite bisimulation exists, can we construct it?

Up to now, answering the above two questions has been done simultaneously by explicitly constructing a partition which is checked to be a bisimulation. This approach has resulted in timed automata [3], multirate automata [2], and initialized rectangular automata [48, 91].

In this chapter we shall deal with the above questions separately. In particular, we first answer the question regarding the existence of finite bisimulations. To answer the finiteness question, we need to identify classes of sets and flows with globally, finite intersection properties. This is provided by the concept of *o-minimal theories* in mathematical logic [106]. Using this concept, we introduce the notion of *o-minimal hybrid systems*, and prove that o-minimal hybrid systems always admit finite bisimulations. We then list various o-minimal hybrid systems which capture versions of most hybrid systems known to admit finite bisimulations. Moreover, we present hybrid systems with much more complex dynamics which are definable in recently discovered o-minimal theories and thus also admit finite bisimulations.

In order to construct bisimulations, we need to symbolically represent, and manipulate sets. The main computational tool for symbolic set manipulation in this context is *quantifier elimination*. Since quantifier elimination is possible for the theory of reals with addition and multiplication [98], we either find or transform subclasses of o-minimal hybrid systems which are definable in this theory. This immediately leads to an extension of the decidability frontier that captures classes of hybrid systems with linear vector fields in each discrete location. The importance of this result is immediately clear given the wide applicability of linear systems in control theory.

In order to get to this desired goal, in the next section we review the well known notion of bisimulation of transition systems.

4.1 Bisimulations of Transition Systems

Transition systems should be thought of as abstract graph models, which do not necessarily consist of a finite number of states. In fact, transition systems are abstract enough to include both finite state machines and differential equations can be thought of as transition systems.

Definition 4.1 (Transition Systems). *A transition system $T = (Q, \Sigma, \rightarrow, Q_O, Q_F)$ consists of:*

- *A set Q of states*
- *An alphabet Σ of events,*

- A transition relation $\rightarrow \subseteq Q \times \Sigma \times Q$,
- A set $Q_O \subseteq Q$ of initial states,
- A set $Q_F \subseteq Q$ of final states.

It is customary to denote a transition $(q_1, \sigma, q_2) \in \rightarrow$ as $q_1 \xrightarrow{\sigma} q_2$. The transition system is *finite* if the cardinality of Q is finite, and it is infinite otherwise. The transition system T is *deadlock free*, if for any state $q \in Q$, there exists a state $q' \in Q$ and an event $\sigma \in \Sigma$ such that $q \xrightarrow{\sigma} q'$.

Example 11. To see the generality of transition systems, consider the simple differential equation $\dot{x} = f(x)$ where $x \in \mathbb{R}^n$. The differential equation defines a transition system whose state space is \mathbb{R}^n , has event alphabet $t \in \Sigma = \mathbb{R}$, and the transition relation $x_1 \xrightarrow{t} x_2$ means that the solution of the differential equation from x_1 reaches x_2 in time t .

A region is a subset $P \subseteq Q$. Given $\sigma \in \Sigma$ we define two regions, $Pre_\sigma(P)$ and $Post_\sigma(P)$ of a region P as

$$Pre_\sigma(P) = \{q \in Q \mid \exists p \in P \text{ and } q \xrightarrow{\sigma} p\} \quad (4.1)$$

$$Post_\sigma(P) = \{q \in Q \mid \exists p \in P \text{ and } p \xrightarrow{\sigma} q\} \quad (4.2)$$

Thus $Pre_\sigma(P)$ is the set of states that can reach P with a single σ event. Similarly, $Post_\sigma(P)$ is the set of states that can be reached from states in P with a single σ event. The set of states that can reach P , or can be reached by P in one step for all σ events is

$$Pre(P) = \{q \in Q \mid \exists \sigma \in \Sigma \exists p \in P \text{ and } q \xrightarrow{\sigma} p\} \quad (4.3)$$

$$Post(P) = \{q \in Q \mid \exists \sigma \in \Sigma \exists p \in P \text{ and } p \xrightarrow{\sigma} q\} \quad (4.4)$$

The set of states that are reachable from P in two steps is simply $Post(Post(P))$ and is denoted $Post^2(P)$. In general, $Post^i(P)$ consists of states that are reachable from P in i steps. Similar definitions hold for $Pre^i(P)$. Then

$$Pre^*(P) = \bigcup_{i \in \mathbb{N}} Pre^i(P) \quad (4.5)$$

$$Post^*(P) = \bigcup_{i \in \mathbb{N}} Post^i(P) \quad (4.6)$$

are simply the set of states that *backward* and *forward* reachable from P . A problem that is of great interest for transition systems is the reachability problem.

Problem 4.2 (Reachability Problem). *Given a transition system T , is a state $q_f \in Q_F$ reachable from a state $q_o \in Q_O$ by a sequence of transitions?*

In other words, we want to check whether $Post^*(Q_O) \cap Q_F \neq \emptyset$ or, similarly, whether $Pre^*(Q_F) \cap Q_O \neq \emptyset$. The reachability problem is also referred to as the *safety verification* problem. The set of final states encode an undesirable or unsafe region of the state space. The reachability problem is tackled using either of the following reachability algorithms.

Forward Reachability Algorithm

```

set  $R := Q_O$ 

while true do
    if  $R \cap Q_F \neq \emptyset$  return unsafe ; stop
    if  $Post(R) \subseteq R$  return safe ; stop
    else  $R := R \cup Post(R)$ 

end while

```

Backward Reachability Algorithm

```

set  $R := Q_F$ 

while true do
    if  $R \cap Q_O \neq \emptyset$  return unsafe ; stop
    if  $Pre(R) \subseteq R$  return safe ; stop
    else  $R := R \cup Pre(R)$ 

end while

```

If the state space of the transition system is finite, then both algorithms are guaranteed to terminate, since in the worst case, both algorithms can only add a finite number of states. If the state space is infinite, then there is, in general, no guarantee that the above reachability computations will terminate after a finite number of steps. In fact, it may be the case that the forward reachability algorithm terminates and the backward reachability

algorithm does not, and vice versa. One must therefore use both algorithms for transition systems for which we have no termination guarantees. Our goal, however, is to find classes of infinite state transition systems for which we can compute the reachable space in a finite number of steps. This is accomplished by reducing the infinite state transition system to a *finite* state quotient system with *equivalent* reachability properties. In order to achieve this, the notion of a quotient transition system needs to be defined.

Given an equivalence relation $\sim \subseteq Q \times Q$ on the state space, the definition of quotient transition system T/\sim is natural. Let Q/\sim denote the quotient space. For a region P , we denote by P/\sim the collection of all equivalence classes which intersect P . Given an equivalence relation \sim on Q , we call a set a \sim -block if it is a union of equivalence classes. The transition relation \rightarrow_{\sim} on the quotient space is defined as follows: for $Q_1, Q_2 \in Q/\sim$, $Q_1 \xrightarrow{\sigma}_{\sim} Q_2$ iff there exist $q_1 \in Q_1$ and $q_2 \in Q_2$ such that $q_1 \xrightarrow{\sigma} q_2$. The quotient transition system is then $T/\sim = (Q/\sim, \Sigma, \rightarrow_{\sim}, Q_0/\sim, Q_F/\sim)$.

Definition 4.3 (Bisimulation). *Let $T = (Q, \Sigma, \rightarrow, Q_O, Q_F)$ be a transition system. The equivalence relation \sim is a bisimulation of T iff*

- Q_O, Q_F are \sim -blocks
- For all $\sigma \in \Sigma$ and all \sim -blocks P , $Pre_{\sigma}(P)$ is a \sim -block.

If \sim induces a bisimulation, then transition systems T and T/\sim are called bisimilar. A bisimulation is called finite if the quotient space is finite.

Therefore, the crucial property of bisimulations is that the intersection of $Pre_{\sigma}(P)$ and Q , for equivalence classes P, Q , and $\sigma \in \Sigma$, is either the empty set or all of Q . Alternatively, if \sim is a bisimulation, it can be easily shown that if $p \sim q$ then

1. $p \in Q_F$ iff $q \in Q_F$, and $p \in Q_O$ iff $q \in Q_O$
2. if $p \xrightarrow{\sigma} p'$ then there exists q' such that $q \xrightarrow{\sigma} q'$ and $p' \sim q'$

The above characterization of bisimulations leads to the following theorem.

Theorem 4.4 (Reachability Equivalence). *Let $T = (Q, \Sigma, \rightarrow, Q_O, Q_F)$ and $T/\sim = (Q/\sim, \Sigma, \rightarrow_{\sim}, Q_0/\sim, Q_F/\sim)$ be bisimilar transition systems. Then the reachability problems for T and T/\sim are equivalent.*

Proof. Let $p_0 \xrightarrow{\sigma_1} p_1 \xrightarrow{\sigma_2} \dots \xrightarrow{\sigma_n} p_n$ be a sequence of transitions of T with $p_0 \in Q_O$ and $p_n \in Q_F$. Let $h : Q \rightarrow Q/\sim$ be the natural map that takes each state to the equivalence class it belongs. Then by the definition of T/\sim , $h(p_0) \xrightarrow{\sigma_1} h(p_1) \xrightarrow{\sigma_2} \dots \xrightarrow{\sigma_n} h(p_n)$ is a sequence of transitions of T/\sim from $h(p_0) \in Q_O/\sim$ to $h(p_n) \in Q_F/\sim$.

Conversely, let $P_0 \xrightarrow{\sigma_1} P_1 \xrightarrow{\sigma_2} \dots \xrightarrow{\sigma_n} P_n$ be a sequence of transitions of T/\sim where P_0, P_1, \dots, P_n are equivalence classes, and $P_0 \in Q_O/\sim$, $P_n \in Q_F/\sim$. Since $P_0 \xrightarrow{\sigma_1} P_1$, then by the definition of T/\sim , there exist $p_0 \in P_0$ and $p_1 \in P_1$ such that $p_0 \xrightarrow{\sigma_1} p_1$. Similarly, since $P_1 \xrightarrow{\sigma_2} P_2$, then by the definition of T/\sim , there exists $r_1 \in P_1$ and $r_2 \in P_2$ such that $r_1 \xrightarrow{\sigma_2} r_2$. But since $p_1 \sim r_1$, $r_1 \xrightarrow{\sigma_2} r_2$ and \sim is a bisimulation, then there exists $p_2 \sim r_2$ such that $p_1 \xrightarrow{\sigma_2} p_2$. Therefore, $p_0 \xrightarrow{\sigma_1} p_1 \xrightarrow{\sigma_2} p_2$. By continuing the same process, there exist $p_0 \in P_0, p_1 \in P_1, \dots, p_n \in P_n$ such that $p_0 \xrightarrow{\sigma_1} p_1 \xrightarrow{\sigma_2} \dots \xrightarrow{\sigma_n} p_n$, and $p_0 \in Q_O, p_n \in Q_F$. \square

Therefore, checking reachability properties on the bisimilar transition system T/\sim is equivalent to checking properties of the original transition system T . This has two immediate applications. If T has a finite state space, then reachability algorithm are always guaranteed of terminating. For finite transition systems, bisimulations are very useful in reducing the complexity of reachability verification algorithms where the state space Q is finite but very large.

If, however, T has an infinite state space, then reachability algorithms are not guaranteed to terminate. For such systems, if we can find a bisimilar but finite transition system T/\sim , then checking reachability of T can be equivalently done on T/\sim for which the reachability algorithm is guaranteed to terminate. Therefore, bisimulations are the main tool for obtaining classes of hybrid systems with a *decidable* reachability problem. Note that even though for finite transition systems, a finite bisimulation always exists (equality), this is not the case for infinite transition systems. This philosophy has successfully resulted in various decidable classes of hybrid systems, like timed automata [3], initialized rectangular automata [91], and linear hybrid automata [45].

A conceptual algorithm that computes a bisimilarity partition of the state space starts with a given transition system T , and computes increasingly finer partitions of the state space Q . If the algorithm terminates, then the resulting quotient transition system is a finite bisimulation.

Bisimulation Algorithm for Transition Systems

```

set  $Q/\sim = \{Q_O, Q_F, Q \setminus (Q_O \cup Q_F)\}$ 
while  $\exists P, P' \in Q/\sim$  and  $\sigma \in \Sigma$  such that  $\emptyset \neq P \cap Pre_\sigma(P') \neq P$  do
    set  $P_1 = P \cap Pre_\sigma(P'), P_2 = P \setminus Pre_\sigma(P')$ 
    refine  $Q/\sim = (Q/\sim \setminus \{P\}) \cup \{P_1, P_2\}$ 
end while

```

Therefore, given an infinite transition system T , the bisimulation algorithm results, if it terminates, in a finite, bisimilar transition system T/\sim . We can then apply either the forward or backward reachability algorithm on T/\sim , which are guaranteed to terminate. It is easy to show however, that if T has a finite bisimulation, then either forward or backward reachability algorithms on T will terminate. Therefore, instead of constructing the bisimulation partition, we can simply compute the reachable set of the original transition system. Therefore, the above bisimulation algorithm is used mainly as a theoretical device for obtaining classes of transition systems with a decidable reachability problem. The actual reachability calculation, in practice, is usually performed with the backward or forward reachability algorithms.

In addition to showing that the bisimulation algorithm terminates, decidability requires that each step of the algorithm is *computable*. This means that we must be able to represent sets symbolically, perform boolean operations, check emptiness of a set, and compute $Pre_\sigma(P)$ for any $\sigma \in \Sigma$. In the next sections, the above ideas will be applied for a class of transition systems generated by hybrid systems.

4.2 Bisimulations of Hybrid Systems

We focus on transition systems generated by the following class of hybrid systems. Even though the following model is rather abstract, we shall eventually introduce enough structure on the model in order to make it amenable to analysis.

Definition 4.5 (Hybrid System). *A hybrid system $H = (X, X_0, X_F, F, E, I, G, R)$ consists of*

- $X = X_D \times X_C$ is the state space with $X_D = \{q_1, \dots, q_n\}$ and X_C a manifold.
- $X_0 \subseteq X$ is the set of initial states

- $X_F \subseteq X$ is the set of final states
- $F : X \rightarrow TX_C$ assigns to each discrete location $q \in X_D$ a vector field $F(q, \cdot)$
- $E \subseteq X_D \times X_D$ is the set of discrete transitions
- $I : X_D \rightarrow 2^{X_C}$ assigns to each location a set $I(q) \subseteq X_C$ called the invariant.
- $G : E \rightarrow X_D \times 2^{X_C}$ assigns to $e = (q_1, q_2) \in E$ a guard of the form $\{q_1\} \times U$, $U \subseteq I(q_1)$.
- $R : E \rightarrow X_D \times 2^{X_C}$ assigns to $e = (q_1, q_2) \in E$ a reset of the form $\{q_2\} \times V$, $V \subseteq I(q_2)$.

Trajectories of the hybrid system H originate at any $(q, x) \in X_0$ and consist of either continuous evolutions or discrete jumps. Continuous trajectories keep the discrete part of the state constant, and the continuous part evolves according to the continuous flow $F(q, \cdot)$ as long as the flow remains inside the invariant set $I(q)$. If the flow exits $I(q)$, then a discrete transition is *forced*. If, during the continuous evolution, a state $(q, x) \in G(e)$ is reached for some $e \in E$, then discrete transition e is *enabled*. The hybrid system may then instantaneously jump from (q, x) to any $(q', x') \in R(e)$ and the system then evolves according to the flow $F(q', \cdot)$. Notice that even though the continuous evolution is deterministic, the discrete evolution may be nondeterministic. Finally, we assume that our hybrid system model is *non-blocking*, that is from every state either a continuous evolution or a discrete transition is possible.

Example 12. A typical hybrid system is shown in Figure 4.1. The state space is $\{q_1, q_2\} \times \mathbb{R}^2$. The initial states are of the form $\{q_1\} \times \{(x, y) \in \mathbb{R}^2 \mid 0 < x < 1, 1 < y < 2\}$. The discrete dynamics consists of two transitions $e_1 = (q_1, q_2)$ and $e_2 = (q_2, q_1)$. Within location q_1 , the continuous variables x and y evolve according to a differential equation as long as $(x, y) \in I(q_1) = \{(x, y) \in \mathbb{R}^2 \mid x \leq 5\}$. Once $x > 5$, discrete transition e_1 is forced and x, y are nondeterministically reset to values in fixed sets. The system then flows according to the flow associated with q_2 . The evolution from that point on is similar. A typical reachability problem asks whether the system will reach the set of final states $\{q_2\} \times \{(x, y) \in \mathbb{R}^2 \mid x < -5\}$.

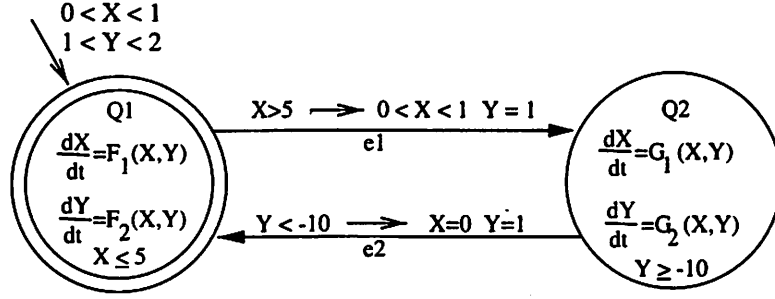


Figure 4.1: A typical hybrid automaton

Every hybrid system $H = (X, X_0, X_F, F, E, I, G, R)$ generates a transition system $T = (Q, \Sigma, \rightarrow, Q_0, Q_F)$ by setting $Q = X$, $Q_0 = X_0$, $Q_F = X_F$, $\Sigma = E \cup \{\tau\}$, and $\rightarrow = (\cup_{e \in E} \xrightarrow{e}) \cup \xrightarrow{\tau}$ where

Discrete Transitions $(q, x) \xrightarrow{e} (q', x')$ for $e \in E$ iff $(q, x) \in G(e)$ and $(q', x') \in R(e)$

Continuous Transitions $(q_1, x_1) \xrightarrow{\tau} (q_2, x_2)$ iff $q_1 = q_2$ and there exists $\delta \geq 0$ and a curve $x : [0, \delta] \rightarrow M$ with $x(0) = x_1$, $x(\delta) = x_2$ and for all $t \in [0, \delta]$ it satisfies $x' = F(q_1, x(t))$ and $x(t) \in I(q_1)$.

The continuous τ transitions are time-abstract transitions, in the sense that the time it takes to reach one state from another is ignored. Having defined the continuous and discrete transitions $\xrightarrow{\tau}$ and \xrightarrow{e} allows us to formally define $Pre_\tau(P)$ and $Pre_e(P)$ for $e \in E$ and any region $P \subseteq X$ using (4.1).

Note that the discrete transitions allowed in our model are memoryless, constant, but possibly set valued. Every time a discrete transition is taken, the whole state *must* be reinitialized. The state is not allowed, for example, to remain the same after the discrete transition is taken. In other words, identity (or other nonconstant) maps are not allowed as reset maps. Because of this restriction, Definition 4.5 does not capture all possible discrete dynamics allowed in timed and rectangular automata [2, 3, 91]. In general, in rectangular automata, the continuous dynamics are decoupled and each component of the continuous part of the state may be either reset nondeterministically to an interval or remain the same. If, however, the dynamics of a particular component changes then the reset map cannot be the identity map on that component. As will be shown, restricting the reset maps will allow us to capture much more complex and coupled dynamics than [2, 3, 91] without violating

the undecidability results of [48].

To show that the reachability problem for a class of hybrid systems defined in Definition 4.5 is decidable, we must show that the bisimulation algorithm will terminate after a finite number of steps. The memoryless structure of the discrete transitions allowed in our hybrid system model results in

$$Pre_e(P) = \begin{cases} \emptyset & \text{if } P \cap R(e) = \emptyset \\ G(e) & \text{if } P \cap R(e) \neq \emptyset \end{cases} \quad (4.7)$$

for all discrete transitions $e \in E$ and regions P . Therefore, if the sets $R(e)$ and $G(e)$ are blocks of any partition of the state space, then no partition refinement is necessary in the bisimulation algorithm due to any discrete transitions $e \in E$. This fact, in a sense, decouples the continuous and discrete components of the hybrid system. In turn, this implies that the initial partition in the bisimulation algorithm should contain the invariants, guards and reset sets, in addition to the initial and final sets. This allows us to carry out the algorithm independently for each discrete location.

More precisely, given any region $P \subseteq X$, and $q \in X_D$, define the set $P_q = \{x \in X_C : (q, x) \in P\}$ as the continuous projection of the set. For each location $q \in X_D$ consider the finite collection of sets

$$\mathcal{A}_q = \{I(q), (X_0)_q, (X_F)_q\} \cup \{G(e)_q, R(e)_q : e \in E\} \quad (4.8)$$

which describes the initial and final states, guards, invariants and resets associated with location q . Let \mathcal{S}_q be the coarsest partition of X_C compatible with the collection \mathcal{A}_q , that is each set in \mathcal{A}_q is a union of sets in \mathcal{S}_q . The finite partitions \mathcal{S}_q can be easily computed by successive intersections between each of the sets in \mathcal{A}_q and their complements. Define (q, \mathcal{S}_q) to be the set $\{\{q\} \times P \mid P \in \mathcal{S}_q\}$. The collections (q, \mathcal{S}_q) will be the starting partition of the bisimulation algorithm. In addition, since by definition $Pre_\tau(P)$ applies to regions $P \subseteq X$ but not to its continuous projection P_q , we define for $Y \subseteq X_C$ and discrete state q the operator $Pre_q(Y) = (Pre_\tau(\{q\} \times Y))_q$. The general bisimulation algorithm for transition systems then takes the following form for the class of hybrid systems under examination.

Bisimulation Algorithm for Hybrid Systems

set $X/\sim = \bigcup_q (q, \mathcal{S}_q)$

```

for  $q \in X_D$ 
  while  $\exists P, P' \in \mathcal{S}_q$  such that  $\emptyset \neq P \cap Pre_q(P') \neq P$  do
    set  $P_1 = P \cap Pre_q(P'), P_2 = P \setminus Pre_q(P')$ 
    refine  $\mathcal{S}_q = (\mathcal{S}_q \setminus \{P\}) \cup \{P_1, P_2\}$ 
  end while
end for

```

It is clear from the structure of the bisimulation algorithm that, the iteration is carried out independently for each discrete location. In order for the above algorithm to terminate, the partition refinement process must terminate for each discrete location $q \in X_D$. It therefore suffices to look at one discrete state of the hybrid system at a time and see whether we can construct a finite bisimulation that is consistent with all relevant sets of each location q as well as with the continuous flows of the vector field $F(q, \cdot)$.

As mentioned before, decidability requires that the above algorithm is computational and the underlying transition must admit a finite bisimulation. We must therefore resolve the following issues:

- **Computability**

- Represent sets symbolically,
- Perform set intersection and complementation,
- Check emptiness of sets,
- Compute $Pre_q(Y)$ of a set Y .

- **Finiteness**

- Determine whether the algorithm terminates in a finite number of steps.

A natural platform for solving some of the computational issues is provided by mathematical logic where sets would be represented as formulas of first-order logic. Boolean operation are natural in logic, and, as noted in Section 2.3, emptiness of a set can be decided in a decidable theory. Furthermore, Examples 6 and 7 hint at the possibility of using quantifier elimination for computing $Pre_q(P)$ for a definable set P . However, none of these concepts allow us to tackle the finiteness issue. The heart of the finiteness problem is illustrated by

the following example which shows a simple hybrid system which does not admit a finite bisimulation.

Example 13. Consider the hybrid system with only one discrete location q and let F be the linear vector field

$$\begin{aligned}\dot{x}_1 &= -x_1 + x_2 \\ \dot{x}_2 &= -x_1 - x_2\end{aligned}\tag{4.9}$$

on \mathbb{R}^2 . Assume the partition of \mathbb{R}^2 consists of the following three sets (see Figure 4.2):

$$\begin{aligned}P_1 &= \{(x, 0) : 0 \leq x \leq 4\} \\ P_2 &= \{(x, 0) : -4 \leq x < 0\} \\ P_3 &= \mathbb{R}^2 \setminus (P_1 \cup P_2)\end{aligned}$$

The integral curves of F are spirals moving away from the origin. The first iteration of the algorithm partitions P_2 into $P_4 = P_2 \cap \text{Pre}_q(P_1) = \{(x, 0) : x_1 \leq x < 0\}$ and $P_2 \setminus \text{Pre}_q(P_1)$. Here $x_1 < 0$ is the x -coordinate of the first intersection point of the spiral through $(4, 0)$ with P_2 . The second iteration subdivides P_1 into $P_5 = P_1 \cap \text{Pre}_q(P_4) = \{(x, 0) : 0 \leq x \leq x_2\}$ and $P_1 \setminus \text{Pre}_q(P_4)$ where $x_2 > 0$ is the x -coordinate of the next point of intersection of the spiral with P_1 . This process continues indefinitely since the spiral intersects P_1 in infinitely many points, and therefore the algorithm does not terminate.

From the above example it is clear that the critical problem is the intersection of the flow of $F(q, \cdot)$ with the sets \mathcal{S}_q for a single location q . For example, it is important that during the partition refinement process, the *global* intersection of the predecessor of an equivalence class with any other equivalence class has a finite number of connected components. Such finite interaction properties are reminiscent of the properties enjoyed by the classes of sets reviewed in Section 2.2. In the past decade, very recent developments in mathematics have captured these desirable geometric properties in the framework of mathematical logic. This astonishing connection between geometry and logic is the mathematical formalism that will allow us to tackle both computational and finiteness issues within a unified framework.

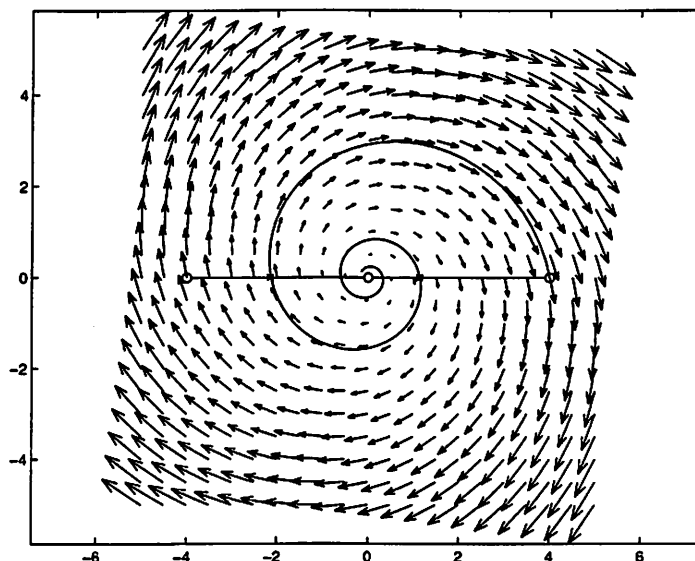


Figure 4.2: Bisimulation algorithm does not terminate

4.3 O-Minimal Theories

Geometric model theory is a very recent and growing area of mathematical logic that studies the relationship between theories of the reals and properties of their definable sets. The search for desirable finiteness properties of definable sets has led to the notion of *o-minimality* or *order-minimality*. While this concept applies to any theory, we only consider theories over the real numbers.

Definition 4.6 (O-Minimal Theories). *A theory of the reals is o-minimal if every definable subset of \mathbb{R} is a finite union of points and intervals (possibly unbounded).*

For example, consider the $A = \{x \in \mathbb{R} \mid p(x) \leq 0\}$, where $p(x)$ is a polynomial. Since any polynomial has a finite number of real roots, the set A can only consist of a finite number of intervals. O-minimality requires that this property is true even if replace the polynomial $p(x)$ by any first order formula $\phi(x)$, including quantifiers!

The class of o-minimal theories is quite rich. Consider first, the theories $\text{Lin}(\mathbb{R})$ and $\text{OF}(\mathbb{R})$, defined in Section 2.3. Since both of these theories admit quantifier elimination [98], every definable subset of \mathbb{R} in $\text{Lin}(\mathbb{R})$ is a semilinear set, whereas every definable set \mathbb{R} in $\text{OF}(\mathbb{R})$ is a semialgebraic set. But then semilinear and semialgebraic sets on the real line can

only have a finite number of connected components. This immediately shows that $\text{Lin}(\mathbb{R})$ and $\text{OF}(\mathbb{R})$ are o-minimal.

The search for new o-minimal theories started by extending $\text{OF}(\mathbb{R})$ by *restricted analytic functions*. Given a real analytic f in a neighborhood of the cube $[-1, 1]^n \subset \mathbb{R}^n$, let $\hat{f}: \mathbb{R}^n \rightarrow \mathbb{R}$ be the function defined by

$$\hat{f}(x) = \begin{cases} f(x) & \text{if } x \in [-1, 1]^n \\ 0 & \text{otherwise} \end{cases}$$

Function \hat{f} is a restricted analytic function since it restricts f on a compact cube. The resulting theory, denoted by $\text{OF}_{an}(\mathbb{R})$ seems quite unnatural. Remarkably, definable sets in $\text{OF}_{an}(\mathbb{R})$ capture the bounded subanalytic sets described in Section 2.2! Furthermore, it was shown in [35] that $\text{OF}_{an}(\mathbb{R})$ is o-minimal.

Therefore, definable sets in o-minimal theories provide a uniform way of capturing exactly the desirable properties enjoyed by semilinear, semialgebraic, and subanalytic sets. However, for the purposes of the bisimulation algorithm, we also need well behaved flows of vector fields. Even though o-minimal theories capture desirable classes of sets, flows of vector fields require functions that are globally defined. As shown in Example 6, the flow of the vector field $\dot{x} = 1$ is definable in $\text{Lin}(\mathbb{R})$. In general, $\text{OF}(\mathbb{R})$ gives us the modeling power to describe vector fields with polynomial flows. Restricted analytic functions are, by definition, restricted on a bounded time interval, and therefore do not capture any new flows of vector fields.

Fortunately, a big breakthrough occurred in [111], where it was shown that $\text{OF}_{exp}(\mathbb{R})$ is o-minimal. Recall from Section 2.3 that $\text{OF}_{exp}(\mathbb{R})$ extends $\text{OF}(\mathbb{R})$ by the globally defined, exponential function e^x . Globally defined exponential functions allow us to capture flows of linear vector fields within an o-minimal theory, as shown in Example 7. Furthermore, in [107], it was shown that $\text{OF}_{exp,an}(\mathbb{R})$, the model that extends both $\text{OF}_{exp}(\mathbb{R})$ and $\text{OF}_{an}(\mathbb{R})$, is o-minimal. Table 4.3 summarizes o-minimal theories, along with some examples of sets and flows that are definable in these theories.

In addition to having desirable finiteness properties, definable sets in o-minimal structures are free of topological pathologies. Many topological and geometric properties of o-minimal theories can be found in [106]. In the remainder of this section, we present those topological properties that are used in subsequent analysis.

Theory	Model	Definable Sets	Definable Flows
$\text{Lin}(\mathbb{R})$	$(\mathbb{R}, +, -, <, 0, 1)$	Semilinear sets	Linear flows
$\text{OF}(\mathbb{R})$	$(\mathbb{R}, +, -, \times, <, 0, 1)$	Semialgebraic sets	Polynomial flows
$\text{OF}_{an}(\mathbb{R})$	$(\mathbb{R}, +, -, \times, <, 0, 1, \{f\})$	Subanalytic sets	Polynomial flows
$\text{OF}_{\text{exp}}(\mathbb{R})$	$(\mathbb{R}, +, -, \times, <, 0, 1, e^x)$	Semialgebraic sets	Exponential flows
$\text{OF}_{\text{exp},an}(\mathbb{R})$	$(\mathbb{R}, +, -, \times, <, 0, 1, e^x, \{f\})$	Subanalytic sets	Exponential flows

Table 4.1: Definable sets and flows in o-minimal theories

Consider a fixed o-minimal theory of the reals, and let definability refer to this theory. Let $f : A \rightarrow B$ be a function. The graph of f is defined as $\Gamma(f) = \{(x, f(x)) \mid x \in A\} \subseteq A \times B$. A function f is definable if its graph is a definable set. We can now define cells as nonempty definable sets of a particularly simple form.

Definition 4.7 (Cells). *Cells in \mathbb{R}^n are inductively defined as follows:*

1. The cells in \mathbb{R} are points $\{c\}$ with $c \in \mathbb{R}$ and open intervals (a, b) with $-\infty \leq a < b \leq +\infty$.
2. Let $C \subseteq \mathbb{R}^n$ be a cell and let $f, g : C \rightarrow \mathbb{R}$ be definable continuous functions such that $f < g$ on C . Then the following are cells in \mathbb{R}^{n+1}
 - $(-\infty, f) = \{(x, r) \in C \times \mathbb{R} : r < f(x)\}$,
 - $\Gamma(f) = \{(x, f(x)) \mid x \in C\}$,
 - $(f, g) = \{(x, r) \in C \times \mathbb{R} : f(x) < r < g(x)\} \subseteq \mathbb{R}^{n+1}$,
 - $\Gamma(g) = \{(x, g(x)) \mid x \in C\}$,
 - $(f, +\infty) = \{(x, r) \in C \times \mathbb{R} : f(x) < r\}$
 - $C \times \mathbb{R}$

A more geometric and useful view of cells is as fibers over their projections, as shown in Figure 4.3. Now recall from Section 2.2 that given a collection of subanalytic sets, there exists a stratification (partition) of \mathbb{R}^n , compatible with the collection of these sets. The following theorem should be thought of as a generalization of Theorem 2.11 for all o-minimal theories.

Theorem 4.8 (Cell Decomposition [53]). *Given any finite family $\{A_1, \dots, A_l\}$ of definable subsets of \mathbb{R}^n there exists a finite partition of \mathbb{R}^n into cells so that each A_i is a union of such cells.*

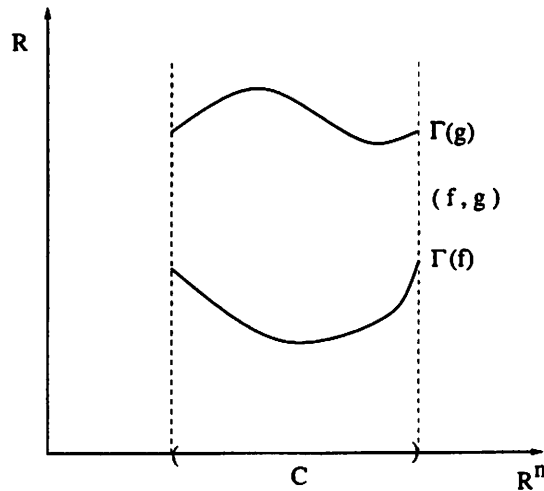


Figure 4.3: Inductive definition of cells

Note that o-minimality is a finiteness condition of definable sets on the real line. However, o-minimality and the cell decomposition theorem constrain definable sets in \mathbb{R}^n to have an analogous finiteness property.

Theorem 4.9 (Uniform Finiteness). *Any definable set has a finite number of connected components, each of which is a definable set. Moreover, if $A \subseteq \mathbb{R}^n \times \mathbb{R}$ is definable then there exists a positive integer N such that for each $x \in \mathbb{R}^n$ the number of connected components of $A_x = \{t \in \mathbb{R} : (x, t) \in A\}$ is less than N .*

Therefore, even though each fiber A_x over x has a finite number of connected components, o-minimality and cell decomposition provide us with a globally uniform bound of connected components that A_x can be partitioned to, independent of x !

Finally, recall from elementary topology that arcwise connected sets are connected. The converse is not always true, and a classic counterexample is related to the construction used in Example 5. Fortunately, sets definable in o-minimal theories are free from such pathologies.

Theorem 4.10 (Connectedness). *If A is definable and connected, then it is arcwise connected.*

Definable sets in o-minimal theories enjoy many more nice topological properties. A very nice introduction to their topology can be found in [106].

4.4 O-Minimal Hybrid Systems

As shown in Example 13, the termination of the bisimulation algorithm critically depends on whether the intersection of trajectories and sets consists of a finite number of connected components. If, however, all relevant sets and trajectories are definable in an o-minimal theory, then such a possibility is avoided. This motivates the following definition.

Definition 4.11 (O-Minimal Hybrid Systems). *A hybrid system, defined in Definition 4.5 as a tuple $H = (X, X_0, X_F, F, E, I, G, R)$, is said to be o-minimal if*

- $X_C = \mathbb{R}^n$
- for each $q \in X_D$ the flow of $F(q, \cdot)$ is complete
- for each $q \in X_D$ the family of sets

$$\mathcal{A}_q = \{I(q), (X_0)_q, (X_F)_q\} \cup \{G(e)_q, R(e)_q : e \in E\}$$

and the flow of $F(q, \cdot)$ are definable in the same o-minimal theory of \mathbb{R} .

Therefore a hybrid system is called o-minimal if for each discrete location, the invariants, guards, resets, initial and final conditions, as well as the flow of the differential equation are definable in the same o-minimal theory. Different o-minimal theories could be used in different discrete locations. For example, in one discrete location one can use $\text{Lin}(\mathbb{R})$ to describe polyhedral sets and linear flows, whereas in another location of the same hybrid system one can use $\text{OF}_{\text{exp}}(\mathbb{R})$ to describe semialgebraic sets and flows of some linear vector fields.

Theorem 4.12 (Finite Bisimulations). *Every o-minimal hybrid system admits a finite bisimulation. Equivalently, the bisimulation algorithm terminates for all o-minimal hybrid systems.*

Proof. We will assume that we are given a fixed o-minimal theory of the reals in which all relevant objects are definable. From now on, definable will mean definable in this fixed o-minimal theory. We start by applying the cell decomposition theorem on each family \mathcal{A}_q . As mentioned in Section 4.2, the special form of $\text{Pre}_e(P)$ allows us to construct the bisimulation quotient on each set $\{q\} \times X_C$ separately. Therefore, we assume given a finite

partition \mathcal{P} of \mathbb{R}^n into definable sets and a vector field F whose flow is definable. Moreover, we will simply write Pre for Pre_q .

The outline of the proof is as follows. We first perform an initial finite refinement $\tilde{\mathcal{P}}$ of \mathcal{P} which has the property that the intersection of any trajectory with each set has one connected component. Because of this property we can use a slight variation of the iterative step of the bisimulation algorithm to construct a finite partition \mathcal{B} which is a further refinement, and satisfies the bisimulation condition, namely, that for any $B \in \mathcal{B}$, the set $Pre(B)$ is a finite union of set in \mathcal{B} . This guarantees that the bisimulation algorithm terminates.

We first notice that if $f : \mathbb{R} \rightarrow \mathbb{R}^n$ is continuous, periodic, and not constant, then f is not definable. Indeed, for such f there is $y \in \mathbb{R}^n$ such that the set $R = \{x \in \mathbb{R} : f(x) = y\}$ consists of an infinite number of isolated points. On the other hand, if f is definable, then so is R , but this contradicts o-minimality.

For each $x \in \mathbb{R}^n$, $\gamma_x(t)$ will denote the integral curve of F which passes through x at $t = 0$. That is, $\dot{\gamma}_x(t) = F(\gamma_x(t))$ and $\gamma_x(0) = x$. Therefore, $\Phi(x, t) = \gamma_x(t)$ denotes the flow of F and is definable by hypothesis. Combining this with the comment above we conclude that for each $x \in \mathbb{R}^n$, $\gamma_x(\cdot)$ is either constant or injective.

We will need the following lemma.

Lemma 4.13. *Let F and $\Phi(x, t)$ be as above, and let γ be an integral curve of F . Define $\Gamma = Im(\gamma) = \{\gamma(t) : t \in \mathbb{R}\}$. Let S be a definable set and C a connected component of $\Gamma \cap S$. If $t_0, t_1 \in \mathbb{R}$ are such that $\gamma(t_0), \gamma(t_1) \in C$, then $\gamma(t) \in C$ for all $t_0 \leq t \leq t_1$.*

Proof. Since C is definable and connected, it is also arcwise connected. Let $\beta : [0, 1] \rightarrow C$ be continuous and such that $\beta(0) = \gamma(t_0)$ and $\beta(1) = \gamma(t_1)$. If γ is constant there is nothing to prove. We can then assume γ is injective and $F(\gamma(t)) \neq 0$ for any t . Therefore, the restriction of γ to any compact interval $[a, b]$ is a homeomorphism between $[a, b]$ and $\gamma([a, b])$. If $\beta([0, 1]) \subseteq \gamma([a, b])$ then $\gamma^{-1} \circ \beta$ is continuous and so $\gamma^{-1} \circ \beta([0, 1])$ is an interval containing t_0, t_1 . Therefore, for all $t \in [t_0, t_1]$, $\gamma(t) \in \beta([0, 1]) \subseteq C$ as desired.

Assume then that $\beta([0, 1])$ is not contained in the image under γ of any finite interval. Hence there exist a sequence $\{t_n\}$ with $|t_n| \rightarrow \infty$ and $\gamma(t_n) \in \beta([0, 1])$ for all n . By taking a subsequence if necessary we may assume that $\gamma(t_n) \rightarrow \tilde{x} \in \beta([0, 1])$. Therefore, $\tilde{x} = \gamma(\tilde{t})$ for some $\tilde{t} \in \mathbb{R}$. We will show that this is a contradiction. In a (definable) neighborhood B of \tilde{x} we can make a definable change of coordinates centered at \tilde{x} , so that

in these coordinates $F \equiv \frac{\partial}{\partial x_1}$. In fact, after a translation and rotation (which are definable¹) we can assume that $\tilde{x} = 0$ and $F(0) = \frac{\partial}{\partial x_1}$. Then the desired change of coordinates is given by

$$(y_1, \dots, y_n) \longrightarrow \Phi((0, y_2, \dots, y_n), y_1)$$

Therefore, in that neighborhood all integral curves of F are of the form $\gamma(t) = (t, a_2, \dots, a_n)$ for some constant a_2, \dots, a_n . By restricting the neighborhood further we may assume it is of the form

$$B = \{(x_1, \dots, x_n) : \underline{a}_i \leq x_i \leq \bar{a}_i\}$$

The set $\Gamma \cap B$ is a union of at most countably many sets of the form $I_{a_2, \dots, a_n} = \{(t, a_2, \dots, a_n) : \underline{a}_1 \leq t \leq \bar{a}_1\}$ and so each such set is a connected component. By o-minimality, $\Gamma \cap B$ is a union of finitely many such sets. By shrinking the set B , if necessary, we may assume that

$$\Gamma \cap B = \{(t, 0, \dots, 0) : \underline{a}_1 \leq t \leq \bar{a}_1\}.$$

For n large enough we must have $\gamma(t_n) \in \Gamma \cap B$. Therefore, for such an n there exists t near \tilde{t} such that $\gamma(t) = \gamma(t_n)$, which contradicts the injectivity of γ . This concludes the proof of the lemma. \square

We now continue with the proof of the main theorem. Given a set S , we define $H = \{(x, t) \in \mathbb{R}^{n+1} : \Phi(x, t) \in S\}$. If S is definable, then H is definable. Moreover, by o-minimality there exists $N_S \in \mathbb{N}$ such that the number of connected components of $H_x = \{t : (x, t) \in H\}$ is less than N for all $x \in \mathbb{R}^n$. This implies that if S is definable and Γ_x denotes the trajectory of F passing through x , then the number of connected components of $\Gamma_x \cap S$ is bounded above by a constant independent of x . We choose $N \in \mathbb{N}$ larger than the corresponding N_S for all sets $S \in \mathcal{P}$.

We begin the construction of the partition \mathcal{B} by subdividing each set S in \mathcal{P} as

¹Note that rotation requires multiplication which does not exist in $\text{Lin}(\mathbb{R})$. However, flows definable in $\text{Lin}(\mathbb{R})$ are already complete and straightened.

follows. Let

$$\begin{aligned}
S_0 &= \{x \in X : \forall t \geq 0 \ \gamma_x(t) \in S\} \\
S_1 &= \{x \in S \setminus S_0 : \forall t \geq 0 (\gamma_x(t) \notin S \setminus S_0 \Rightarrow \forall t' \geq t \ \gamma_x(t') \notin S \setminus S_0)\} \\
&\vdots \\
S_i &= \{x \in S \setminus (S_0 \cup \dots \cup S_{i-1}) : \\
&\quad \forall t \geq 0 (\gamma_x(t) \notin S \setminus (S_0 \cup \dots \cup S_{i-1}) \Rightarrow \forall t' \geq t \ \gamma_x(t') \notin S \setminus (S_0 \cup \dots \cup S_{i-1}))\} \\
&\vdots
\end{aligned}$$

The set S_i is clearly definable for every i . For $i \geq 1$ the set S_i consists of those x for which γ_x leaves the set $S \setminus (S_0 \cup \dots \cup S_{i-1})$ but never returns to it.

Claim: $S_k = \emptyset$ for $k \geq N$.

To prove the claim it suffices to show that if $x \in S_i$ with $i \geq 1$, then $\Gamma_x \cap S$ has at least i connected components. To prove this we will use a couple of lemmas.

Lemma 4.14. *Let S and S_i , $i \geq 0$ be as above. Let I be an interval and $\gamma(\cdot)$ an integral curve of F such that $\gamma(I) \subseteq S$. If $\gamma(t_0) \in S_i$ for some $t_0 \in I$, then $\gamma(I) \subseteq S_i$.*

Proof. We proceed by induction. The statement is clearly true for S_0 . Assume it holds for $i \leq k$. Let $\gamma(I) \subseteq S$, $t_0 \in I$ and $\gamma(t_0) \in S_{k+1}$. Then $\gamma(t_0) \in S \setminus (S_0 \cup \dots \cup S_k)$. For any $t \in I$, if $\gamma(t) \in S_0 \cup \dots \cup S_k$ then there is $j \leq k$ such that $\gamma(t) \in S_j$. By the inductive hypothesis, $\gamma(I) \subseteq S_j$, but this contradicts $\gamma(t_0) \notin S_j$. Therefore we have $\gamma(I) \subseteq S \setminus (S_0 \cup \dots \cup S_k)$. Let $t \in I$ and $t' > t$ be such that $\gamma(t') \notin S \setminus (S_0 \cup \dots \cup S_k)$. Then $t' \notin I$ and so $t' > t_0$. Since $\gamma(t_0) \in S_{k+1}$ we conclude that for any $t'' > t'$ we get $\gamma(t'') \notin S \setminus (S_0 \cup \dots \cup S_k)$. This shows that $\gamma(I) \subseteq S_{k+1}$. \square

Lemma 4.15. *If $x \in S_i$ for $i \geq 2$ then there exist $t_1 > s_1 > t_2 > \dots > s_{i-2} > t_{i-1} > s_{i-1} > 0$ such that $\gamma_x(s_j) \notin S$ and $\gamma_x(t_j) \in S_j$ for $j = 1, \dots, i-1$.*

Proof. We proceed by induction. Let $x \in S_2$. Then $x \in S \setminus (S_0 \cup S_1) \subseteq S \setminus S_1$. Therefore there exist $t > s > 0$ such that $\gamma_x(s) \notin S \setminus S_0$ and $\gamma_x(t) \in S \setminus S_0$. We can not have $\gamma_x(s) \in S_0$ because then we would also have $\gamma_x(t) \in S_0$. Therefore $\gamma_x(s) \notin S$. We set $s_1 = s$. If $\gamma_x(t) \in S_1$ then we set $t_1 = t$. Otherwise, there exist $t' > s' > t$ such that $\gamma_x(s') \notin S \setminus S_0$ and $\gamma_x(t') \in S \setminus S_0$. Since $x \in S_2$, $\gamma_x(s) \notin S \setminus (S_0 \cup S_1)$, and $t' > s$ we must

have $\gamma_x(t') \notin S \setminus (S_0 \cup S_1)$. Therefore $\gamma_x(t') \in S_1$ and we set $t_1 = t'$. This completes the proof for the case $i = 2$.

Assume now the conclusion holds for i and let $x \in S_{i+1}$. In particular, $x \in S \setminus S_i$, and there are $t > s > 0$ such that $\gamma_x(s) \notin S \setminus (S_0 \cup \dots \cup S_{i-1})$ and $\gamma_x(t) \in S \setminus (S_0 \cup \dots \cup S_{i-1})$. If $\gamma_x(s) \in S_j$ for some $j \leq i - 1$ and $\gamma_x(\bar{s}) \in S$ for all $s \leq \bar{s} \leq t$, then Lemma 4.14 would imply that $\gamma_x(t) \in S_j$ which is not true. Therefore there exists \bar{s} , $s \leq \bar{s} < t$ such that $\gamma_x(\bar{s}) \notin S$. We set $s_i = \bar{s}$.

If $\gamma_x(t) \in S_i$ then we set $t_i = t$. Otherwise, there exist $t' > s' > t$ such that $\gamma_x(s') \notin S \setminus (S_0 \cup \dots \cup S_{i-1})$ and $\gamma_x(t') \in S \setminus (S_0 \cup \dots \cup S_{i-1})$. Since $x \in S_{i+1}$, $\gamma_x(\bar{s}) \notin S \setminus (S_0 \cup \dots \cup S_i)$, and $t' > \bar{s}$ we must have $\gamma_x(t') \notin S \setminus (S_0 \cup \dots \cup S_i)$. Therefore $\gamma_x(t') \in S_i$ and we set $t_i = t'$.

By the inductive hypothesis there exist $\tilde{t}_1 > \tilde{s}_1 > \dots > \tilde{t}_{i-1} > \tilde{s}_{i-1} > 0$ such that $\gamma_{\gamma_x(t_i)}(\tilde{s}_j) \notin S_j$, $\gamma_{\gamma_x(t_i)}(\tilde{t}_j) \in S_j$, for $j = 1, \dots, i - 1$. Setting $s_j = \tilde{s}_j + t_i$, $t_j = \tilde{t}_j + t_i$ for $j = 1, \dots, i - 1$ we get the desired conclusion. \square

The last lemma together with Lemma 4.13 proves that if $x \in S_i$ then $\Gamma_x \cap S$ has at least i connected components. This, in turn, proves the claim.

Notice that Lemma 4.13 also implies that if $x \in S_i$ then $\Gamma_x \cap S_i$ has exactly one connected component.

By carrying out the subdivision into the sets S_i for all $S \in \mathcal{P}$ we obtain a new finite partition $\tilde{\mathcal{P}}$ of \mathbb{R}^n with the property

(P) For each $S \in \tilde{\mathcal{P}}$, and each trajectory γ of F such that $\gamma(t_0), \gamma(t_1) \in S$ we have $\gamma(t) \in S$ for all t with $t_0 \leq t \leq t_1$. In particular, for each $x \in S$, the set $\Gamma_x \cap S$ has exactly one connected component.

We will denote by $\rho = \rho(\tilde{\mathcal{P}})$ the number of sets in $\tilde{\mathcal{P}}$ and write $\tilde{\mathcal{P}} = \{S_i : i = 1, \dots, \rho\}$.

We introduce two functions, I and C , acting on pairs of sets, defined by

$$I(A, B) = A \cap \text{Pre}(B)$$

$$C(A, B) = A \setminus \text{Pre}(B)$$

It is clear that if A and B are definable, then $I(A, B)$ and $C(A, B)$ are definable. Notice also that for each A, B the sets $I(A, B), C(A, B)$ form a partition of A .

For each i , $1 \leq i \leq \rho$ consider all the partitions of S_i defined by

$$I(S_i, Q(S_{j_1}, Q(S_{j_2}, \dots, Q(S_{j_{k-1}}, S_{j_k}) \dots))) \quad (4.10)$$

$$C(S_i, Q(S_{j_1}, Q(S_{j_2}, \dots, Q(S_{j_{k-1}}, S_{j_k}) \dots))) \quad (4.11)$$

where Q is either I or C and $1 \leq j_l \leq \rho$ for $l = 1, \dots, k$. This is a finite collection of partitions. We let \mathcal{B} denote the coarsest partition of \mathbb{R}^n compatible with all such partitions.

Claim: \mathcal{B} is a bisimulation.

The intuitive basis for this proof is the fact that the partitions constructed so far are done “along the flow of F .” That is, two sets in \mathcal{B} which are subsets of the same set in $\tilde{\mathcal{P}}$ can not be connected by a trajectory of F .

To prove the claim first notice that the sets in \mathcal{B} are (finite) intersections of sets of the form (4.10) or (4.11). Notice also that by construction \mathcal{B} is a refinement of \mathcal{P} .

To check the bisimulation property let $B \in \mathcal{B}$, $B \subseteq S \in \tilde{\mathcal{P}}$, be written as

$$B = \bigcap_{l=1}^m P_l$$

where each P_l is of the form (4.10) or (4.11). We want to prove first that

$$Pre(B) = \bigcap_{l=1}^m Pre(P_l). \quad (4.12)$$

The inclusion $Pre(B) \subseteq \bigcap_{l=1}^m Pre(P_l)$ is straightforward. For the other one let $x \in \bigcap_{l=1}^m Pre(P_l)$. For each l there exists $t_l \geq 0$ such that $\gamma_x(t_l) \in P_l$. Each set P_l is of the form $I(S_i, A_l)$ or $C(S_i, A_l)$ for some A_l 's. Hence, $\gamma_x(t_l) \in S_i$ for all l . We now want to show that indeed $\gamma_x(t_l) \in B$ for all t_l . Consider the following property of a set A .

(Q) for any trajectory γ of F , if $\gamma(s_0) \in A \subseteq S \in \tilde{\mathcal{P}}$, then for all s with $\gamma(s) \in S$, $\gamma(s) \in A$.

We show that if a set A has Property **(Q)**, then so do $I(S', A)$ and $C(S', A)$ for any $S' \in \tilde{\mathcal{P}}$. Let $\gamma(s_0) \in I(S', A) \subseteq S'$. Then $\gamma(s_0) \in S'$ and there exists $t \geq s_0$ such that $\gamma(t) \in A$. If $\gamma(t) \in S'$, then we have $S = S'$ since both belong to $\tilde{\mathcal{P}}$. By **(Q)** $\gamma(s) \in A \subseteq Pre(A)$ for all s such that $\gamma(s) \in S'$. Therefore $\gamma(s) \in I(S', A)$ for all such s . On the other hand, if $\gamma(t) \notin S'$, then $A \cap S' \subseteq S \cap S' = \emptyset$. Let $\gamma(s) \in S'$. By Property **(P)** applied to S' we get that $s \leq t$. But then $\gamma(s) \in Pre(A) \cap S'$ as desired. The proof for $C(S', A)$ is analogous.

Proceeding by induction it is easy to show that the sets P_l have Property **(Q)** and this completes the proof of (4.12).

Notice also, that $Pre(A \cup B) = Pre(A) \cup Pre(B)$ for all sets A, B .

To complete the proof that \mathcal{B} is a bisimulation we only need to show that for each l , and each set $S \in \tilde{\mathcal{P}}$, the set $S \cap Pre(P_l)$ is a union of sets in \mathcal{B} . The set $S \cap Pre(P_l) = I(S, P_l)$ is of the form (4.10) with $k \leq \rho + 1$. If $k < \rho + 1$ we already know that $I(S, P_l)$ is a union of sets in \mathcal{B} . We only need to consider the case $k = \rho + 1$.

There are two possibilities for $I(S, P_l)$:

1. there are two occurrences of C in $I(S, P_l)$,
2. there are $\rho + 1$ occurrences of I in $I(S, P_l)$, and therefore, at least one $S_i \in \tilde{\mathcal{P}}$ is repeated as an argument of I .

In case 1 the following two formulas, and boolean algebra, show how to rewrite $I(S, P_l)$ either with fewer terms or using only I .

$$C(S_3, C(S_2, S_1)) = C(S_3, S_2) \cup I(S_3, I(S_2, S_1)) \quad (4.13)$$

$$C(S_3, I(S_2, S_1)) = C(S_3, S_2) \cup I(S_3, C(S_2, S_1)) \quad (4.14)$$

Both formulas can be proved with arguments similar to the ones above, relying on Property (P).

Finally, in case 2 we can show, again using (P) that $I(S, A) = \emptyset$. This concludes the proof that \mathcal{B} is a bisimulation. \square

4.5 Classes of O-Minimal Hybrid Systems

In this section, Theorem 4.12 is applied to several classes of o-minimal hybrid systems. For each o-minimal theory of Table 4.3, we provide examples of definable, o-minimal hybrid systems. While it is clearly possible to identify other special cases, the ones described below cover versions of most known results, and several natural extensions.

The theory $\text{Lin}(\mathbb{R})$

The definable sets in this theory capture semilinear sets whereas the definable flows capture linear flows. Therefore, Theorem 4.12 applied to the o-minimal theory $\text{Lin}(\mathbb{R})$ results in the following corollary.

Corollary 4.16. *Consider hybrid system H where*

- *all relevant sets are semilinear*
- *all flows are linear*

Then H admits a finite bisimulation.

Furthermore, since $\text{Lin}(\mathbb{R})$ is not only o-minimal but also decidable, there is a computational procedure for computing $\text{Pre}_q(P)$ of a definable set P . Therefore, the bisimulation algorithm is both finite and computable, which immediately leads to a decidable class of o-minimal hybrid systems. As a result, Corollary 4.16 captures versions of timed automata [3], multirate automata [2], and rectangular automata [48, 91]. In general, timed automata also allow identity maps as reset maps, whereas rectangular automata allow for identity reset maps as long as the dynamics from one location to another remain the same. This discrete behavior is not allowed in our hybrid model.

The theory $\text{OF}(\mathbb{R})$

The definable sets in $\text{OF}(\mathbb{R})$ are the semialgebraic sets, whereas the definable flows in this theory are polynomial. Therefore, the hybrid systems corresponding to this theory are hybrid systems H , where all sets are semialgebraic and all flows are polynomial. Moreover, since $\text{OF}(\mathbb{R})$ is a decidable theory, we immediately obtain a new class of decidable hybrid systems.

Corollary 4.17. *Consider hybrid system H where*

- *all relevant sets are semialgebraic,*
- *all flows are polynomial.*

Then H admits a finite bisimulation.

The o-minimality of this structure can also be used to show the existence of finite bisimulations in special cases when the flow is not definable. This was illustrated in [64] for the case of planar hybrid systems whose vector fields admit definable (polynomial) Hamiltonians. This captures the decidability result of [28].

The theory $\text{OF}_{an}(\mathbb{R})$

In addition to semialgebraic sets, the definable sets in this theory, include bounded subanalytic sets. Even though polynomial flows are definable in this theory, since the restricted analytic functions \hat{f} are zero outside a compact set, they cannot be used to define complete flows. However, the Pre_q operator corresponding to some periodic flows may still be definable. Consider for example, a hybrid system H whose vector fields are diagonalizable linear vector fields with purely imaginary eigenvalues. Since the restriction of \sin on $[-\pi, \pi]$ is definable, the Pre_q operator corresponding to this linear vector field is definable. This leads to the following corollary.

Corollary 4.18. *Consider hybrid system H where*

- *all relevant sets are semialgebraic or bounded subanalytic,*
- *all vector fields are diagonalizable, linear vector fields with purely imaginary eigenvalues.*

Then H admits a finite bisimulation.

Note however that since $\text{OF}_{an}(\mathbb{R})$ is not known to be a decidable theory, the above corollary is not a decidability result.

The theory $\text{OF}_{exp}(\mathbb{R})$

The main difference between $\text{OF}_{exp}(\mathbb{R})$ and the previous theories, besides enriching the class of definable sets, is the fact that the symbol e^x represents a globally defined function. This allows new classes of definable flows. In particular, the flows of linear vector fields with real eigenvalues are definable.

Corollary 4.19. *Consider hybrid system H where*

- *all relevant sets are semialgebraic,*
- *all vector fields are diagonalizable, linear vector fields with real eigenvalues.*

Then H admits a finite bisimulation.

Recall from Section 2.3 that it is not known whether $\text{OF}_{exp}(\mathbb{R})$ is decidable. In fact in [73] it was shown that it would be a consequence of Schanuel's conjecture in number theory.

The theory $\text{OF}_{\text{exp,an}}(\mathbb{R})$

This theory extends both $\text{OF}_{\text{an}}(\mathbb{R})$ and $\text{OF}_{\text{exp}}(\mathbb{R})$. Corollaries 4.18 and 4.19 can be combined to obtain the following result.

Corollary 4.20. *Consider hybrid system H where*

- *all relevant sets are semialgebraic or bounded subanalytic,*
- *all vector fields are*
 - *either diagonalizable, linear vector fields with real eigenvalues,*
 - *or diagonalizable, linear vector fields with purely imaginary eigenvalues.*

Then H admits a finite bisimulation.

The above corollary extends the planar results in [64] to \mathbb{R}^n . Note that relaxations of Corollary 4.20 would allow spiraling, linear vector fields which are not definable in $\text{OF}_{\text{exp,an}}(\mathbb{R})$. As was shown in Example 13, such systems, in general, do not admit finite bisimulations. Therefore even though Theorem 4.12 provides sufficient conditions for obtaining finite bisimulations, Corollaries 4.16 to 4.20 as well as Example 13 show that the sufficient conditions of Theorem 4.12 are tight.

4.6 Linear Hybrid Systems

Whereas the goal of the previous two sections were to find conditions that guarantee the termination of the bisimulation algorithm, the goal of this section is to make the bisimulation algorithm constructive. This critically depends on being able to compute $\text{Pre}_q(Y)$ for definable sets Y in each discrete location q . If the sets and flows are definable in a theory which admits quantifier elimination, then this reachability computation can be performed as shown in Examples 6 and 8. Since $\text{Lin}(\mathbb{R})$ and $\text{OF}(\mathbb{R})$ admit quantifier elimination, Corollaries 4.16 and 4.17 are not only existential but also constructive, immediately leading to decidability results. However, the theories associated with Corollaries 4.18 to 4.20 do not admit quantifier elimination. In order to be able to perform reachability computations in these theories, the strategy will be to transform formulas in these theories to *equivalent* formulas in the decidable theory $\text{OF}(\mathbb{R})$. Even though this forces us to use semi-algebraic sets in the system description, it will allow us to compute reachable sets for hybrid

systems with linear vector fields in each discrete location. This class of hybrid systems is defined next.

Definition 4.21 (Linear Hybrid Systems). *A hybrid system $H = (X, X_O, X_F, F, E, I, G, R)$ is called linear if*

- $X_C = \mathbb{R}^n$.
- for each $q \in X_D$ the family of sets

$$\mathcal{A}_q = \{I(q), (X_O)_q, (X_F)_q\} \cup \{G(e)_q, R(e)_q \mid e \in E\}$$

is definable in $OF(\mathbb{R})$.

- for each $q \in X_D$ the vector field $F(q, x) = A_q x$, where $A_q \in \mathbb{Q}^{n \times n}$.

Linear hybrid systems should be distinguished from the notion of linear hybrid automata which are hybrid automata with linear flows, and not vector fields, in each discrete location. As indicated previously, because of the structure of the bisimulation algorithm, we only need to investigate a single location and a single linear vector field $F(x) = Ax$ where the subscript q is dropped for notational convenience.

Since the invariant $I(q)$ associated with discrete state q is a definable set, there exists a formula $I(x)$ such that $I(q) = \{x \in \mathbb{R}^n \mid I(x)\}$. Now let $Y \triangleq \{y \in \mathbb{R}^n \mid P(y)\}$ be a definable set. Then we can write explicitly

$$\begin{aligned} Pre(Y) = \{x \in \mathbb{R}^n \mid \exists y \exists t : P(y) \wedge t \geq 0 \wedge x = e^{-tA}y \\ \wedge \forall t' : 0 \leq t' \leq t \implies I(e^{-t'A}y)\} \end{aligned}$$

In order to simplify the following presentation, we will assume that $I(x)$ is *true*. In this case, the above definition reduces to

$$Pre(Y) = \{x \in \mathbb{R}^n \mid \exists y \exists t : P(y) \wedge t \geq 0 \wedge x = e^{-tA}y\} \quad (4.15)$$

$$= \{x \in \mathbb{R}^n \mid \eta(x)\} \quad (4.16)$$

It will be clear from the following results that more complicated invariant sets can be dealt with by the same techniques.

From equation (4.15), we have that $Pre(Y)$ is definable in theories which do not admit quantifier elimination. Our goal in this section is to transform formula $\eta(x)$ to an equivalent formula in $OF(\mathbb{R})$, which is indeed decidable. Based on the eigenstructure of A , we identify several classes of linear vector fields for which this transformation is feasible.

4.6.1 Nilpotent matrices

We consider first the special case when the vector field is linear with a nilpotent matrix A , that is, $A^n = 0$. Recall that nilpotent matrices can only have zero as an eigenvalue. Another important property of nilpotent matrices is that we can express e^{-tA} explicitly as a finite sum

$$e^{-tA} = \sum_{k=0}^{n-1} (-1)^k \frac{t^k}{k!} A^k \quad (4.17)$$

Thus, the formula $\eta(x)$ can be rewritten as follows:

$$\begin{aligned} \eta(x) &\triangleq \exists y \exists t : P(y) \wedge t \geq 0 \wedge x = \sum_{k=0}^{n-1} (-1)^k \frac{t^k}{k!} A^k y \\ &\triangleq \exists y : P(y) \wedge \mu(x, y) \end{aligned}$$

Clearly, $\mu(x, y)$ is a formula in $\text{OF}(\mathbb{R})$, and so is $\eta(x)$, which implies that the following proposition holds.

Proposition 4.22. *Let $F(x) = Ax$ be a linear vector field and $A \in \mathbb{Q}^{n \times n}$ a nilpotent matrix, and $Y \subseteq \mathbb{R}^n$ definable in $\text{OF}(\mathcal{R})$. Then $\text{Pre}(Y)$ is definable in $\text{OF}(\mathcal{R})$.*

Therefore, based on the computational procedure for eliminating quantifiers in $\text{OF}(\mathbb{R})$, we can compute $\text{Pre}(Y)$ for linear vector fields with nilpotent matrices. Note that nilpotent linear vector fields capture integrators which are an extremely important class of linear systems.

Example 14. Consider the nilpotent linear vector field defined by

$$\begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \frac{dx_3}{dt} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (4.18)$$

and consider the set $Y = \{(y_1, y_2, y_3) \in \mathbb{R}^3 \mid P(y_1, y_2, y_3)\}$ where

$$P(y_1, y_2, y_3) \triangleq y_1 = 4 \wedge y_2 > 2 \wedge y_2 < 4 \wedge y_3 = 5$$

Then $Pre(Y) = \{(x_1, x_2, x_3) \in \mathbb{R}^3 \mid \eta(x_1, x_2, x_3)\}$ where it can be easily checked that

$$\begin{aligned} \eta(x_1, x_2, x_3) &\triangleq \exists y_1 \exists y_2 \exists y_3 \exists t : P(y_1, y_2, y_3) \wedge t \geq 0 \wedge \\ &\quad x_1 = y_1 - ty_2 + \frac{t^2}{2}y_3 \wedge \\ &\quad x_2 = y_2 - ty_3 \wedge \\ &\quad x_3 = y_3 \end{aligned}$$

Using REDLOG to perform quantifier elimination we get that $\eta(x_1, x_2, x_3)$ is equivalent to the quantifier free formula

$$\begin{aligned} \eta(x_1, x_2, x_3) &\equiv 2x_1x_3 - x_2^2 - 8x_3 + 16 > 0 \wedge \\ &\quad 2x_1x_3 - x_2^2 - 8x_3 + 4 < 0 \wedge \\ &\quad x_3 - 5 = 0 \wedge \\ &\quad (x_1x_3 - 4x_3 \leq 0 \vee x_2x_3 \leq 0) \end{aligned}$$

4.6.2 Diagonalizable matrices with rational eigenvalues

In this case we can write $A = TDT^{-1}$ where D is a diagonal matrix with the eigenvalues of A along the diagonal and both T and T^{-1} have rational entries. Then

$$e^{-tA} = e^{-tTDT^{-1}} = T \begin{bmatrix} e^{-t\lambda_1} & & \\ & \ddots & \\ & & e^{-t\lambda_n} \end{bmatrix} T^{-1} = [f_{ij}(t)] \quad (4.19)$$

where $f_{ij}(t) = \sum_{k=1}^n a_{ijk} e^{-\lambda_k t}$ with $a_{ijk} \in \mathbb{Q}$ for all i, j, k , and $\{\lambda_k\}$ are the eigenvalues of A . Moreover, $x = e^{-tA}y$ can be written component-wise as follows

$$\begin{aligned} x_i &= \sum_{j=1}^n \left(\sum_{k=1}^n a_{ijk} e^{-\lambda_k t} \right) y_j \\ &= \sum_{k=1}^n \left(\sum_{j=1}^n a_{ijk} y_j \right) e^{-\lambda_k t} \\ &= \sum_{k=1}^n \psi_{ik}(y) e^{-\lambda_k t} \end{aligned}$$

Therefore, $\eta(x)$ can be rewritten as follows

$$\begin{aligned}\eta(x) &\triangleq \exists y \exists t : P(y) \wedge t \geq 0 \wedge \bigwedge_{i=1}^n x_i = \sum_{k=1}^n \psi_{ik}(y) e^{-\lambda_k t} \\ &\triangleq \exists y : P(y) \wedge \varphi(x, y)\end{aligned}$$

Since the formula for Y , $P(y)$, is already in $\text{OF}(\mathcal{R})$, we will concentrate on studying $\varphi(x, y)$. First we reparameterize the time t to reduce the problem to integers in the exponent. More precisely, for each $k = 1, \dots, n$ let d_k denote the denominator of λ_k and let $d_0 = \prod d_k$. We assume that the λ_k are in reduced form, with positive denominators. Then $d_0 > 0$ and for each $k = 1, \dots, n$ we write $r_k = \lambda_k d_0$. Then we have that $\varphi(x, y) \equiv \varphi_Z(x, y)$ where

$$\varphi_Z(x, y) \triangleq \exists s : s \geq 0 \wedge \bigwedge_{i=1}^n x_i = \sum_{k=1}^n \psi_{ik}(y) e^{-r_k s} \quad (4.20)$$

Still, φ_Z contains exponentials. We consider a second formula $\zeta(x, y)$ which does not involve the exponential function:

$$\zeta(x, y) \triangleq \exists z : 0 < z \leq 1 \wedge \bigwedge_{i=1}^n x_i = \sum_{k=1}^n \psi_{ik}(y) z^{r_k} \quad (4.21)$$

The following lemma holds.

Lemma 4.23. *Formulas $\varphi_Z(x, y)$ and $\zeta(x, y)$ are equivalent.*

Proof. \Rightarrow . If $\varphi_Z(x, y)$ holds, then there exists $s \geq 0$ such that

$$\bigwedge_{i=1}^n x_i = \sum_{k=1}^n \psi_{ik}(y) e^{-r_k s}$$

Set $z = e^{-s}$. Then $0 < z \leq 1$ and

$$\bigwedge_{i=1}^n x_i = \sum_{k=1}^n \psi_{ik}(y) z^{r_k}$$

so $\zeta(x, y)$ holds.

\Leftarrow . Conversely, if $\zeta(x, y)$ holds, then there exists z , with $0 < z \leq 1$ such that

$$\bigwedge_{i=1}^n x_i = \sum_{k=1}^n \psi_{ik}(y) z^{r_k}$$

By well known properties of the exponential function (continuity, monotonicity, and $e^0 = 1$, $e^{-\infty} = 0$) there exists $s \geq 0$ such that $z = e^{-s}$. Then

$$\bigwedge_{i=1}^n x_i = \sum_{k=1}^n \psi_{ik}(y) e^{-\tau_k s}$$

Hence, $\varphi_z(x, y) \equiv \zeta(x, y)$. □

The third step eliminates negative polynomial powers. It consists of grouping the indices $1, \dots, n$ according to the sign of the corresponding eigenvalue. Let $I^+ = \{k \mid r_k > 0\}$, $I^- = \{k \mid r_k < 0\}$, and $I^0 = \{k \mid r_k = 0\}$. Consider now the following formula:

$$\nu(x, y) \stackrel{\Delta}{=} \exists w_1 \exists w_2 : \tag{4.22}$$

$$\begin{aligned} & w_1 > 0 \wedge w_2 > 0 \wedge w_1 w_2 = 1 \\ & \wedge \bigwedge_{i=1}^n x_i = \sum_{k \in I^+} \psi_{ik}(y) w_1^{r_k} + \sum_{k \in I^-} \psi_{ik}(y) w_2^{-r_k} + \sum_{k \in I^0} \psi_{ik}(y) \end{aligned}$$

Clearly, $\nu(x, y)$ is a formula in $\text{OF}(\mathcal{R})$. Furthermore, we have the following.

Lemma 4.24. *The formulas $\zeta(x, y)$ and $\nu(x, y)$ are equivalent.*

Proof. The equivalence is immediate from the change of variables $w_1 = z$, $w_2 = 1/z$. □

The combination of the above lemmas gives the following proposition.

Proposition 4.25. *Let $F(x) = Ax$ be a linear vector field and $A \in \mathbb{Q}^{n \times n}$ a diagonalizable matrix with rational eigenvalues, and $Y \subseteq \mathbb{R}^n$ definable in $\text{OF}(\mathcal{R})$. Then $\text{Pre}(Y)$ is definable in $\text{OF}(\mathcal{R})$.*

Proof. By the previous lemmas we have that $\eta(x) \equiv \exists y : P(y) \wedge \nu(x, y)$ and $\nu(x, y)$ definable in $\text{OF}(\mathcal{R})$. □

Proposition 4.25 implies that we have a computational procedure for computing reachable sets for diagonalizable linear vector fields with rational eigenvalues. As an illustration of Proposition 4.25, consider the following example.

Example 15. Consider again Example 7. Let $Y = \{(y_1, y_2) \in \mathbb{R}^2 \mid y_1 = 4 \wedge y_2 = 3\}$. Recall that $Pre(Y) = \{(x_1, x_2) \in \mathbb{R}^2 \mid \psi(x_1, x_2)\}$. Applying the previous lemmas we have that

$$\begin{aligned} \psi(x_1, x_2) &\triangleq \exists y_1 \exists y_2 \exists t : y_1 = 4 \wedge y_2 = 3 \wedge t \geq 0 \wedge x_1 = y_1 e^{-2t} \wedge x_2 = y_2 e^t \\ &\equiv \exists y_1 \exists y_2 \exists z : y_1 = 4 \wedge y_2 = 3 \wedge 0 < z \leq 1 \wedge x_1 = y_1 z^{-2} \wedge x_2 = y_2 z \\ &\equiv \exists y_1 \exists y_2 \exists w_1 \exists w_2 : y_1 = 4 \wedge y_2 = 3 \wedge w_1 > 0 \wedge w_2 > 0 \wedge w_1 w_2 = 1 \\ &\quad \wedge x_1 = y_1 w_1^2 \wedge x_2 = y_2 w_2 \\ &\equiv x_1 x_2^2 - 36 = 0 \wedge x_2 > 0 \end{aligned}$$

4.6.3 Diagonalizable matrices with imaginary eigenvalues

In this case the matrix A is similar to a matrix in a special block-diagonal form, a real Jordan form. First, the number of rows (and columns) of A , is even. Second, there exist D and T such that $A = TDT^{-1}$, T invertible, and D is block diagonal with each block of size 2×2 and of the form

$$\begin{bmatrix} 0 & b \\ -b & 0 \end{bmatrix}$$

where b is the imaginary part of an eigenvalue of A . Moreover, if each eigenvalue is of the form ir with $r \in \mathbb{Q}$, then the entries of D , T , and T^{-1} are all rational.

We analyze the formula $x = e^{-tA}y$ in more detail. Assume D has diagonal blocks D_1, \dots, D_m ($n = 2m$). We can write

$$e^{-tA} = e^{-tTDT^{-1}} = T \begin{bmatrix} e^{-tD_1} & & \\ & \ddots & \\ & & e^{-tD_m} \end{bmatrix} T^{-1}.$$

In fact, for a matrix $D = \begin{bmatrix} 0 & b \\ -b & 0 \end{bmatrix}$ we have

$$e^{-tD} = \begin{bmatrix} \cos(bt) & -\sin(bt) \\ \sin(bt) & \cos(bt) \end{bmatrix}.$$

Therefore, we also get

$$e^{-tA} = [f_{ij}(t)]$$

with

$$f_{ij}(t) = \sum_{k=1}^n a_{ijk} \cos(\beta_k t) + b_{ijk} \sin(\beta_k t)$$

with $a_{ijk}, b_{ijk}, \beta_k \in \mathbb{Q}$. The formula $x = e^{-tA}y$ can be written component-wise as follows

$$\begin{aligned} x_i &= \sum_{j=1}^n \left(\sum_{k=1}^n (a_{ijk} \cos(\beta_k t) + b_{ijk} \sin(\beta_k t)) \right) y_j \\ &= \sum_{k=1}^n \left(\sum_{j=1}^n a_{ijk} y_j \right) \cos(\beta_k t) + \sum_{k=1}^n \left(\sum_{j=1}^n b_{ijk} y_j \right) \sin(\beta_k t) \\ &= \sum_{k=1}^n \psi_{ik}^a(y) \cos(\beta_k t) + \psi_{ik}^b(y) \sin(\beta_k t) \end{aligned}$$

Therefore, $\eta(x)$ can be rewritten as follows:

$$\begin{aligned} \eta(x) &\triangleq \exists y \exists t : P(y) \wedge t \geq 0 \wedge \bigwedge_{i=1}^n x_i = \sum_{k=1}^n \psi_{ik}^a(y) \cos(\beta_k t) + \psi_{ik}^b(y) \sin(\beta_k t) \\ &\triangleq \exists y : P(y) \wedge \varphi(x, y) \end{aligned}$$

We now study the formula $\varphi(x, y)$. We start by reparameterizing t as before. That is, for each $k = 1, \dots, n$ let d_k denote the denominator of β_k and let $d_0 = \prod d_k$. We assume that the β_k are in reduced form, with positive denominators. Then $d_0 > 0$ and for each $k = 1, \dots, n$ we write $r_k = \beta_k d_0$. Then we have that $\varphi(x, y) \equiv \varphi_Z$ where

$$\varphi_Z(x, y) \triangleq \exists s : s \geq 0 \wedge \bigwedge_{i=1}^n x_i = \sum_{k=1}^n \psi_{ik}^a(y) \cos(r_k s) + \psi_{ik}^b(y) \sin(r_k s) \quad (4.23)$$

The equivalence is obtained by using the change of variable $t = d_0 s$. The following result will allow us to rewrite $\cos(r_k s)$ and $\sin(r_k s)$ in terms of $\cos s$ and $\sin s$.

Proposition 4.26. *For each integer $m \geq 1$ there exist homogeneous polynomials $f_m(x, y)$ and $g_m(x, y)$ of degree m such that*

$$\cos(ms) = f_m(\cos s, \sin s)$$

$$\sin(ms) = g_m(\cos s, \sin s)$$

Proof. We give a recursive definition. For $m = 1$ we set $f_1(x, y) = x$ and $g_1(x, y) = y$. For $m > 1$ the trigonometric identities

$$\cos(ms) = \cos(s) \cos((m-1)s) - \sin(s) \sin((m-1)s)$$

$$\sin(ms) = \cos(s) \sin((m-1)s) + \sin(s) \cos((m-1)s)$$

lead to the following formulas for f_m and g_m ,

$$\begin{aligned} f_m(x, y) &= x f_{m-1}(x, y) - y g_{m-1}(x, y) \\ g_m(x, y) &= x g_{m-1}(x, y) + y f_{m-1}(x, y) \end{aligned}$$

It is immediate from the formulas that $f_m(x, y)$ will be homogeneous provided that both $f_{m-1}(x, y)$ and $g_{m-1}(x, y)$ are homogeneous of the same degree. To conclude the proof we need to check that the degree of $f_m(x, y)$ and $g_m(x, y)$ is m . It is easy to show by induction that one of the terms of $f_m(x, y)$ is x^m . Moreover, it is also clear that one of the terms of $g_m(x, y)$ is $m x^{m-1} y$. \square

It is now clear that:

$$\begin{aligned} \varphi(x, y) &\equiv \exists s : s \geq 0 \wedge \\ &\bigwedge_{i=1}^n x_i = \sum_{k=1}^n \psi_{ik}^a(y) f_{|r_k|}(\cos s, \text{sign}(r_k) \sin s) + \psi_{ik}^b(y) g_{|r_k|}(\cos s, \text{sign}(r_k) \sin s) \end{aligned}$$

where $f_{|r_k|}$ and $g_{|r_k|}$ are the polynomials given in the previous proposition. Due to the periodicity of both \sin and \cos we have that

$$\begin{aligned} \varphi(x, y) &\equiv \varphi_{\mathbb{Z}}(x, y) \\ &\equiv \exists s : 0 \leq s \leq 2\pi \wedge \\ &\bigwedge_{i=1}^n x_i = \sum_{k=1}^n \psi_{ik}^a(y) f_{|r_k|}(\cos s, \text{sign}(r_k) \sin s) + \psi_{ik}^b(y) g_{|r_k|}(\cos s, \text{sign}(r_k) \sin s) \end{aligned}$$

Restricting s to a bounded interval (in this case $[0, 2\pi]$) is extremely important as this makes the above formula definable in the o-minimal theory $\text{OF}_{an}(\mathbb{R})$. We define now a new formula:

$$\begin{aligned} \zeta(x, y) &\stackrel{\Delta}{=} \exists z_1 \exists z_2 : z_1^2 + z_2^2 = 1 \wedge \\ &\bigwedge_{i=1}^n x_i = \sum_{k=1}^n \psi_{ik}^a(y) f_{|r_k|}(z_1, \text{sign}(r_k) z_2) + \psi_{ik}^b(y) g_{|r_k|}(z_1, \text{sign}(r_k) z_2) \end{aligned} \quad (4.24)$$

Lemma 4.27. *The formulas $\varphi(x, y)$ and $\zeta(x, y)$ are equivalent.*

Proof. The equivalence is shown by setting up $z_1 = \cos s$, $z_2 = \sin s$. \square

The combination of the above lemmas give the main proposition which shows the desired decidability result.

Proposition 4.28. *Let $F(x) = Ax$ be a linear vector field and $A \in \mathbb{Q}^{n \times n}$ a matrix with pure imaginary eigenvalues of the form ir with $r \in \mathbb{Q}$, and $Y \subseteq \mathbb{R}^n$ definable in $OF(\mathcal{R})$. Then $Pre(Y)$ is definable in $OF(\mathcal{R})$.*

Proof. By the previous lemmas we have that $\eta(x) \equiv \exists y : P(y) \wedge \zeta(x, y)$ with $\zeta(x, y)$ definable in $OF(\mathcal{R})$. \square

Proposition 4.28 implies that we have a computational procedure for the reachability problem of linear vector fields with pure imaginary eigenvalues of the form ir with $r \in \mathbb{Q}$.

Example 16. Consider the linear vector field defined by

$$\begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (4.25)$$

and let $Y = \{(y_1, y_2) \in \mathbb{R}^2 \mid y_1 = 4 \wedge y_2 = 3\}$. We have that:

$$\begin{aligned} \eta(x_1, x_2) &\equiv \exists y_1 \exists y_2 \exists t : y_1 = 4 \wedge y_2 = 3 \wedge t \geq 0 \\ &\quad \wedge x_1 = y_1 \cos t - y_2 \sin t \wedge x_2 = y_2 \cos t + y_1 \sin t \\ &\equiv \exists y_1 \exists y_2 \exists z_1 \exists z_2 : y_1 = 4 \wedge y_2 = 3 \wedge z_1^2 + z_2^2 = 1 \\ &\quad \wedge x_1 = y_1 z_2 - y_2 z_1 \wedge x_2 = y_2 z_2 + y_1 z_1 \\ &\equiv x_1^2 + x_2^2 - 25 = 0 \end{aligned}$$

The above three classes of linear vector fields for which $Pre(Y)$ can be computed, immediately lead to the following constructive theorem.

Theorem 4.29 (Semidecidable Linear Hybrid Systems). *Let H be a linear hybrid system where for each discrete location $q \in X_D$ the vector field is of the form $F(q, x) = Ax$ where*

- $A \in \mathbb{Q}^{n \times n}$ is nilpotent or
- $A \in \mathbb{Q}^{n \times n}$ is diagonalizable with rational eigenvalues or
- $A \in \mathbb{Q}^{n \times n}$ has pure imaginary eigenvalues of the form ir , $r \in \mathbb{Q}$.

Then the reachability problem for H is semidecidable.

Thus, the bisimulation algorithm could be implemented for the above class of linear hybrid systems without guarantee that it would ever terminate. If it happens that the algorithm terminates, then we can compute the reachable regions of the hybrid system. In fact, Theorem 4.29 can be upgraded easily to include more complicated discrete transitions, as long as there is a constructive method to compute $Pre_e(Y)$ for any discrete transition e .

We can now combine the semidecision result of Theorem 4.29 and the termination result of Theorem 4.12 in order to obtain the desired decidability result.

Theorem 4.30 (Decidable Linear Hybrid Systems). *Let H be a linear hybrid system where for each discrete location $q \in X_D$ the vector field is of the form $F(q, x) = Ax$ where*

- $A \in \mathbb{Q}^{n \times n}$ is nilpotent or
- $A \in \mathbb{Q}^{n \times n}$ is diagonalizable with rational eigenvalues or
- $A \in \mathbb{Q}^{n \times n}$ has purely imaginary eigenvalues of the form ir , $r \in \mathbb{Q}$.

Then the reachability problem for H is decidable.

Proof. All relevant sets of linear hybrid systems are by definition definable in $\text{OF}(\mathbb{R})$ and the flows of linear vector fields are complete. Therefore, given the semidecision result of Theorem 4.29, all we have to show is that the flow of the linear vector field Ax is definable in an o-minimal theory. Then Theorem 4.12 would guarantee termination of the bisimulation algorithm. If A is nilpotent then the flow is also definable in $\text{OF}(\mathbb{R})$ which is o-minimal. If A is diagonalizable then the flow is definable in $\text{OF}_{\text{exp}}(\mathbb{R})$ which is also o-minimal. If A has purely imaginary eigenvalues, then the flow contains the functions \sin and \cos which are not definable in any of the o-minimal theories of Table 1. However, o-minimality of the flow is only used in the proof of Theorem 4.12 to show o-minimality of the Pre operator. Even though the flow of this vector field is not definable, the Pre operator corresponding to these periodic flows is still definable, as all we need is the restriction of \sin and \cos on $[0, 2\pi]$. These restrictions are indeed definable in $\text{OF}_{\text{an}}(\mathbb{R})$ which is also o-minimal. \square

Theorem 4.30 is the first decidability result in the area of hybrid systems that provides the modeling expressiveness to capture relatively complex continuous dynamics. The importance of these results is immediate given the wide application of (piecewise) linear systems in control theory. In addition, Theorem 4.30 contains in it a purely continuous version of reachability analysis for linear systems under state constraints, a problem which

is known to be very difficult. As a result, its potential application to analyze various realistic hybrid systems using computational methods is significant.

4.7 Conclusions

This chapter presented a unified framework for obtaining classes of hybrid systems with a decidable reachability problem. Decidability requires both the termination and computability of the well known bisimulation algorithm. Termination of the algorithm was guaranteed for \mathcal{O} -minimal hybrid systems which are initialized hybrid systems whose relevant sets and flows are definable in an \mathcal{O} -minimal theory. Various examples from recently discovered \mathcal{O} -minimal theories were presented. The search for computable subclasses within \mathcal{O} -minimal theories leads to new decidable classes of hybrid systems. This resulted in classes of hybrid systems with linear vector fields in each discrete location having a decidable reachability problem.

Even though decidability may guarantee termination of an algorithm, the complexity of the algorithm may be extremely expensive. Useful algorithms must be applicable to systems of large scale and complexity. One of the main tools in tackling complexity is abstraction, or extracting simple models from complex ones while retaining all relevant information of interest. The next chapter develops a theory of abstraction for reachability properties of continuous systems. A theory of abstraction of continuous systems will also be extremely useful in understanding and designing large scale, hierarchical systems which utilize a hierarchy of models at various levels of abstraction.

Chapter 5

Abstractions of Control Systems

In order to tackle the complexity involved in verifying that a given large scale system satisfies certain properties, one tries to extract a simpler but qualitatively equivalent system, called an *abstraction*. Checking the desired property on the abstracted system should be *equivalent* or *sufficient* to checking the property on the original system. The area of computer aided verification, which must be credited with this notion of abstraction, typically faces problems of exponential complexity and abstractions are frequently used for complexity reduction [31, 49, 66, 91]. Depending on the property, special graph quotients which preserve the property of interest are constructed. Bisimulations, the topic of the Chapter 4, is an example of such a special abstraction.

In addition to analysis, modeling abstractions are also useful in hierarchical control. Large scale systems such as automated highway systems [109] and air traffic management systems [88] are systems of very high complexity. Complexity is typically reduced by imposing a hierarchical structure on the system architecture. Figure 5.1 shows a typical two-layer control hierarchy which is frequently used in the quite common planning and control hierarchical systems. Multi-layered versions of Figure 5.1 are used in both [88] and [109]. In this layered control paradigm, each layer has different objectives. In performing their tasks, the higher level uses a coarser system model than the lower level. One of the main challenges in hierarchical systems is the extraction of a hierarchy of models at various levels of abstraction which are compatible with the functionality and objectives of each layer.

In the literature, the notions of *abstraction* or *aggregation* refer to grouping the system states into equivalence classes. Depending on the cardinality of the resulting quotient space we may have *discrete* or *continuous* abstractions. With this notion of abstraction, the

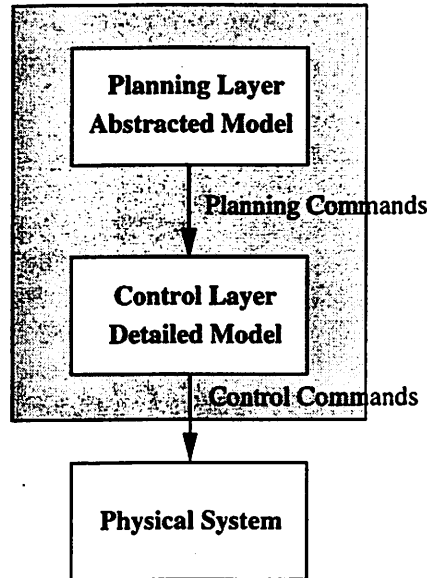


Figure 5.1: Two layer control hierarchy

abstracted system will be defined as the induced quotient dynamics. Discrete abstractions of continuous systems have been considered in [24, 26] as well as [8, 32, 93]. Discrete abstractions of hybrid systems were the main topic of Chapter 4. Hierarchical systems for discrete event systems have been formally considered in [25, 113, 114, 119]. In this chapter, we focus on *continuous abstractions* of continuous systems. Therefore, our first priority is to have a formal notion of quotient control systems. More precisely,

Problem 5.1. *Given a control system*

$$\dot{x} = f(x, u) \quad x \in \mathbb{R}^n \quad u \in \mathbb{R}^m \quad (5.1)$$

and some map $y = \Phi(x)$, where $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^p$, we would like to define a control system

$$\dot{y} = g(y, v) \quad y \in \mathbb{R}^p \quad v \in \mathbb{R}^k \quad (5.2)$$

which can produce as trajectories all functions of the form $y(t) = \Phi(x(t))$, where $x(t)$ is a trajectory of system (5.1). That is, Φ maps trajectories of system (5.1) to trajectories of system (5.2).

The function Φ is the “quotient map” which performs the state aggregation. System (5.2) will be referred to as the *abstraction* [86] or *macromodel* of the finer *micromodel*

(5.1). Note that the control input v of the coarser model (5.2) is not the same input u of system (5.1) and should be thought of as a macroinput. For example, v can be velocity inputs of a kinematic model whereas u may be force and torque inputs of a dynamic model. This is therefore quite different from model reduction techniques which reduce or aggregate dynamics while using the same control inputs [9, 57, 58, 59, 60].

We will solve Problem 5.1 by first generalizing the geometric notion of Φ -related vector fields to control systems. A notion of Φ -related control systems would allow us to push forward control systems through quotient maps and obtain well defined control systems describing the aggregate dynamics. The notion of Φ -related control systems introduced in this paper is more general than the notion of projectable systems defined in [60] and [77] as we will show that given any control system and any surjective map Φ , there always exists another system that is Φ -related to it. Our notion of Φ -related control systems mathematically formalizes the concept of *virtual inputs* used in backstepping designs [56]. The fact that the aggregation map sends trajectories of (5.1) to trajectories of (5.2) will enable us to propagate controllability from the micromodel to the macromodel.

Aggregation, however, is not independent of the functionality of the layer at which the abstracted system will be used. Therefore, when an abstracted model is extracted from a more detailed model, one would also like to ensure that certain properties propagate from the macromodel to the micromodel. The properties that are of interest at each layer may include optimality, controllability, stabilizability, and trajectory tracking. If one considers the property of controllability, then one would like to determine conditions under which controllability of the abstracted system (5.2) implies controllability of system (5.1). Obtaining such conditions would ensure that the macromodel is a *consistent abstraction* of the micromodel in the sense that controllability requests from the macromodel are *implementable* by the micromodel. Such conditions will serve as good design principles for hierarchical control systems. Different properties may require different conditions. For example, the notions of consistency [78], dynamic consistency [25] and hierarchical consistency [119] have been defined in order to ensure feasible execution of high level objectives for discrete event systems. In this chapter, we will focus on controllability of linear control systems and characterize consistent linear abstractions. More precisely, we will solve the following problem:

Problem 5.2. *Given the linear control system*

$$\dot{x} = Ax + Bu \quad x \in \mathbb{R}^n \quad u \in \mathbb{R}^m \quad (5.3)$$

characterize linear quotient maps $y = Cx$, so that the abstracted linear system

$$\dot{y} = Fy + Gv \quad y \in \mathbb{R}^p \quad v \in \mathbb{R}^k \quad (5.4)$$

is controllable if and only if system (5.3) is controllable.

After having characterized consistent linear abstractions, we obtain a hierarchical controllability criterion which has computational and conceptual advantages over the Kalman rank condition and the Popov-Belevitch-Hautus (PBH) tests for large scale systems. Intuitively, instead of checking controllability of a large scale system, we construct a sequence of consistent abstractions and then check the controllability of a system which is much smaller in size. Consistency will then propagate controllability along this sequence of abstractions from the simpler quotient system to the original complex system. The computational advantages of this approach are verified by recovering the best of the known controllability algorithms from numerical linear algebra [39] as a special case of the hierarchical controllability criterion.

5.1 Abstractions of Vector Fields

In this section, a notion of an abstraction for a dynamical system or vector field is introduced. Consider a vector field X on a manifold M , the state space of the system. In abstracting system dynamics, information about the state of the system which is not useful in the analysis process is usually ignored in order to produce a simplified model of reduced complexity. For example, each state could be mapped to part of the state or to certain outputs of interest. What state information is relevant usually depends on the properties which need to be satisfied. Our goal is to try to obtain another dynamical system or vector field which describe the evolution of the dynamics of interest.

The system state $p \in M$ is thus mapped to an abstracted state $q \in N$ by some aggregation or abstraction map $\Phi : M \rightarrow N$. This map, which we will assume from now on to be surjective, groups the states in a very simple way: states p_1 and p_2 on M are equivalent if $\Phi(p_1) = \Phi(p_2)$. In order for the quotient space to have a manifold structure, the equivalence relation must be regular [1].

Once a map Φ has been given, then given a vector field X which governs the state evolution on M , one is interested in obtaining the evolution of the abstracted dynamics. The evolution of a dynamical system is characterized by its integral curves. Let c be any

integral curve of X . Then if we push forward the curve c by the map Φ we obtain that $\Phi(c)$ describes the evolution of the abstracted dynamics on N . If we therefore want to abstract the vector field X on M by a vector field Y on N , then $\Phi(c)$ should be an integral curve of Y . This motivates the following definition.

Definition 5.3 (Abstractions of Dynamical Systems). *Let X and Y be vector fields on M and N respectively and let $\Phi : M \rightarrow N$ be a smooth surjective map. Then vector field Y is an abstraction of vector field X with respect to Φ iff for every integral curve c of X , $\Phi \circ c$ is an integral curve of Y .*

Therefore if the integral curve c satisfies

$$c' = c_*(1) = X(c)$$

then it must also be true that

$$(\Phi \circ c)' = (\Phi \circ c)_*(1) = Y(\Phi \circ c)$$

Therefore, if Σ_X and Σ_Y denote all integral curves of vector fields X and Y respectively, then vector field Y overapproximates the collection of curves $\Phi(\Sigma_X)$ and allows redundant evolutions. Then, instead of checking reachability of vector field X , it is *sufficient* to check it on Y , which is of smaller dimension.

From Definition 5.3 it is clear that a vector field Y may be an abstraction of some vector field X for some map Φ_1 , but may not be for another map Φ_2 . In building hierarchical models of large scale systems, the system may be modeled at many levels of abstraction. The following proposition shows that abstracting dynamical systems is transitive.

Proposition 5.4 (Transitivity of Abstractions). *Let X_1, X_2, X_3 be vector fields on manifolds M_1, M_2 and M_3 respectively. If X_2 is an abstraction of X_1 with respect to the map $\Phi_1 : M_1 \rightarrow M_2$ and X_3 is an abstraction of X_2 with respect to map $\Phi_2 : M_2 \rightarrow M_3$ then X_3 is an abstraction of X_1 with respect to map $\Phi_2 \circ \Phi_1$.*

Proof. Let c be any integral curve of X_1 . Since X_2 is an abstraction of X_1 , then by definition $\Phi_1(c)$ is an integral curve of X_2 . But since X_3 is an abstraction of X_2 , $\Phi_2(\Phi_1(c)) = (\Phi_2 \circ \Phi_1)(c)$ is an integral curve of X_3 . Thus for any integral curve c of X_1 , $(\Phi_2 \circ \Phi_1)(c)$ is an integral curve of X_3 . Thus X_3 is an abstraction of X_1 with respect to abstracting map $\Phi_2 \circ \Phi_1$. \square

Definition 5.3 is not an easily checkable condition since it involves integral curves of vector fields. The following theorem shows that Definition 5.3 is equivalent to saying that the two vector fields are Φ -related.

Theorem 5.5 (Abstracted Vector Fields are Φ -Related). *Vector field Y on N is an abstraction of vector field X on M with respect to the map Φ if and only if X and Y are Φ -related.*

Proof. Let vector field Y on N be an abstraction with respect to Φ of vector field X on M . Then by Definition 5.3, for any integral curve c of X , $\Phi \circ c$ is an integral curve of Y . Thus

$$\begin{aligned} (\Phi \circ c)' &= (\Phi \circ c)_*(1) = Y(\Phi \circ c) \Rightarrow \\ \Phi_* \circ c_*(1) &= Y \circ \Phi \circ c \Rightarrow \\ \Phi_* \circ X(c) &= Y \circ \Phi \circ c \Rightarrow \\ \Phi_* \circ X \circ c &= Y \circ \Phi \circ c \Rightarrow \\ \Phi_* \circ X &= Y \circ \Phi \end{aligned}$$

But then, by Definition 2.4, X and Y are Φ -related. Conversely, let X and Y be Φ related. Then for any integral curve c of X ,

$$\begin{aligned} \Phi_* \circ X &= Y \circ \Phi \Rightarrow \\ \Phi_* \circ X \circ c &= Y \circ \Phi \circ c \Rightarrow \\ \Phi_* \circ X(c) &= Y(\Phi \circ c) \Rightarrow \\ \Phi_* \circ c_*(1) &= Y(\Phi \circ c) \Rightarrow \\ (\Phi \circ c)_*(1) &= Y(\Phi \circ c) \end{aligned}$$

and thus $\Phi \circ c$ is an integral curve of Y . Therefore Y is an abstraction of vector field X with respect to Φ . \square

Theorem 5.5 allows us to check a condition on the vector fields rather than explicitly computing integral curves and verifying Definition 5.3.

Example 17. Consider for example the linear vector field

$$\dot{x} = Ax \quad x \in \mathbb{R}^n \tag{5.5}$$

and the linear, onto quotient map $y = Cx$. Then in order to obtain a well defined quotient vector field,

$$\dot{y} = Fy \quad y \in \mathbb{R}^m \quad (5.6)$$

by C -relatedness we must have $CAx = FCx$ for all $x \in \mathbb{R}^n$. But for $x \in \text{Ker}(C) = \{x \in \mathbb{R}^n \mid Cx = 0\}$ we must have $CAx = F(Cx) = 0$ and thus $Ax \in \text{Ker}(C)$. Thus, a necessary condition to obtain a well defined quotient vector field is

$$A\text{Ker}(C) \subseteq \text{Ker}(C) \quad (5.7)$$

It turns out that this is also sufficient for the existence of a unique quotient vector field [116].

As can be seen from Theorem 5.5 and Example 17, Φ -relatedness of two vector fields is a very restrictive condition which limits the cases where one dynamical system is an exact abstraction of another. Even though Φ -relatedness of vector fields is a rather restrictive condition, the above discussion provides the correct conceptual framework for generalizing these concepts to control systems, where due to the freedom of control inputs the equivalent conditions will not be as restrictive.

5.2 Control System Abstractions

In this section, the notions of Section 2.1 for vector fields are extended to control systems. We will develop such notions for rather general control systems. Generality will ensure that the concepts of this section do not depend on the particular system structure. We first present a global and coordinate-free description of control systems which is due to Brockett [21, 22] and can also be found in [82]. This global description is based on the notion of fiber bundles which were defined in Section 2.1.

Definition 5.6 (Control Systems). *A control system $S = (B, F)$ consists of a fiber bundle $\pi : B \rightarrow M$ called the control bundle and a smooth map $F : B \rightarrow TM$ which is fiber preserving, that is $\pi' \circ F = \pi$ where $\pi' : TM \rightarrow M$ is the tangent bundle projection.*

Essentially, the base manifold M of the control bundle is the state space and the fibers $\pi^{-1}(p)$ can be thought of as the state dependent control spaces. Given the state p and the input, the map F selects a tangent vector from T_pM . The notion of trajectories of control systems is now defined.

Definition 5.7 (Trajectories of Control Systems). A smooth curve $c : I \rightarrow M$ is called a trajectory of the control system $S = (B, F)$ if there exists a curve $c^B : I \rightarrow B$ satisfying

$$\begin{aligned}\pi \circ c^B &= c \\ c' &= c_*(1) = F \circ c^B\end{aligned}$$

In local (bundle) coordinates, Definition 5.7 simply says that a trajectory of a control system is a curve $x : I \rightarrow M$ for which there exists a function $u : I \rightarrow U$ satisfying, satisfying $\dot{x} = F(x, u)$. Note that even though Definition 5.7 assumes c to be smooth, the bundle curve c^B is not necessarily smooth. The definition therefore allows nonsmooth control inputs as long as the projection $\pi \circ c^B = c$ is smooth.

Recall that for vector fields, the notion of abstraction was equivalent to the notion of Φ -related vector fields. We now define Φ -related control systems in a manner similar to Definition 2.4 for vector fields.

Definition 5.8 (Φ -Related Control Systems). Let $S_M = (B_M, F_M)$ with $\pi_M : B_M \rightarrow M$ and $S_N = (B_N, F_N)$ with $\pi_N : B_N \rightarrow N$ be two control systems. Let $\Phi : M \rightarrow N$ be a smooth map. Then control systems S_M and S_N are Φ -related iff for every $p \in M$

$$\Phi_* \circ F_M (\pi_M^{-1}(p)) \subseteq F_N (\pi_N^{-1}(\Phi(p))) \quad (5.8)$$

Control system S_N will be referred to as an *abstraction* of control system S_M ([86]). Condition (5.8) states that for each $p \in M$ the left hand side of (5.8) first takes the input space available at p , and pushes it through F_M to obtain all possible tangent directions of the control system S_M at p . This set of tangent directions is pushed through Φ_* to obtain a set of tangent vectors in $T_{\Phi(p)}N$. In order for S_M and S_N to be Φ -related, this set must be contained in the image under F_N of the input space available at $\Phi(p)$. Note that many control systems S_N may be Φ -related to S_M as the set of tangent vectors on N that must be captured, can be generated using many control parameterizations.

In a manner similar to Proposition 5.4, it is easy to show that Φ -relatedness is transitive. Indeed, if $\Phi_1 : M_1 \rightarrow M_2$, $\Phi_2 : M_2 \rightarrow M_3$, S_{M_1} is Φ_1 -related to S_{M_2} , and S_{M_2} is Φ_2 -related to S_{M_3} , then S_{M_1} is $\Phi_2 \circ \Phi_1$ -related to S_{M_3} . It therefore makes sense to consider a sequence of Φ -related systems. In addition, given M, N , a map $\Phi : M \rightarrow N$ and a system S_M , one can put a partial order on all possible Φ -related systems S_N , where the partial ordering arises from pointwise subset inclusion of the right hand side of (5.8) (see [86]).

To see that Definition 5.8 is a generalization of Definition 2.1, consider vector fields X_M on M and X_N on N . Then X_M and X_N can be thought of as trivial control systems on M and N respectively by letting $B_M = M$, $B_N = N$, $\pi_M = id_M$, $\pi_N = id_N$, and $F_M = X_M$, $F_N = X_N$. Condition (5.8) becomes $\Phi_* \circ X_M(p) = X_N \circ \Phi(p)$, which is Definition 2.1 of Φ -related vector fields.

The following proposition, which is an immediate consequence of Definition 5.8, shows that every control or dynamical system is Φ -related to some control system for any map Φ .

Proposition 5.9 (Existence of Abstractions). *Given any control system $S_M = (B_M, F_M)$ and any smooth map $\Phi : M \rightarrow N$, then there exists a control system $S_N = (B_N, F_N)$ which is Φ -related to S_M . In particular, every vector field X on M is Φ -related to some control system S_N .*

Proof. Given S_M , construct S_N by simply letting $B_N = TN$ and $F_N : TN \rightarrow TN$ equal the identity. Then condition (5.8) is trivially satisfied. Thus $S_N = (B_N, F_N)$ is Φ -related to S_M . \square

In local coordinates, Proposition 5.9 simply states that the push forward of a control system or a vector field is a differential inclusion which can be thought of as another control system. Even though Proposition 5.9 is a simple existential result, it is important as it shows that given any control system and any aggregation map, then an abstracted control system always exists.

The concept of Φ -related control systems is a generalization of the notion of *projectable* control systems defined in [60, 77]. A control system is projectable, essentially, when each vector field corresponding to a fixed input value is Φ -related to some vector field. Definition 5.8, instead of globally pushing a vector field for each fixed value of the control input, takes a pointwise approach by pushing forward all possible tangent directions at a state for all possible inputs available at that state. By Proposition 5.9, any projectable system in the sense of [60, 77] is also Φ -related in the sense of Definition 5.8. The following example illustrates that Φ -related control systems are not necessarily projectable.

Example 18. Consider the double integrator

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= u\end{aligned}$$

with $x_1, x_2, u \in \mathbb{R}$ and the projection $\Phi(x_1, x_2) = x_1$. Using Definition 5.8, we obtain that

$$\dot{x}_1 = x_2$$

is a valid Φ -related system. The double integrator, however, is not projectable in the sense of [77, 60] with respect to this map as for any fixed value of u , the vector field $[x_2 \ u]^T$ is not Φ -related to any vector field on \mathbb{R} . For the nonlinear control system,

$$\begin{aligned}\dot{x}_1 &= f_1(x_1, x_2) \\ \dot{x}_2 &= f_2(x_1, x_2, u)\end{aligned}$$

with states x_1, x_2 , input u , and the projection $\Phi(x_1, x_2) = x_1$, a Φ -related system is

$$\dot{x}_1 = f_1(x_1, x_2)$$

with state x_1 but where x_2 is now thought of as an input. This is the notion of virtual inputs used in backstepping designs [56]. A more constructive methodology for generating abstractions of linear systems will be presented in Section 5.4.

The following theorem should be thought of as a generalization of Theorem 5.5 for control systems. It establishes the connection between trajectories of Φ -related control systems.

Theorem 5.10 (Trajectories of Φ -Related Control Systems). *Let $S_N = (B_N, F_N)$ and $S_M = (B_M, F_M)$ be two control systems and $\Phi : M \rightarrow N$ be a smooth map. Then S_M and S_N are Φ -related if and only if for every trajectory c_M of S_M , $\Phi \circ c_M$ is a trajectory of S_N .*

Proof. (Sufficiency) Assume that S_M and S_N are Φ -related and thus for all $p \in M$ we have

$$\Phi_* \circ F_M(\pi_M^{-1}(p)) \subseteq F_N(\pi_N^{-1}(\Phi(p))) \quad (5.9)$$

Let $c_M : I \rightarrow M$ be any trajectory of S_M . We must show that $\Phi \circ c_M$ is a trajectory of S_N . We must therefore find a curve $c_N^B : I \rightarrow B_N$ such that for all $t \in I$ we have $\pi_N \circ c_N^B(t) = \Phi \circ c_M(t)$ and $(\Phi \circ c_M)'(t) = F_N \circ c_N^B(t)$.

Since $c_M : I \rightarrow M$ is a trajectory of S_M , by Definition 5.7 there exists a curve $c_M^B : I \rightarrow B_M$ such that for all $t \in I$ we have $\pi_M \circ c_M^B(t) = c_M(t)$ and $c_M'(t) = F_M \circ c_M^B(t)$.

By Φ -relatedness of S_M and S_N we obtain that for all $t \in I$,

$$\begin{aligned} \Phi_* \circ F_M (\pi_M^{-1}(c_M(t))) &\subseteq F_N (\pi_N^{-1}(\Phi(c_M(t)))) \implies \\ \Phi_* \circ F_M \circ c_M^B(t) &\in F_N (\pi_N^{-1}(\Phi(c_M(t)))) \end{aligned} \quad (5.10)$$

Condition (5.10) implies that for each $t \in I$ there must exist at least one element $c_N^B(t) \in \pi_N^{-1}(\Phi(c_M(t)))$ (and thus $\pi_N \circ c_N^B(t) = \Phi \circ c_M(t)$) such that

$$\begin{aligned} \Phi_* \circ F_M \circ c_M^B(t) &= F_N \circ c_N^B(t) \\ \Phi_* \circ c'_M(t) &= F_N \circ c'_N(t) \\ (\Phi \circ c_M)'(t) &= F_N \circ c'_N(t) \end{aligned}$$

Therefore $\Phi \circ c_M$ is a trajectory of S_N .

(Necessity) Assume that for every trajectory $c_M : I \rightarrow M$ of S_M , $\Phi \circ c_M$ is a trajectory of S_N . Now for any point $p \in M$ let

$$Y_{\Phi(p)} \in \Phi_* (F_M(\pi_M^{-1}(p))) \quad (5.11)$$

We must show that $Y_{\Phi(p)} \in F_N(\pi_N^{-1}(\Phi(p)))$. We can write $Y_{\Phi(p)} = \Phi_*(X_p)$ for some (not necessarily unique) tangent vector $X_p \in F_M(\pi_M^{-1}(p))$. Then there exists a trajectory $c_M : I \rightarrow M$ such that at some $t^* \in I$ we have

$$c_M(t^*) = p \quad (5.12)$$

$$c'_M(t^*) = X_p \quad (5.13)$$

Indeed, a curve c_M satisfying (5.12,5.13) always exists by the existence theorems for differential equations. To show that c_M is a trajectory, we need to find $c_M^B : I \rightarrow B_M$ such that $\pi \circ c_M^B = c_M$. Let O be a bundle trivializing neighborhood of p and $\Psi : \pi^{-1}(O) \rightarrow O \times U$ the trivializing map. There exists $u \in U$ such that $X_p = F_M \circ \Psi^{-1}(p, u)$. Restricting I if necessary we may assume $c_M(I) \subset O$. We can then define the desired curve by $c_M^B(t) = F_M \circ \Psi^{-1}(c_M(t), u)$.

Since c_M is a trajectory of S_M satisfying (5.12,5.13), then by assumption we have that $\Phi \circ c_M$ is a trajectory of S_N . Therefore by Definition 5.7, there must exist a curve $c_N^B : I \rightarrow B_N$ such that for all $t \in I$ we have $\pi_N \circ c_N^B(t) = \Phi \circ c_M(t)$ and $(\Phi \circ c_M)'(t) =$

$F_N \circ c_N^B(t)$. In particular, at $t^* \in I$ we have

$$\begin{aligned} (\Phi \circ c_M)'(t^*) &= F_N \circ c_N^B(t^*) \\ \Phi_* \circ c_M'(t^*) &\in F_N(\pi_N^{-1}(\Phi(c_M(t^*)))) \\ Y_p = \Phi_*(X_p) &\in F_N(\pi_N^{-1}(\Phi(p))) \end{aligned}$$

Therefore, at all points $p \in M$ we must have $\Phi_* \circ F_M(\pi_M^{-1}(p)) \subseteq F_N(\pi_N^{-1}(\Phi(p)))$ and thus S_M and S_N are Φ -related. This completes the proof. \square

If Σ_{S_M} and Σ_{S_N} denote all trajectories of control systems S_M and S_N respectively, then Theorem 5.10 simply states that S_M and S_N are Φ -related if and only if $\Phi(\Sigma_{S_M}) \subseteq \Sigma_{S_N}$. The quotient system therefore overapproximates the abstracted trajectories of the original system which may result in trajectories that the macrosystem S_N may generate but are infeasible in the micromodel S_M .

Theorem 5.10 does not guarantee that the curve $c_N^B(t)$ is a smooth curve. The main obstacle for generating smooth $c_N^B(t)$ is whether the map $F_N : B_N \rightarrow TN$ is an embedding. An example showing that F_N being only an immersion is not enough can be found in [85]. The following theorem shows that F_N being an injective embedding is sufficient to guarantee smoothness of the $c_N^B(t)$. Note that requiring F_N to be an injective embedding implies that the dimension of the input space is less than the dimension of TN and thus there are no redundant inputs (which covers the cases of interest). In particular, if the control system S_N is affine in the controls then this is equivalent to saying that the “controlled” vector fields are linearly independent at each point. That is, if we write the system in local (bundle) coordinates of B_N and local (vector bundle) coordinates of TN as

$$\dot{x} = f(x) + \sum_{i=1}^k g_i(x)u_i$$

then for each x the vectors $g_1(x), \dots, g_k(x)$ are linearly independent.

Theorem 5.11 (Control Input Smoothness). *Let $S_N = (B_N, F_N)$ and $S_M = (B_M, F_M)$ be two Φ -related control systems where $F_N : B_N \rightarrow TN$ is an injective embedding. Let $c_M : I \rightarrow M$ be a trajectory of S_M and assume that the corresponding $c_M^B : I \rightarrow B_M$ is smooth. Then there exists a smooth curve $c_N^B : I \rightarrow B_N$ such that for all $t \in I$, $\pi_N \circ c_N^B(t) = \Phi \circ c_M(t)$ and $F_N \circ c_N^B(t) = (\Phi \circ c_M)'(t)$.*

Proof. Since S_M and S_N are Φ -related we have $\Phi_* \circ F_M \circ c_M^B(t) \in F_N(\pi_N^{-1}(\Phi(c_M(t))))$ for each $t \in I$. Moreover, since by assumption F_N is an embedding, the space B_N is diffeomorphic to its image under F_N . We can then define

$$c_N^B(t) = F_N^{-1}(\Phi_* \circ F_M \circ c_M^B(t))$$

which is clearly smooth and satisfies the desired properties. \square

5.3 Consistent Control Abstractions

In general, we are not simply interested in abstracting systems but also propagating properties between the original and abstracted model. In particular, we shall focus on various notions of controllability.

Definition 5.12 (Controllability). *Let $S = (B, F)$ be a control system on M . For $p \in M$, define $Reach(p, S)$ to be the set of points $q \in M$ for which there exists a trajectory $c : I \rightarrow M$ of S such that for some $t_1, t_2 \in I$ we have $c(t_1) = p$ and $c(t_2) = q$. The control system S is called controllable iff for all $p \in M$, $Reach(p, S) = M$.*

Theorem 5.10 allows us to always propagate the property of controllability from the micromodel to the macromodel for any aggregation map.

Theorem 5.13 (Controllability Propagation). *Let control systems $S_M = (B_M, F_M)$ and $S_N = (B_N, F_N)$ be Φ -related with respect to some smooth surjection $\Phi : M \rightarrow N$. Then for all $p \in M$,*

$$\Phi(Reach(p, S_M)) \subseteq Reach(\Phi(p), S_N)$$

Thus, if S_M is controllable then S_N is controllable.

Proof. Consider any $p \in M$ and let $q \in \Phi(Reach(p, S_M))$. Then there exists $p_1 \in \Phi^{-1}(q)$ with $p_1 \in Reach(p, S_M)$. Thus there exists a trajectory c_M of S_M such that $c_M(t_1) = p$ and $c_M(t_2) = p_1$. By Φ -relatedness, the curve $\Phi \circ c_M$ is a trajectory of S_N which connects $\Phi(c_M(t_1)) = \Phi(p)$ and $\Phi(c_M(t_2)) = \Phi(p_1) = q$. Therefore $q \in Reach(\Phi(p), S_N)$.

If S_M is controllable, then for all $p \in M$ we have $Reach(p, S_M) = M$. But then $\Phi(Reach(p, S_M)) = \Phi(M) = N = Reach(\Phi(p), S_N)$. Thus S_N is controllable. \square

Note that Theorem 5.13 is true regardless of the structure of the aggregation map Φ . From a hierarchical perspective, the reverse question is a lot more interesting since it

would guarantee that controllability requests are implementable by the lower level system. In order to arrive at this goal, we define the notions of implementability and consistency. We also give descriptions of those concepts in terms of reachable sets.

Definition 5.14 (Controllability Implementation). *Let $S_M = (B_M, F_M)$ and $S_N = (B_N, F_N)$ be two control systems and $\Phi : M \rightarrow N$ be a smooth surjection. Then S_N is implementable¹ by S_M iff whenever there is a trajectory of S_N connecting $q_1 \in N$ and $q_2 \in N$, then there exist $p_1 \in \Phi^{-1}(q_1)$ and $p_2 \in \Phi^{-1}(q_2)$ and a trajectory of S_M connecting p_1 and p_2 .*

Implementability is therefore an existential property. If one thinks of the map Φ as a quotient map, then implementability requires that a reachability request is implementable by at least one member of the equivalence class. It is clear from Definition 5.14 that implementability is transitive, that is if S_{M_1} is implementable by S_{M_2} with respect to Φ_1 , and S_{M_2} is implementable by S_{M_3} with respect to Φ_2 , then S_{M_1} is implementable by S_{M_3} with respect to $\Phi_1 \circ \Phi_2$. This is important in hierarchical systems which should consist of a sequence of implementable abstractions. It should be noted that the notion of implementability defined above is related to the notion of between-block controllability, defined in [25, 26].

Proposition 5.15 (Implementation Condition). *Consider control systems $S_M = (B_M, F_M)$ and $S_N = (B_N, F_N)$ and a smooth surjection $\Phi : M \rightarrow N$. Then S_N is implementable by S_M if and only if for all $q \in N$,*

$$\text{Reach}(q, S_N) \subseteq \Phi(\text{Reach}(\Phi^{-1}(q), S_M)) \quad (5.14)$$

where $\text{Reach}(\Phi^{-1}(q), S_M) = \cup_{p \in \Phi^{-1}(q)} \text{Reach}(p, S_M)$.

Proof. Let $q' \in \text{Reach}(q, S_N)$. By implementability, there exists a trajectory of S_M connecting some $p \in \Phi^{-1}(q)$ to some $p' \in \Phi^{-1}(q')$ and thus $p' \in \text{Reach}(p, S_M)$. But then $q' = \Phi(p') \in \Phi(\text{Reach}(p, S_M)) \subseteq \Phi(\text{Reach}(\Phi^{-1}(q), S_M))$.

Conversely, let $q_2 \in \text{Reach}(q_1, S_N)$ for some $q_1 \in N$. By assumption,

$$\begin{aligned} q_2 \in \Phi(\text{Reach}(\Phi^{-1}(q_1), S_M)) &= \Phi(\cup_{p_1 \in \Phi^{-1}(q_1)} \text{Reach}(p_1, S_M)) \\ &= \cup_{p_1 \in \Phi^{-1}(q_1)} \Phi(\text{Reach}(p_1, S_M)) \end{aligned}$$

¹In this paper, we only consider implementation of controllability requests. Thus implementability will refer to controllability implementation.

But then there must exist at least one $p'_1 \in \Phi^{-1}(q_1)$ such that $q_2 \in \Phi(\text{Reach}(p'_1, S_M))$ which in turn implies that there exists $p'_2 \in \text{Reach}(p'_1, S_M)$ with $\Phi(p'_2) = q_2$ and thus S_N is implementable by S_M . This completes the proof. \square

We will mostly be interested in implementability of Φ -related systems, in which case the above inclusion becomes an equality, by Theorem 5.13.

Implementability may depend on the particular element chosen from the equivalence class $\Phi^{-1}(q)$. In order to make the controllability request well defined, it would have to be independent of the particular element chosen from the equivalence class. This leads to the important notion of consistency.

Definition 5.16 (Controllability Consistency). *Let $S_M = (B_M, F_M)$ be a control system on M and let $\Phi : M \rightarrow N$ be a smooth surjection. Then S_M is called consistent with respect to Φ whenever the following holds: if there exists a trajectory of S_M connecting p and q , then for all p' such that $\Phi(p) = \Phi(p')$ there exists a trajectory of S_M connecting p' to some q' with $\Phi(q) = \Phi(q')$.*

Note that while implementability is a condition between two systems S_M and S_N , consistency is a condition on a single system with respect to some quotient map Φ . Consistency requires that the ability to reach a particular equivalence class is independent of the chosen element from the initial equivalence class. Notice that $\Phi^{-1}(\Phi(p))$ is the equivalence class of p with respect to Φ .

Proposition 5.17 (Consistency Condition). *Consider a control system $S = (B, F)$ on M and a smooth surjection $\Phi : M \rightarrow N$. Then S is consistent with respect to Φ if and only if for all $p \in M$,*

$$\Phi(\text{Reach}(\Phi^{-1}(\Phi(p)), S)) = \Phi(\text{Reach}(p, S)). \quad (5.15)$$

Proof. Clearly $\Phi(\text{Reach}(p, S)) \subseteq \Phi(\text{Reach}(\Phi^{-1}(\Phi(p)), S))$ for any $p \in M$. Let $q = \Phi(p')$ with $p' \in \text{Reach}(\Phi^{-1}(\Phi(p)), S)$. There exists $p_0 \in \Phi^{-1}(\Phi(p))$ such that $p' \in \text{Reach}(p_0, S)$. By consistency, since $\Phi(p_0) = \Phi(p)$, there exists $p'' \in \text{Reach}(p, S)$ with $\Phi(p'') = \Phi(p')$. But then $q = \Phi(p'') \in \Phi(\text{Reach}(p, S))$.

Conversely, assume (5.15) holds. Let $q \in \text{Reach}(p, S)$ and $\Phi(p') = \Phi(p)$. Then $\Phi(q) \in \Phi(\text{Reach}(\Phi^{-1}(\Phi(p)), S)) = \Phi(\text{Reach}(p', S))$ and there exists $q' \in \text{Reach}(p', S)$ with $\Phi(q) = \Phi(q')$. \square

Consistency does not place any conditions on which element of the final equivalence class the system will be steered to. In some hierarchical systems, this may be acceptable as the high level system S_N may be interested in its command having a feasible execution by S_M without being interested about the particular state of S_M as long as it steers it to the correct equivalence class. This form of generalized output controllability is now defined.

Definition 5.18 (Macrocontrollability). *Let $S = (B, F)$ be a control system on M and let $\Phi : M \rightarrow N$ be a smooth surjection. Then S is called macrocontrollable if for all $p \in M$ and any $q \in N$ there exists an trajectory of S connecting p to some $p' \in M$ with $\Phi(p') = q$.*

By combining the notions of implementability and consistency, we can propagate some controllability information from the coarser system S_N to the more detailed system S_M .

Proposition 5.19 (Macrocontrollability Propagation). *Consider control systems $S_M = (B_M, F_M)$ and $S_N = (B_N, F_N)$ which are Φ -related with respect to the smooth surjection $\Phi : M \rightarrow N$. Assume that S_M is an implementation of S_N , and S_M is consistent. Then S_M is macrocontrollable if and only if S_N is controllable.*

Proof. Let $p \in M$ and $q \in N$ be any points. Let $q_0 = \Phi(p)$. Since S_N is controllable, there exists a trajectory of S_N connecting q_0 and q . Since S_M is an implementation of S_N , there exists a trajectory of S_M connecting some $p_1 \in \Phi^{-1}(q_0)$ and some $p_2 \in \Phi^{-1}(q)$. Moreover, since S_M is also consistent, there is a trajectory of S_M connecting p to some p' with $\Phi(p') = \Phi(p_2) = q$. Therefore, S_M is macrocontrollable. The other direction follows easily from Theorem 5.13. \square

In order to propagate full controllability from S_M to S_N , we need a stronger notion of consistency which would be independent from the elements chosen from both the initial and final equivalence class.

Definition 5.20 (Strong Controllability Consistency). *Let $S_M = (B_M, F_M)$ be a control system on M and $\Phi : M \rightarrow N$ a smooth surjection. Then S_M is called strongly consistent with respect to Φ whenever the following holds: if there exists a trajectory of S_M connecting p and q , then for all p' and for all q' such that $\Phi(p) = \Phi(p')$, $\Phi(q) = \Phi(q')$ there exists a trajectory connecting p' to q' .*

Definition 5.20 is weaker than the notion of in-block controllability of [25, 26] as it does not restrict the system to remain within the equivalence class in order to steer from one element to another in the same class.

Proposition 5.21 (Strong Consistency Condition). *Consider control system $S = (B, F)$ on M and the smooth surjection $\Phi : M \rightarrow N$. Then S is strongly consistent with respect to Φ if and only if for all $p \in M$,*

$$\text{Reach}(p, S) = \Phi^{-1}(\Phi(\text{Reach}(\Phi^{-1}(\Phi(p)), S))). \quad (5.16)$$

Proof. The inclusion $\text{Reach}(p, S) \subseteq \Phi^{-1}(\Phi(\text{Reach}(\Phi^{-1}(\Phi(p)), S)))$ always holds. Let $q \in \Phi^{-1}(\Phi(\text{Reach}(\Phi^{-1}(\Phi(p)), S)))$. Then there exists $q' \in \text{Reach}(\Phi^{-1}(\Phi(p)), S)$ with $\Phi(q') = \Phi(q)$. Let $p' \in \Phi^{-1}(\Phi(p))$ be such that $q' \in \text{Reach}(p', S)$. Since $\Phi(q) = \Phi(q')$ and $\Phi(p) = \Phi(p')$, strong consistency implies $q \in \text{Reach}(p, S)$.

Conversely, assume (5.16) holds. Let $q \in \text{Reach}(p, S)$ and p', q' be such that $\Phi(p') = \Phi(p)$, $\Phi(q') = \Phi(q)$. Then

$$\begin{aligned} q' \in \Phi^{-1}(\Phi(q)) &\subseteq \Phi^{-1}(\Phi(\text{Reach}(p, S))) \\ &\subseteq \Phi^{-1}(\Phi(\text{Reach}(\Phi^{-1}(\Phi(p)), S))) \\ &= \Phi^{-1}(\Phi(\text{Reach}(\Phi^{-1}(\Phi(p')), S))) \\ &= \text{Reach}(p', S) \end{aligned}$$

Therefore, S is strongly consistent. □

Since strong consistency is a more restrictive notion, it is natural that condition (5.16) is stronger than condition (5.15) for consistency.

Proposition 5.22 (Controllability Equivalence). *Consider control systems $S_M = (B_M, F_M)$ and $S_N = (B_N, F_N)$ which are Φ -related with respect to smooth surjection $\Phi : M \rightarrow N$. Assume that S_M is an implementation of S_N , and S_M is strongly consistent. Then S_N is controllable if and only if S_M is controllable.*

Proof. Let $p_1, p_2 \in M$ any points. Let $q_1 = \Phi(p_1)$ and $q_2 = \Phi(p_2)$. Since S_N is controllable, there exists a trajectory of S_N connecting q_1 and q_2 . Since S_M is an implementation of S_N , there exists a trajectory of S_M connecting some $p'_1 \in \Phi^{-1}(q_1)$ and some $p'_2 \in \Phi^{-1}(q_2)$. Then, since S_M is strongly consistent, there is a trajectory of S_M connecting p_1 to p_2 . The other direction is given by Theorem 5.13. □

In this section we identified the relevant notions for the study of controllability in Φ -related systems. We also described them for arbitrary systems in terms of reachable sets. In the following sections we give concrete characterizations of these concepts for linear systems. Moreover, we show how to use them to construct explicit Φ -related systems with the desirable properties.

5.4 Consistent Linear Abstractions

The notion of Φ -related control systems is now specialized for the case of linear, time invariant systems with linear aggregation maps. Consider the linear control systems

$$(\Sigma_1) \quad \dot{x} = Ax + Bu$$

$$(\Sigma_2) \quad \dot{y} = Fy + Gv$$

with $x \in \mathbb{R}^n$, $u \in \mathbb{R}^k$, $y \in \mathbb{R}^m$, $v \in \mathbb{R}^l$, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times k}$, $F \in \mathbb{R}^{m \times m}$, $G \in \mathbb{R}^{m \times l}$, and the surjective, linear aggregation map $y = Cx$. Then by Definition 5.8, Σ_1 and Σ_2 are C -related if for all $x \in \mathbb{R}^n$, and $u \in \mathbb{R}^k$ there exists $v \in \mathbb{R}^l$ such that

$$C(Ax + Bu) = FCx + Gv \quad (5.17)$$

By Proposition 5.9, given any control system and any map Φ , there always exists another control system which is Φ -related to it. We are interested, however, in a constructive methodology for generating Φ -related systems. The following proposition gives us a systematic way to generate C -related linear abstractions of a linear system with respect to a linear aggregation map $y = Cx$.

Proposition 5.23 (Construction of Linear Abstractions). *Consider the linear system*

$$(\Sigma_1) \quad \dot{x} = Ax + Bu$$

and a surjective map $y = Cx$. Let

$$(\Sigma_2) \quad \dot{y} = Fy + Gv$$

be the system where

$$F = CAC^+$$

$$G = [CB \ CAv_1 \ \dots \ CAv_r]$$

with C^+ a left pseudoinverse of C and v_1, \dots, v_r spanning $\text{Ker}(C)$. Then Σ_1 and Σ_2 are C -related.

Proof. We need to show that for all $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^k$, there exists $v \in \mathbb{R}^l$ such that

$$\begin{aligned} C(Ax + Bu) &= Fy + Gv \quad \text{or equivalently} \\ Gv &= CBu + (CA - FC)x \end{aligned}$$

Clearly, CBu belongs in the range of G for all u . Decompose $\mathbb{R}^n = \text{Ker}(C) \oplus \text{Ker}(C)^\perp$. If $x \in \text{Ker}(C)^\perp$ then $C^+Cx = x$ and thus

$$(CA - FC)x = (CA - CAC^+C)x = 0$$

If $x \in \text{Ker}(C)$ then $(CA - FC)x = CAx$ which also belongs in the range of G . \square

It is immediate from Proposition 5.23 that an abstraction of a linear system with respect to a linear aggregation map can also be a linear system. Proposition 5.23 is interesting as it constructively generates for linear systems the so called *virtual inputs* used in backstepping designs [56]. In particular, if the aggregation map is a projection on some of the states, then the states that are ignored appear as inputs at the abstracted system. As another special case, suppose that $\text{Ker}(C) = \text{Im}(B)$. Then we can take as v_1, \dots, v_r the columns of B . The input vectors for Σ_2 are the images under C of the vectors Av_i , which correspond to the next r vectors in the controllability matrix of Σ_1 . That is, the image under C of the first order Lie brackets of Σ_1 become the new input vectors for Σ_2 . The following example illustrates the proposition.

Example 19. Consider again the double integrator

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u \end{aligned}$$

and the projection $y = x_1$. Then $\text{Ker}(C) = \text{span}\{[0 \ 1]^T\}$ and the procedure of Proposition 5.23 results in $F = 0$, $G = 1$, so

$$\dot{y} = v.$$

Now consider the dynamics of the oscillating vector field

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 \end{aligned}$$

with the same projection map $y = x_1$. Then Proposition 5.23 results in the same control system (or better, differential inclusion)

$$\dot{y} = v$$

The fact that the coarser system may have control inputs, even though the original one did not, is clearly undesirable. However, as will be shown, this will be taken care of by the notion of consistency.

From linear systems theory we know that for the linear system

$$(\Sigma_1) \quad \dot{x} = Ax + Bu$$

the reachable space from any $x_0 \in \mathbb{R}^n$ is given by

$$Reach(x_0, \Sigma_1) = \bigcup_{T \geq 0} e^{AT} x_0 + Reach(0, \Sigma_1) = \bigcup_{T \geq 0} e^{AT} x_0 + \mathcal{R}(A, B) \quad (5.18)$$

where

$$\mathcal{R}(A, B) = Im[B \ AB \ \dots \ A^{n-1}B]$$

is the reachable space from the origin. In particular, system Σ_1 is controllable if and only if $\mathcal{R}(A, B) = \mathbb{R}^n$. As a corollary of Theorem 5.13 we obtain the following result.

Theorem 5.24 (Controllability Propagation for Linear Abstractions). *Consider the linear systems*

$$(\Sigma_1) \quad \dot{x} = Ax + Bu$$

$$(\Sigma_2) \quad \dot{y} = Fy + Gv$$

which are C-related which respect to the surjective map $y = Cx$. Then

$$C\mathcal{R}(A, B) \subseteq \mathcal{R}(F, G)$$

In particular, if Σ_1 is controllable then Σ_2 is controllable.

Proof. Simple application of Theorem 5.13. □

In order to propagate controllability from the linear system Σ_2 to Σ_1 , the notions of implementability and consistency were defined in Section 5.3.

Proposition 5.25 (Implementability Characterization for Linear Systems). *Consider two linear systems*

$$(\Sigma_1) \quad \dot{x} = Ax + Bu$$

$$(\Sigma_2) \quad \dot{y} = Fy + Gv$$

and surjective map $y = Cx$. Then Σ_2 is implementable by Σ_1 if and only if for all y we have

$$\bigcup_{T \geq 0} e^{FT}y + \mathcal{R}(F, G) \subseteq \bigcup_{T \geq 0} \bigcup_{x \in C^{-1}(y)} Ce^{AT}x + C\mathcal{R}(A, B) \quad (5.19)$$

Proof. Follows from Proposition 5.15 and Equation (5.18). \square

The following theorem gives a simple characterization of consistency for linear systems in terms of subspace invariance.

Theorem 5.26 (Consistency Characterization for Linear Systems). *The linear system*

$$(\Sigma_1) \quad \dot{x} = Ax + Bu$$

is consistent with respect to the map $y = Cx$ if and only if

$$AKer(C) \subseteq Ker(C) + \mathcal{R}(A, B) \quad (5.20)$$

Proof. First notice that for any set $\mathcal{V} \subseteq \mathbb{R}^n$ we have $C^{-1}(C\mathcal{V}) = \mathcal{V} + Ker(C)$.

Assume (5.20) holds. We must show consistency condition (5.15), which for linear systems requires, for all x that $C(Reach(x + Ker(C), \Sigma_1)) = C(Reach(x, \Sigma_1))$, or, equivalently

$$C \left(\bigcup_{T \geq 0} e^{AT}(x + Ker(C)) + \mathcal{R}(A, B) \right) = C \left(\bigcup_{T \geq 0} e^{AT}x + \mathcal{R}(A, B) \right). \quad (5.21)$$

Clearly, $CReach(x, \Sigma_1) \subseteq C(Reach(x + Ker(C), \Sigma_1))$. Condition (5.20) and A -invariance of $\mathcal{R}(A, B)$ imply that for all $T \geq 0$ we have

$$\begin{aligned} e^{AT}Ker(C) &\subseteq Ker(C) + \mathcal{R}(A, B) && \text{and therefore} \\ Ce^{AT}Ker(C) &\subseteq C\mathcal{R}(A, B). \end{aligned}$$

This gives the other inclusion, proving consistency.

Conversely, assume that Σ_1 is consistent. Let $x_0 \in Ker(C)$. From (5.21) with $x = 0$ we get for any $T > 0$ there exists $r \in \mathcal{R}(A, B)$ such that $Ce^{AT}x_0 = Cr$. Therefore, $e^{AT}x_0 = x'_0 + r$ for some $x'_0 \in Ker(C)$.

We have therefore shown that for all $T > 0$, $e^{TA}x_0 \in Ker(C) + \mathcal{R}(A, B)$. By using $\frac{de^{tA}}{dt} = Ae^{tA}$ and taking limits as $T \rightarrow 0$ we conclude that $Ax_0 \in Ker(C) + \mathcal{R}(A, B)$. \square

Note that condition (5.20) is clearly weaker than the well known condition

$$AKer(C) \subseteq Ker(C) + \mathcal{R}(B)$$

for $Ker(C)$ to be a controlled-invariant (or (A, B) -invariant) subspace.

Theorem 5.27 (Strong Consistency Characterization for Linear Systems). *The linear system*

$$(\Sigma_1) \quad \dot{x} = Ax + Bu$$

is strongly consistent with respect to the map $y = Cx$ if and only if

$$Ker(C) \subseteq \mathcal{R}(A, B) \tag{5.22}$$

Proof. Assume Σ_1 is strongly consistent. Condition 5.16 for linear systems becomes

$$\bigcup_{T \geq 0} e^{AT}x + \mathcal{R}(A, B) = \bigcup_{T \geq 0} e^{AT}(x + Ker(C)) + \mathcal{R}(A, B) + Ker(C). \tag{5.23}$$

Using (5.23) with $x = 0$ gives $\mathcal{R}(A, B) \supseteq Ker(C)$.

Conversely, assume (5.22) holds. By A -invariance of $\mathcal{R}(A, B)$ we get, for all $T \geq 0$,

$$e^{AT}Ker(C) \subseteq \mathcal{R}(A, B).$$

This gives the inclusion

$$\bigcup_{T \geq 0} e^{AT}x + \mathcal{R}(A, B) \supseteq \bigcup_{T \geq 0} e^{AT}(x + Ker(C)) + \mathcal{R}(A, B) + Ker(C).$$

The other inclusion always holds. \square

Note that by the A -invariance of $\mathcal{R}(A, B)$, condition (5.22) is indeed stronger than condition (5.20). Consistency conditions (5.20) and (5.22) are rather intuitive. Condition (5.20) essentially says that whatever piece of $Ker(C)$ is not A -invariant can be compensated by controls and their Lie brackets. On the other hand, condition (5.22) is a form

of controllability within the equivalence classes. The trajectories of the system which connect two points of the same equivalence class (as defined by C) are not, however, restricted to remain within the equivalence class. The following example illustrates the notions of implementability and consistency.

Example 20. Consider the linear system (without controls) $\dot{x} = Ax$, where

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad C = [1 \quad 0]$$

and the C -related (one-dimensional) system $\dot{y} = Fy + Gv$, where $F = 0$ $G = 1$. We also have

$$\text{Ker}(C) = \text{span}\{[0 \quad 1]^T\} \quad A\text{Ker}(C) = \text{span}\{[1 \quad 0]^T\} \not\subseteq \text{Ker}(C)$$

Therefore, the system Σ_1 is not consistent. To show it is implementable we simply solve the system explicitly. Notice that since $\dot{y} = v$, any two points (of \mathbb{R}) can be connected by a trajectory of Σ_2 in arbitrary positive time. Let $y_0, y_f \in \mathbb{R}$. The curve

$$\begin{aligned} x_1(t) &= \frac{y_f - y_0}{T}t + y_0 \\ x_2(t) &= \frac{y_f - y_0}{T} \end{aligned}$$

is a trajectory of Σ_1 from $[y_0 \quad \frac{y_f - y_0}{T}]^T$ to $[y_f \quad \frac{y_f - y_0}{T}]^T$ at time T . Therefore, Σ_2 is implementable by Σ_1 . Notice, that if $y_f \neq y_0$ there is not trajectory of Σ_1 connecting $[y_0 \quad 0]^T$ to any point x with $Cx = y_f$. The reason is that all the points $[x_1 \quad 0]^T$ are equilibria of Σ_1 .

In order to propagate some form of controllability from Σ_2 to Σ_1 , we need to check two properties, namely implementability and (strong) consistency. Unfortunately, Condition (5.19) is not easy to check since it involves the explicit integration of the differential equation. However, condition (5.19) in conjunction with consistency conditions (5.20) or (5.22) results in checkable characterizations of implementations which are also (strongly) consistent. To achieve this, we will need the following lemma.

Lemma 5.28. *Let A ($n \times n$), C ($m \times m$), F ($m \times m$) and G ($m \times l$) be matrices with $l \leq m$ and G of full rank. If for all $x \in \mathbb{R}^n$ $(CA - FC)x \in \mathcal{R}(F, G)$, then for all $t \geq 0$,*

$$(Ce^{tA} - e^{tF}C)x \in \mathcal{R}(F, G) .$$

In particular, the conclusion holds if A, F, G are the corresponding matrices for the C -related systems Σ_1 and Σ_2 .

Proof. We have the following identity for all $t \geq 0$

$$Ce^{tA} - e^{tF}C = \sum_{j=0}^{\infty} (CA^j - F^jC) \frac{t^j}{j!}. \quad (5.24)$$

We prove by induction the statement

$$(P_j) \quad \forall x \in \mathbb{R}^n \quad (CA^j - F^jC)x \in \mathcal{R}(F, G)$$

It is clearly true for $j = 0$ and by hypothesis it is also true for $j = 1$. Assume P_i holds for $i \leq j$. We can write,

$$(CA^{j+1} - F^{j+1}C)x = (CA^j - F^jC)Ax + F^j(CA - FC)x.$$

By the inductive hypothesis applied to x and Ax , $(CA^j - F^jC)Ax \in \mathcal{R}(F, G)$ and $(CA - FC)x \in \mathcal{R}(F, G)$. But then $F^j(CA - FC)x \in \mathcal{R}(F, G)$ for all j since $\mathcal{R}(F, G)$ is F -invariant. Therefore,

$$(CA^j - F^jC)Ax + F^j(CA - FC)x \in \mathcal{R}(F, G).$$

By taking the limit in (5.24) we conclude the proof. \square

Theorem 5.29 (Implementability and Consistency Characterization). *Consider the linear systems*

$$(\Sigma_1) \quad \dot{x} = Ax + Bu$$

$$(\Sigma_2) \quad \dot{y} = Fy + Gv$$

which are C -related which respect to the surjective map $y = Cx$. Then Σ_2 is implementable by Σ_1 and Σ_1 is consistent if and only if

$$C\mathcal{R}(A, B) = \mathcal{R}(F, G) \quad (5.25)$$

In addition, Σ_2 is implementable by Σ_1 and Σ_1 is strongly consistent if and only if

$$\mathcal{R}(A, B) = C^{-1}(\mathcal{R}(F, G)) \quad (5.26)$$

Proof. Assume $C\mathcal{R}(A, B) = \mathcal{R}(F, G)$ and thus $\mathcal{R}(F, G) \subseteq C\mathcal{R}(A, B)$. Now let $x \in \text{Ker}(C)$. By C -relatedness, there exists $v \in \mathbb{R}^l$ such that $CAx = FCx + Gv = Gv$ (using $u = 0$ and since $Cx = 0$). So, $CAx \in \mathcal{R}(F, G)$ and by assumption, there is $x_1 \in \mathcal{R}(A, B)$ such that $Cx_1 = CAx$. Therefore, $Ax - x_1 \in \text{Ker}(C)$ and $Ax = Ax - x_1 + x_1 \in \text{Ker}(C) +$

$\mathcal{R}(A, B)$. Thus $AKer(C) \subseteq Ker(C) + \mathcal{R}(A, B)$ and Σ_1 is consistent. We must now show that condition (5.19) holds. Consider any

$$y_f = e^{FT} y_0 + r_F^1 \in Reach(y_0, \Sigma_2) = \bigcup_{T \geq 0} e^{FT} y_0 + \mathcal{R}(F, G)$$

with $r_F^1 \in \mathcal{R}(F, G)$. By Lemma 5.28, we have that $e^{FT} y_0 = Ce^{AT} x_0 + Cr_F^2$ for some $r_F^2 \in \mathcal{R}(A, B)$, and for any x_0 with $y_0 = Cx_0$. But then

$$\begin{aligned} y_f = Ce^{AT} x_0 + r_F^1 + r_F^2 &= Ce^{AT} x_0 + Cr_A \in \bigcup_{T \geq 0} \bigcup_{x \in C^{-1}(y_0)} Ce^{AT} x + C\mathcal{R}(A, B) \\ &= C(Reach(C^{-1}(y_0), \Sigma_1)) \end{aligned}$$

for some $r_A \in \mathcal{R}(A, B)$ since $\mathcal{R}(F, G) \subseteq C\mathcal{R}(A, B)$. Therefore Σ_2 is implementable by Σ_1 .

For the converse notice that, since the systems are C -related, Proposition 5.24 implies $\mathcal{R}(F, G) \supseteq C\mathcal{R}(A, B)$. Moreover, the implementability condition (5.19) with $y = 0$ gives

$$\mathcal{R}(F, G) \subseteq \bigcup_{T \geq 0} Ce^{AT} Ker(C) + C\mathcal{R}(A, B).$$

And the consistency condition (5.21) with $x = 0$ gives

$$\bigcup_{T \geq 0} Ce^{AT} Ker(C) \subseteq C\mathcal{R}(A, B).$$

These two combined give $\mathcal{R}(F, G) \subseteq C\mathcal{R}(A, B)$. This concludes the proof of the first equivalence.

Now assume that $\mathcal{R}(A, B) = C^{-1}(\mathcal{R}(F, G))$. Then $C\mathcal{R}(A, B) = \mathcal{R}(F, G)$ and therefore Σ_1 implements Σ_2 . Since $0 \in \mathcal{R}(F, G)$ we also have $Ker(C) \subseteq \mathcal{R}(A, B)$. Therefore Σ_1 is strongly consistent.

If Σ_1 is strongly consistent and implements Σ_2 then Σ_1 is also consistent and therefore must satisfy $C\mathcal{R}(A, B) = \mathcal{R}(F, G)$. Therefore, $\mathcal{R}(A, B) \subseteq C^{-1}(\mathcal{R}(F, G)) = \mathcal{R}(A, B) + Ker(C)$. By strong consistency $Ker(C) \subseteq \mathcal{R}(A, B)$, and thus $C^{-1}(\mathcal{R}(F, G)) \subseteq \mathcal{R}(A, B)$. Therefore $C^{-1}(\mathcal{R}(F, G)) = \mathcal{R}(A, B)$. \square

We now have the main ingredients for propagating controllability from the coarser to the more complex model.

Theorem 5.30 (Consistency and Implementability imply Controllability). *Consider the linear systems*

$$(\Sigma_1) \quad \dot{x} = Ax + Bu$$

$$(\Sigma_2) \quad \dot{y} = Fy + Gv$$

which are C -related system with respect to the surjection $y = Cx$. Assume that Σ_1 implements Σ_2 , and Σ_1 is consistent, that is $CR(A, B) = \mathcal{R}(F, G)$. Then Σ_2 is controllable if and only if Σ_1 is macrocontrollable. If in addition Σ_1 is strongly consistent, that is $\mathcal{R}(A, B) = C^{-1}(\mathcal{R}(F, G))$, then Σ_1 is controllable if and only if Σ_2 is controllable.

Proof. Same as the proof of Propositions 5.19 and 5.22. □

Thus, in order to propagate controllability between two linear systems, we have to ensure that the systems are C -related and check either condition (5.25) or (5.26) depending on the notion of controllability that is needed. It is desirable to have a methodology for constructing C related systems with the desirable properties. Fortunately, for the C -related system constructed in Proposition 5.23, (strong) consistency implies implementability. In order to show this, we will need the following lemma.

Lemma 5.31. *Let $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times k}$, and full rank $C \in \mathbb{R}^{m \times n}$, be such that*

$$AKer(C) \subseteq Ker(C) + \mathcal{R}(A, B)$$

and let $F = CAC^+$. Then $CR(A, B)$ is F -invariant, that is

$$FCR(A, B) \subseteq CR(A, B)$$

Proof. Let $y = Cx$ for $x \in \mathcal{R}(A, B)$ and consider

$$Fy = CAC^+y = CAC^+Cx$$

Decompose $x = x^c + x^n$ where $x^c \in Ker(C)$ and $x^n \in Ker(C)^\perp$. Then

$$Fy = CAC^+C(x^c + x^n) = CAx^n = CA(x - x^c)$$

Since $x \in \mathcal{R}(A, B)$ and $\mathcal{R}(A, B)$ is A -invariant, we get that $CAx \in CR(A, B)$. By consistency, there exist $z^c \in Ker(C)$ and $z^r \in \mathcal{R}(A, B)$ such that

$$CAx^c = C(z^c + z^r) = Cz^r \tag{5.27}$$

Thus CAx^c also belongs in $CR(A, B)$ resulting in $Fy \in CR(A, B)$. □

Theorem 5.32 (Consistency implies Implementability). *Consider the linear system*

$$(\Sigma_1) \quad \dot{x} = Ax + Bu$$

which is consistent with respect to the surjective map $y = Cx$. Let

$$(\Sigma_2) \quad \dot{y} = Fy + Gv$$

be the system where

$$\begin{aligned} F &= CAC^+ \\ G &= [CB \ CAv_1 \ \dots \ CAv_r] \end{aligned}$$

with C^+ a left pseudoinverse of C and v_1, \dots, v_r spanning $\text{Ker}(C)$. Then Σ_2 is implementable by Σ_1 .

Proof. By Theorem 5.24 we have that $\mathcal{R}(F, G) \supseteq CR(A, B)$ and thus we only need to show that $\mathcal{R}(F, G) \subseteq CR(A, B)$. Let $y_f \in \mathcal{R}(F, G)$. Then

$$y_f = [G \ FG \ \dots \ F^{m-1}G] x \quad (5.28)$$

for some $x \in \mathbb{R}^{ml}$. By an appropriate partition of $x = [x_1 \ x_2 \ \dots \ x_m]^T$ we get

$$y_f = Gx_1 + FGx_2 + \dots + F^{m-1}Gx_m \quad (5.29)$$

It suffices to show that $\mathcal{R}(G) \subseteq CR(A, B)$ since then, by Lemma 5.31, we get that $\mathcal{R}(FG) \subseteq CR(A, B), \dots, \mathcal{R}(F^{m-1}G) \subseteq CR(A, B)$. Now consider

$$y_1 = Gx_1 = [CB \ CAv_1 \ \dots \ CAv_k] \begin{bmatrix} x_1^1 \\ x_1^2 \end{bmatrix} = CBx_1^1 + [CAv_1 \ \dots \ CAv_k] x_1^2 \quad (5.30)$$

Clearly, $CBx_1^1 \in CR(A, B)$. By consistency we have

$$AKer(C) \subseteq Ker(C) + \mathcal{R}(A, B) \quad (5.31)$$

and therefore for $i = 1, \dots, k$

$$Av_i = v_i^c + v_i^r \quad (5.32)$$

for some $v_i^c \in Ker(C)$ and $v_i^r \in \mathcal{R}(A, B)$. Thus

$$\begin{aligned} CAv_i &= C(v_i^c + v_i^r) = Cv_i^r \\ &= C [B \ AB \ \dots \ A^{n-1}B] q_i \end{aligned} \quad (5.33)$$

for some vectors q_i of appropriate dimension. But then

$$\begin{aligned} [CAv_1 \dots CAv_k] x_1^2 &= C [B AB \dots A^{n-1}B] [q_1 \dots q_k] x_1^2 \\ &= C [B AB \dots A^{n-1}B] X_1^2 \end{aligned} \quad (5.34)$$

and thus $\mathcal{R}(G) \in C\mathcal{R}(A, B)$. \square

As a result of the above theorem, if we use Proposition 5.23 to construct our abstracted models, then consistency (or strong consistency) is the only condition on the aggregation map that is needed to propagate controllability.

Theorem 5.33 (Consistency Implies Controllability). *Consider the linear system*

$$(\Sigma_1) \quad \dot{x} = Ax + Bu$$

and surjective map $y = Cx$. Let

$$(\Sigma_2) \quad \dot{y} = Fy + Gv$$

be the C -related system where

$$\begin{aligned} F &= CAC^+ \\ G &= [CB CAv_1 \dots CAv_r] \end{aligned}$$

with C^+ the pseudoinverse of C and v_1, \dots, v_r spanning $\text{Ker}(C)$. If

$$A\text{Ker}(C) \subseteq \text{Ker}(C) + \mathcal{R}(A, B)$$

then Σ_2 is macrocontrollable if and only if Σ_1 is controllable. In particular, if

$$\text{Ker}(C) \subseteq \mathcal{R}(A, B)$$

then Σ_1 is controllable if and only if Σ_2 is controllable.

Proof. Follows from Theorems 5.30 and 5.32. \square

It is interesting to notice what happens to conditions (5.22) and (5.20) when the linear system is a linear vector field and thus $B = 0$. In that case, condition (5.20) reduces to

$$A\text{Ker}(C) \subseteq \text{Ker}(C)$$

which, recall from Section 2.1, is the necessary and sufficient condition to obtain a well defined quotient vector field. Therefore a consistent abstraction of a linear vector field cannot have any control inputs (or cannot be a differential inclusion). Condition (5.22) reduces to

$$\text{Ker}(C) = \{0\}$$

and thus $y = Cx$ must be an invertible linear transformation (since it is already surjective). We will be typically interested in consistent abstractions which are *nontrivial*, in the sense that some state space reduction is performed (thus $\text{Ker}(C) \neq \{0\}$), but the abstracted system is also nontrivial ($\text{Ker}(C) \neq \mathbb{R}^n$).

Corollary 5.34. *Consider the assumptions of Theorem 5.33 and assume that $0 < \text{rank}(B) < n$. Then a nontrivial, strongly consistent abstraction always exists.*

Proof. If $\text{rank}(B) > 0$ then we can always find a linear map C such that $\text{Ker}(C) = \text{Im}[B]$. \square

Theorem 5.33 and Corollary 5.34 are important as they show that a *consistent abstraction always exists as long as there are control inputs*. In addition, the notions of consistency are important from a hierarchical perspective as they provide good design principles for constructing valid hierarchies. For example, the condition for strong consistency, $\text{Ker}(C) \subseteq \mathcal{R}(A, B)$, suggests that in order to ignore dynamics at a higher level (captured by $\text{Ker}(C)$) then one would have to ensure the ignored dynamics can be accommodated at the lower level.

As one imposes more restrictions on the matrix C further properties can be propagated from one system to the other. The following results show conditions under which full trajectories can be implemented by the lower level system.

Theorem 5.35 (Trajectory Implementation). *Consider two linear systems*

$$(\Sigma_1) \quad \dot{x} = Ax + Bu$$

$$(\Sigma_2) \quad \dot{y} = Fy + Gv$$

and the surjective map $y = Cx$. Assume $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$ with $m \leq n$, and $u \in \mathbb{R}^k$ with $k \leq n$. We assume B is of full rank. Let $\mathcal{K} = \text{Ker}(C)$, $\mathcal{B} = \text{Im}[B]$, $\mathcal{G} = \text{Im}[G]$, and let P denote the orthogonal projection from \mathbb{R}^m onto $C\mathcal{K} + C\mathcal{B}$. We make the following two assumptions:

1. $CAx = FCx$ for all $x \in \mathcal{K}^\perp$ (the orthogonal complement of \mathcal{K}).

2. $C^{-1}((I - P)\mathcal{G}) \subseteq \mathcal{B}$

Then for every trajectory $y(\cdot)$ of Σ_2 corresponding to a differentiable control there exists a trajectory $x(\cdot)$ of Σ_1 such that $y(t) = Cx(t)$ for all t in the domain of $y(\cdot)$.

Proof. Let $y(\cdot)$ be a trajectory of Σ_2 corresponding to the control v . First we define $x_a(t) = C^+y(t)$ where C^+ is the Moore-Penrose pseudo-inverse of C ($C^+ = C^T(CC^T)^{-1}$). If $z \in \mathcal{K}$ then

$$z^T x_a(t) = z^T C^T (CC^T)^{-1} y(t) = (Cz)^T (CC^T)^{-1} y(t) = 0.$$

Therefore, $x_a(t) \in \mathcal{K}^\perp$ for all t . Moreover, $\dot{x}_a(t) = C^+ \dot{y}(t)$ where $\dot{y}(t) = Fy(t) + Gv(t)$.

Let P denote the orthogonal projection from \mathbb{R}^m onto $CA\mathcal{K} + CB$. Let D be the restriction of C on $A\mathcal{K} + \mathcal{B}$ and let D^+ be its pseudoinverse. Define $\bar{x}(t) = D^+P(Gv(t))$, and therefore by construction we have that $C\bar{x}(t) = P(Gv(t))$ and $\bar{x}(t) \in A\mathcal{K} + \mathcal{B}$. Thus there exist $x_b(t) \in \mathcal{K}$ and $b(t) \in \mathcal{B}$ such that $\bar{x}(t) = Ax_b(t) + b(t)$. Since $\bar{x}(t)$ is differentiable we may choose $x_b(t)$ and $b(t)$ to be differentiable as well (using a suitable pseudoinverse). Let $x(t) = x_a(t) + x_b(t)$. Then $Cx(t) = C(x_a(t) + x_b(t)) = Cx_a(t) = y(t)$ and in addition

$$C\dot{x} = C(\dot{x}_a + \dot{x}_b) = C\dot{x}_a = \dot{y} = Fy + Gv = FCx_a + Gv = CAx_a + Gv$$

where the last equality holds by Assumption 1. Set $z(t) = \dot{x}(t) - Ax_a(t) - \bar{x}(t)$. Then for all t , $Cz(t) = C(\dot{x}_a(t) + \dot{x}_b(t)) - CAx_a(t) - C\bar{x}(t) = CAx_a(t) + Gv(t) - CAx_a(t) - P(Gv(t)) = (I - P)Gv(t)$. By Assumption 2, for each t there is $u(t) \in \mathbb{R}^k$ such that $z(t) = Bu(t)$. In fact, we can take $u(t) = B^+z(t)$ (here $B^+ = (B^T B)^{-1} B^T$ since $k \leq n$). Then if we let $x(t) = x_a(t) + x_b(t)$ we get $\dot{x}(t) = Ax(t) + Bu(t)$ and $Cx(t) = Cx_a(t) = y(t)$ for all t . \square

Corollary 5.36. *Let Σ_1 , Σ_2 , and C be as in Proposition 5.23. If $\text{Ker}(C) \subseteq \text{Im}[B]$, then for every trajectory $y(\cdot)$ of Σ_2 corresponding to a differentiable control there exists a trajectory $x(\cdot)$ of Σ_1 such that $y(t) = Cx(t)$ for all t in the domain of $y(\cdot)$.*

Proof. Set $\mathcal{K} = \text{Ker}(C)$, $\mathcal{B} = \text{Im}[B]$, and $\mathcal{G} = \text{Im}[G]$. Since $C^+Cx = x$ for $x \in \mathcal{K}^\perp$, Assumption 1 of Theorem 5.35 is satisfied. Now $G = [CB \ CAv_1 \ \dots \ CAv_r]$, and since P is the orthogonal projection onto $CA\mathcal{K} + CB$, we get $(I - P)\mathcal{G} = 0$. Then Assumption 2 of Theorem 5.35 reduces to $C^{-1}(0) = \text{Ker}(C) \subseteq \text{Im}[B]$ which is our assumption. \square

5.5 Hierarchical Controllability Algorithm

In this section, we will take advantage of the results of Section 5.4 in order to analyze the controllability of large scale linear systems. Theorem 5.33 enables us to have a hierarchical controllability criterion which decomposes the controllability problem into a sequence of smaller problems. Such an approach is numerically more efficient and robust than the standard Kalman rank and Popov-Belevitch-Hautus (PBH) eigenvalue tests.

Conceptually the algorithm, starts with the linear system in question, and determines the number of linearly independent input vector fields. If this number is zero, then the system is uncontrollable and the algorithm terminates. If the number of linearly independent inputs is equal to the number of states, then the system is trivially controllable and the algorithm terminates as well. If the number of linearly independent vector fields is less than the number of states but greater than zero, then by Corollary 5.34 we can always find an aggregation matrix C satisfying the strong consistency condition $Ker(C) \subseteq \mathcal{R}(A, B)$. Since $Im[B \ AB \ \dots \ A^k B] \subseteq Im[B \ AB \ \dots \ A^{n-1} B]$ for any $0 \leq k \leq n - 1$, from a computational standpoint, we can actually choose any matrix C satisfying $Ker(C) = Im[B \ AB \ \dots \ A^k B]$ for $0 \leq k \leq n - 1$. If $k = 0$, then the abstracted system essentially ignores the directions spanned by the input vector fields (which are trivially controllable). As k goes up, we not only ignore the directions of the input vector fields, but also their Lie brackets with the drift dynamics. If $k = n - 1$ then the matrix C will ignore the whole reachable space.

After a consistent C matrix is determined, the construction of Theorem 5.33 is used in order to obtain a system of smaller dimension with equivalent controllability properties. We recursively apply the same procedure to this new abstracted system. Eventually, by dimension count, either there will be no inputs left and the system will be trivially uncontrollable, or there should be as many linearly independent inputs as number of states in which case controllability follows trivially. Since at each step, the abstractions that are constructed are consistent, then by Theorem 5.33, the outcome of the algorithm at the coarsest level will propagate along this sequence of consistent abstractions to the original complex model.

Algorithm 5.37. (Hierarchical Controllability Algorithm)

1. Start with system $\dot{x} = Ax + Bu$, $A \in \mathbb{R}^{n \times n}$, $0 \leq k \leq n - 1$
2. If $rank(B)$ is

- 0 : System is uncontrollable. Algorithm Terminates
 - n : System is controllable. Algorithm Terminates
3. Find matrix C such that $\text{Ker}(C) = \text{Im}[B \ AB \ \dots \ A^k B]$
 4. Obtain new system matrices A, B of the abstracted system using Theorem 5.33
 5. Return to 2

The higher the order of the Lie brackets (the larger k is), the fewer steps the algorithm will need to terminate. On the other hand, as k increases, the amount of computation per step will be higher. Before we discuss computational and implementation aspects of the above algorithm, we will demonstrate its application on various examples.

Example 21. Consider the linear system

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} u = A_1 x + B_1 u \quad (5.35)$$

Since there is one linearly independent input field, we can find a consistent abstraction satisfying

$$\text{Ker}(C_1) = \text{Im}[B_1] \subseteq \text{Im}[B_1 \ A_1 B_1 \ A_1^2 B_1]$$

For example, we can choose

$$C_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The construction of Theorem 5.33, then results in

$$A_2 = C_1 A_1 C_1^+ = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad B_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (5.36)$$

Since B_2 is nonzero and the number of linearly independent inputs is strictly less than the number of states, we can obtain another consistent abstraction by choosing $C_2 = [1 \ 0]$. The resulting abstraction is

$$A_3 = C_2 A_2 C_2^+ = 0 \quad B_3 = 1 \quad (5.37)$$

At this point, the number of inputs is equal to the number of states and thus the pair (A_3, B_3) is trivially controllable. By consistency, the pairs (A_2, B_2) and (A_1, B_1) are also controllable.

There is a much more intuitive explanation of the sequence of steps taken above. Note that the system started with the pair (A_1, B_1) and in the first iteration, we essentially removed the dynamics of x_2 (second row) from equation (5.35) since they have direct connection to the input u . This results in the pair (A_2, B_2) where x_2 can now be thought of as an input. We re-apply the above procedure by now removing the dynamics of x_3 (second row of (5.36)) since they can be directly controlled by the new controls. This results in the pair (A_3, B_3) which is trivially controllable.

Example 22. Consider the linear system

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u = A_1 x + B_1 u \quad (5.38)$$

A consistent abstraction results by choosing the aggregation matrix

$$C_1 = \begin{bmatrix} -1 & 1 \end{bmatrix}$$

resulting in

$$A_2 = C_1 A_1 C_1^+ = 0 \quad B_2 = 0 \quad (5.39)$$

Therefore, by Theorem 5.33, the pairs (A_2, B_2) and (A_1, B_1) are both uncontrollable.

In the case where we select $k = 0$ in Algorithm 5.37, then we choose matrices C satisfying $\text{Ker}(C) = \text{Im}[B]$. In this particular case $CB = 0$, and in addition the columns of B span $\text{Ker}(C)$. From a computational standpoint, it is advantageous to actually choose a matrix C which not only satisfies $\text{Ker}(C) = \text{Im}[B]$ but is also a projection to $\text{Im}[B]^\perp$. This reduces some of the computations of Theorem 5.33 and results in the following variation of Algorithm 5.37.

Algorithm 5.38. (Hierarchical Controllability Algorithm)

1. Start with system $\dot{x} = Ax + Bu$, $A \in \mathbb{R}^{n \times n}$.
2. If $\text{rank}(B)$ is

- 0 : System is uncontrollable. Algorithm Terminates
 - n : System is controllable. Algorithm Terminates
3. Find matrix C such that $Ker(C) = Im[B]$
 4. Let $A := CAC^+$, $B := CAB$
 5. Return to 2

Intuitively, Algorithm 5.38 starts with the system in question and, since $Im[B]$ is in the controllable region, it chooses an abstraction matrix C which essentially projects the system in a direction which is orthogonal to the space spanned by B . Thus the macroinputs of the first abstraction are spanned by CAB , which are the first order Lie brackets of the original system, *projected on the orthogonal complement of $Im[B]$* . Similarly, the second abstraction will have as input vector fields the second order Lie brackets projected on the orthogonal complement of both $Im[B]$ and $Im[AB]$. Because of this selection of inputs at each abstraction layer, we simply have to add the dimension of the span of the input vector fields at each abstraction layer in order to obtain the dimension of the controllability subspace. From the above discussion, it is also clear that, if the system is uncontrollable, then the algorithm computes the uncontrollable part of the system since at each iteration we are projecting on the space orthogonal to parts of the controllable space. The sequence of abstracting maps can then be used in a straightforward manner in order to decompose the system into controllable and uncontrollable subsystems.

We now focus on the implementation issues of Algorithms 5.37 and 5.38. For simplicity, we consider Algorithm 5.38 ; Algorithm 5.37 can be treated in a similar manner. From a computational perspective, the two main problems for implementing Algorithm 5.38 are: first, the construction of a consistent aggregation matrix C satisfying $Ker(C) = Im[B]$, and second, given such a matrix, to perform the computations required for the construction of a consistent abstraction. In order to tackle the first problem, we perform a singular value decomposition on the matrix B . The $n \times m$ ($n \geq m$) matrix B with rank r is decomposed as

$$B = U\Sigma V^T = [U_1 \ U_2] \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma_r V_1^T \quad (5.40)$$

where Σ_r is the $r \times r$ matrix of nonzero singular values. From the above decomposition we immediately obtain that $Ker(C) = Im[B] = Im[U_1]$ and we can therefore choose

the abstracting map $C = U_2^T$. In addition, $C^+ = U_2$ and therefore the singular value decomposition gives us for free the pseudoinverse calculation. Similar constructions are used in the implementation of Algorithm 5.37. Of course, singular value decompositions are computationally expensive. If speed of computation is of great interest, then QR type decompositions could be used instead of singular value decompositions in order to accelerate the algorithm. However, as is typical in such cases, this may result in a less robust algorithm. The Matlab code that implements Algorithms 5.37 and 5.38 can be found in Appendix A.

Various experimental, comparative studies were performed on a Matlab platform. Given the dimension of the state and input space, random A , B matrices were generated, and their controllability was checked using the Kalman rank condition, the PBH test and Algorithm 5.38. Floating point operations were measured for each test, and the following ratios

$$\text{Ratio} = \frac{\text{Floating Point Operations of Kalman or PBH Test}}{\text{Floating Point Operations of Algorithm 5.38}}$$

are plotted as a function on state and input dimension in Figures 5.2 and 5.3. The plane with ratio equal to one is also plotted. Whenever the unreliable Kalman rank test fails to recognize a controllable system, the ratio is set to zero. Note from Figure 5.2, that the Kalman rank test is more efficient for very low dimensional systems but Algorithm 5.38 is up to 15 times faster for most systems. In addition, the Kalman condition fails to be reliable for systems with more than approximately 15 states. Figure 5.3 compares the PBH test with Algorithm 5.38. Even though the PBH test is more reliable than the Kalman rank condition, it is significantly slower than Algorithm 5.38 (up to 150 times for some systems). In addition, it is well known (see [84]) that the PBH test is very sensitive to parameter perturbations due to eigenvalue calculations.

The computational and conceptual advantages of Algorithm 5.38 are verified by the fact that Algorithm 5.38 is identical to the controllability algorithm of [39], derived from a purely numerical analysis perspective. In [39], the above algorithm is shown to be numerically stable and is a stabilized version of the realization algorithm of [94] (Matlab command CTRBF). Figure 5.4 compares Algorithm 5.38 with the more general Algorithm 5.37 with $k = 1$. Figure 5.4 clearly shows that it may be advantageous to use Algorithm 5.37 with $k = 1$ only in cases where the state dimension is much larger than the input dimension.

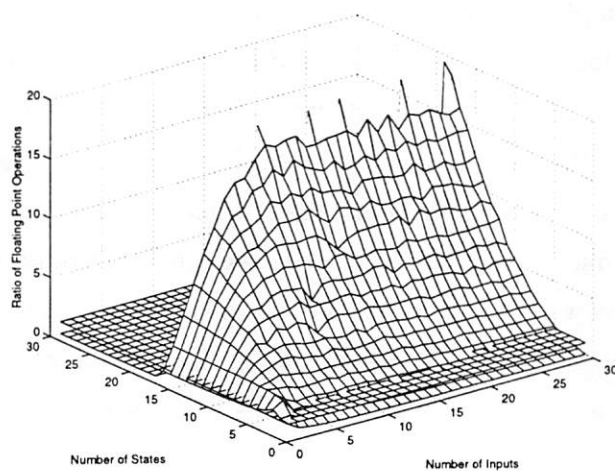


Figure 5.2: Comparison of Algorithm 5.38 and the Kalman rank condition

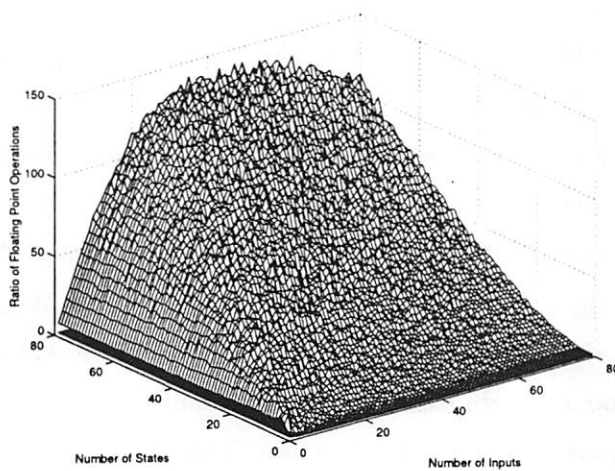


Figure 5.3: Comparison of Algorithm 5.38 and the Popov-Belevitch-Hautus test

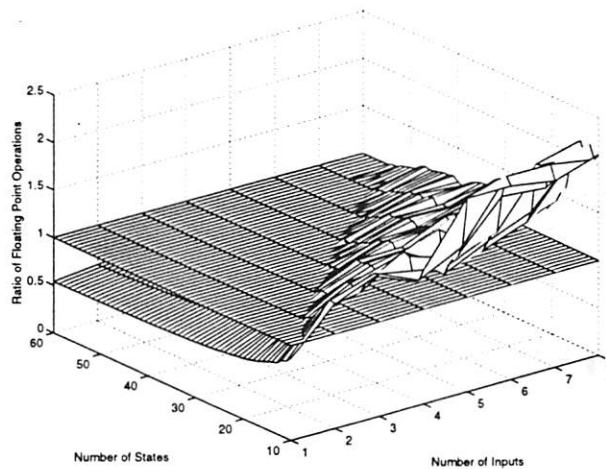


Figure 5.4: Comparison of Algorithm 5.38 and Algorithm 5.37 with $k = 1$

5.6 Conclusions

In this chapter, we considered a notion of control system abstractions which are typically used in hierarchical and multi-layered systems. This was achieved by generalizing the notion of Φ -related vector fields to control systems. This notion mathematically formalizes the concept of virtual inputs used in backstepping designs [56]. The notions of implementability and consistency were then defined in order to propagate controllability from the abstracted system to the more detailed one. These notions were completely characterized for linear systems, and the easily checkable conditions allowed us to construct a hierarchical controllability algorithm for linear systems.

The fact that the hierarchical framework developed in this paper places a geometric and conceptual framework on the best of the known controllability algorithms from numerical linear algebra, is strong evidence that hierarchical decompositions of control problems are indeed reducing the complexity of control algorithms. It is therefore worthwhile pursuing this direction of research for more general classes of systems (nonlinear) as well as for other properties of interest (stabilizability, optimality).

Chapter 6

Conclusions

Next generation large scale systems have motivated us to think of a new control paradigm. As a result, there is a clear need for new modeling frameworks accompanied by powerful analysis and design tools. Hybrid systems, which combine discrete event and continuous dynamics, offer a solution to the modeling challenges faced by system engineers. This dissertation has focused on the modeling and analysis of hierarchical, hybrid systems.

One of the most important problems for safety critical, hybrid systems is the reachability problem which asks whether some unsafe region is reachable from an initial region. Computer aided verification is the main computational approach for formally checking that the system avoids an undesired or unsafe region of the state space. Due to the infinite cardinality of the state space, the decidability of these reachability algorithms is extremely important. Even though state of the art hybrid automata with a decidable reachability problem, rectangular hybrid automata, are expressive enough to capture and verify real time software and hardware properties, their modeling power from a control perspective was rather limited. Chapter 3 shows that the conditions for converting rectangular differential inclusions to constant, decoupled differential inclusion are very restrictive. This severely limits their applicability to systems with complex continuous behavior.

This negative result inspired the work presented in Chapter 4 in an effort to expand the known decidability frontier to capture hybrid systems with more sophisticated continuous dynamics. In this endeavor, very recent results in o-minimal theories from mathematical logic, allowed us to show that all hybrid systems whose relevant sets and continuous flows are definable in an o-minimal theory admit finite bisimulations. This result was then immediately used in order to extend the decidability frontier by capturing

classes of hybrid systems with linear dynamics in each discrete location. The importance of these results is immediately clear given the wide applicability of linear systems in control theory.

Chapter 5 takes the next step for analyzing large scale systems by tackling complexity. Complexity has usually been reduced by hierarchical structures, where higher levels of the hierarchy utilize coarser models or abstractions of the system resulting by aggregating the detailed lower level models. Even though the notion of system abstraction is mature in the computer science community, no such notion exists for continuous systems. Chapter 5 presents the first formal approach to abstracting continuous control systems. Furthermore, hierarchies of linear systems which are consistent with respect to controllability objectives were characterized. This immediately resulted in a hierarchical controllability algorithm for linear systems from which the best known controllability algorithm from numerical linear algebra was recovered. This was strong evidence that hierarchical decompositions of control problems are indeed reducing the complexity of large scale control problems.

As the field of hierarchical, hybrid systems is young there are many more questions than answers. As a result, there are many fundamental and interesting issues for further research.

- **Modeling:** We need to identify classes of hybrid systems which, in addition to being expressive, must also have enough structure to be amenable to analysis. Notions of existence, uniqueness, continuity of solutions, and robustness need be reconsidered in a broader context. Furthermore, the issue of zenoness, systems with infinite switching in finite time, must be resolved in order to have robust models. Also modeling frameworks must be equipped with appropriate compositional and abstraction operators in order to tackle complexity issues.
- **Verification:** The decidability results of Chapter 4 enable us to start building a verification tool for reachability computation of linear hybrid systems. The heart of this tool will be a quantifier elimination engine. This tool will be the first one of its kind that will have both the ability to handle a reasonable number of discrete states as well as linear dynamics in each location. The abilities of this tool will be enhanced as we discover more classes of decidable hybrid systems, in particular, hybrid systems with more general switching behaviors and linear dynamics with control inputs and disturbances.

- **Controller Synthesis:** As quantifier elimination with parameters is possible, the tool will also have the ability to perform controller synthesis for linear hybrid systems. The tool can determine ranges of parameter values for either control inputs or switching surfaces for which the system is guaranteed to be safe. This will allow us to construct hybrid systems which are safe by design, as opposed to verifying completed designs.
- **Simulation:** Even though verification is applied to high level mathematical abstractions of the original system, simulation is needed for model validation purposes. Even though hybrid simulators are currently available, there are no theoretical guarantees that the simulated trajectories are feasible in the original system. Results that determine optimal time steps so that switching surfaces are not missed but also minimize integration time are needed in order to gain confidence in simulation results. In the presence of multiple time scales, this problem becomes even harder. The combination of verification and simulation tools is also a very important issue as there are limits to both sides.
- **Hierarchical Control:** The results in Chapter 5 enable the development of an open loop backstepping methodology which, given a sequence of consistent abstractions would recursively generate the actual control input, by first generating a control input for the abstracted system and then recursively refine it as one adds more modeling detail. Nonlinear analogues of the results of Section 5.4, will provide a hierarchical controllability algorithm for nonlinear systems which may be more efficient and robust from a symbolic computation point of view. Many other properties are also of interest and will be investigated both for linear and nonlinear control systems. For example, obtaining consistent abstractions for nonlinear systems with respect to stabilizability would essentially classify all backsteppable systems. Other properties of interest include trajectory tracking, optimality and the proper propagation of state and input constraints.

Last but not least, the above research must be motivated by, and applied to meaningful large scale systems, like automated highway systems, air traffic management systems, flight management systems, communication and power networks.

Bibliography

- [1] R. Abraham, J. Marsden, and T. Ratiu. *Manifolds, Tensor Analysis and Applications*. Applied Mathematical Sciences. Springer-Verlag, 1988.
- [2] R. Alur, C. Courcoubetis, N. Halbwachs, T.A. Henzinger, P.H. Ho, X. Nicolin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138:3–34, 1995.
- [3] R. Alur and D.L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.
- [4] R. Alur, T.A. Henzinger, and E.D. Sontag, editors. *Hybrid Systems III*, volume 1066 of *Lecture Notes in Computer Science*. Springer-Verlag, 1996.
- [5] M. Andersson. *Object-Oriented Modeling and Simulation of Hybrid Systems*. PhD thesis, Lund Institute of Technology, Lund, Sweden, December 1994.
- [6] P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, editors. *Hybrid Systems II*, volume 999 of *Lecture Notes in Computer Science*. Springer-Verlag, 1995.
- [7] P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, editors. *Hybrid Systems IV*, volume 1273 of *Lecture Notes in Computer Science*. Springer-Verlag, 1997.
- [8] P.J. Antsaklis, J.A. Stiver, and M. Lemmon. Hybrid system modeling and autonomous control systems. In R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, editors, *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 366–392. Springer-Verlag, 1993.
- [9] M. Aoki. Control of large scale dynamic systems by aggregation. *IEEE Transactions on Automatic Control*, 13(3):246–253, June 1968.

- [10] D. S. Arnon, G. E. Collins, and S. McCallum. Cylindrical algebraic decomposition I: The basic algorithm. *SIAM Journal on Computing*, 13(4):865–877, November 1984.
- [11] E. Asarin, O. Maler, and A. Pnueli. Symbolic controller synthesis for discrete and timed systems. In P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, editors, *Hybrid Systems II*, volume 999 of *Lecture Notes in Computer Science*. Springer-Verlag, 1995.
- [12] J.P. Aubin. *Viability Theory*. Systems and Control: Foundations and Applications. Birkhauser, 1991.
- [13] A. Back, J. Guckenheimer, and M. Myers. A dynamical simulation facility for hybrid systems. In R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, editors, *Hybrid Systems*, pages 255–267. Springer Verlag, New York, 1993.
- [14] T. Basar and P. Bernhard. *H^∞ -Optimal Control and Related Minimax Design Problems*. Systems and Control: Foundations and Applications. Birkhauser, 1991.
- [15] J. Bengtsson, K.G. Larsen, F. Larsson, P. Pettersson, and W. Yi. UPPAAL - a tool suite for automatic verification of real-time systems. In *DIMACS Workshop on Verification and Control of Hybrid Systems*. Springer Verlag, 1995.
- [16] E. Bierstone and P.D. Milman. Semianalytic and subanalytic sets. *Inst. Hautes Études Sci. Publ. Math.*, 67:5–42, 1988.
- [17] N. Bjorner, A. Browne, E. Chang, M. Colon, A. Kapur, Z. Manna, H. Sipma, and T. Uribe. STeP: Deductive-algorithmic verification of reactive and real-time systems. In *Computer Aided Verification*, Lecture Notes in Computer Science. Springer-Verlag, July 1996.
- [18] M. Branicky. *Studies in Hybrid Systems: Modeling, Analysis and Control*. PhD thesis, Massachusetts Institute of Technology, 1995.
- [19] M. Branicky. Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):475–482, April 1998.
- [20] M. Branicky, V. Borkar, and S. Mitter. A unified framework for hybrid control: Model and optimal control theory. *IEEE Transactions on Automatic Control*, 43(1):31–45, January 1998.

- [21] R. W. Brockett. Control theory and analytical mechanics. In C. Martin and R. Hermann, editors, *Geometric Control Theory*, Lie Groups: History, Frontiers and Applications, pages 1–46. Math. Sci. Press, 1977.
- [22] R. W. Brockett. Global descriptions of nonlinear control problems; vector bundles and nonlinear control theory. manuscript, 1980.
- [23] R.W. Brockett. Hybrid models for motion control systems. In H. Trentelman and J.C. Willems, editors, *Perspectives in Control*, pages 29–54. Birkhauser, Boston, 1993.
- [24] P. E. Caines and Y.J. Wei. Hierarchical hybrid control systems. In S. Morse, editor, *Control Using Logic Based Switching*, volume 222 of *Lecture Notes in Control and Information Sciences*, pages 39–48. Springer Verlag, 1996.
- [25] P.E. Caines and Y.J. Wei. The hierarchical lattices of a finite state machine. *Systems and Control Letters*, 25:257–263, 1995.
- [26] 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.
- [27] B. Carlson and V. Gupta. Hybrid CC with interval constraints. In T. Henzinger and S. Sastry, editors, *Hybrid Systems : Computation and Control*, volume 1386 of *Lecture Notes in Computer Science*, pages 80–95. Springer Verlag, Berlin, 1998.
- [28] K. Cerans and J. Viksna. Deciding reachability for planar multi-polynomial systems. In R. Alur, T. Henzinger, and E.D. Sontag, editors, *Hybrid Systems III*, volume 1066 of *Lecture Notes in Computer Science*, pages 389–400. Springer Verlag, Berlin, Germany, 1996.
- [29] A. Church. Logic, arithmetic, and automata. In *Proceedings of the International Congress of Mathematics*, pages 23–35, 1962.
- [30] G.E. Collins and H. Hong. Partial cylindrical algebraic decomposition for quantifier elimination. *Journal of Symbolic Computation*, 12:299–328, September 1991.
- [31] P. Cousot and R. Cousot. Systematic design of program analysis framework. In *Proceedings of the 6th ACM Symposium on Principles of Programming Languages*, 1979.

- [32] 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.
- [33] J. Davoren. Topologies, continuity, and bisimulations. Technical report, Cornell University, Ithaca, NY, 1998.
- [34] C. Daws, A. Olivero, S. Tripakis, and S. Yovine. The tool KRONOS. In *Hybrid Systems III*, volume 1066 of *Lecture Notes in Computer Science*, pages 208–219. Springer-Verlag, 1996.
- [35] J. Denef and L. van den Dries. p-adic and real subanalytic sets. *Annals of Mathematics*, 128:79–138, 1988.
- [36] A. Deshpande. *Control of Hybrid Systems*. PhD thesis, University of California at Berkeley, 1994.
- [37] A. Deshpande, A. Gollu, and L. Semenzato. The SHIFT programming language for dynamic networks of hybrid automata. *IEEE Transactions on Automatic Control*, 43(4):584–587, April 1998.
- [38] A. Dolzmann and T. Sturm. REDLOG : Computer algebra meets computer logic. *ACM SIGSAM Bulletin*, 31(2):2–9, June 1997.
- [39] P. M. Van Dooren. The generalized eigenstructure problem in linear system theory. *IEEE Transactions on Automatic Control*, 26(1):111–129, 1981.
- [40] A.F. Fillipov. *Differential Equations with Discontinuous Right Hand Sides*. Mathematics and Its Applications. Kluwer Academic Press, 1988.
- [41] R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, editors. *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*. Springer-Verlag, 1993.
- [42] R.M. Hardt. Stratifications of real analytic mappings and images. *Inventiones Mathematicae*, 28:193–208, 1975.
- [43] T. Henzinger, P. Ho, and H. Wong-Toi. Algorithmic analysis of nonlinear hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):540–554, April 1998.

- [44] T. Henzinger and S. Sastry, editors. *Hybrid Systems : Computation and Control*, volume 1386 of *Lecture Notes in Computer Science*. Springer-Verlag, 1998.
- [45] T.A. Henzinger. Hybrid automata with finite bisimulations. In Z. Fülöp and F. Gécseg, editors, *ICALP 95: Automata, Languages, and Programming*, pages 324–335. Springer-Verlag, 1995.
- [46] 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.
- [47] T.A. Henzinger, P.-H. Ho, and H. Wong-Toi. A user guide to HYTECH. In E. Brinksma, W.R. Cleaveland, K.G. Larsen, T. Margaria, and B. Steffen, editors, *TACAS 95: Tools and Algorithms for the Construction and Analysis of Systems*, volume 1019 of *Lecture Notes in Computer Science 1019*, pages 41–71. Springer-Verlag, 1995.
- [48] T.A. Henzinger, P.W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata? In *Proceedings of the 27th Annual Symposium on Theory of Computing*, pages 373–382. ACM Press, 1995.
- [49] T.A. Henzinger and H. Wong-Toi. Linear phase-portrait approximations for nonlinear hybrid systems. In R. Alur, T.A. Henzinger, and E.D. Sontag, editors, *Hybrid Systems III*, volume 1066 of *Lecture Notes in Computer Science*, pages 377–388. Springer-Verlag, 1996.
- [50] M. Heymann, F. Lin, and G. Meyer. Synthesis and viability of minimally interventive legal controllers for hybrid systems. *Discrete Event Dynamic Systems*, 8(2):105–136, June 1998.
- [51] A. Isidori. *Nonlinear Control Systems*. Springer-Verlag, second edition, 1989.
- [52] M. Johansson and A. Rantzer. Computation of piecewise quadratic Lyapunov functions for hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):555–559, April 1998.
- [53] J.F. Knight, A. Pillay, and C. Steinhorn. Definable sets in ordered structures. II. *Transactions of the American Mathematical Society*, 295(2):593–605, 1986.

- [54] W. Kohn, A. Nerode, J.B. Remmel, and X. Ge. Multiple agent hybrid control : Carrier manifolds and chattering approximations to optimal control. In *Proceedings of the 33th IEEE Conference on Decision and Control*, pages 4221–4227, Lake Buena Vista, FL, December 1994.
- [55] W. Kohn, A. Nerode, J.B. Remmel, and A. Yakhnis. Viability in hybrid systems. *Theoretical Computer Science*, 138(1):141–168, February 1995.
- [56] M. Kristic, I. Kanellakopoulos, and P. Kokotovic. *Nonlinear and Adaptive Control Design*. Adaptive and Learning systems for signal processing, communications and control. John Wiley & Sons, New York, 1995.
- [57] C.P. Kwong. Optimal chained aggregation for reduced order modeling. *International Journal of Control*, 35(6):965–982, 1982.
- [58] C.P. Kwong. Disaggregation, approximate disaggregation, and design of suboptimal control. *International Journal of Control*, 37(4):843–854, 1983.
- [59] C.P. Kwong and C.F. Chen. A quotient space analysis of aggregated models. *IEEE Transactions on Automatic Control*, 27(1):203–205, February 1982.
- [60] C.P. Kwong and Y.K. Zheng. Aggregation on manifolds. *International Journal of Systems Science*, 17(4):581–589, 1986.
- [61] G. Lafferriere, G. J. Pappas, and S. Sastry. Hybrid systems with finite bisimulations. In P. Antsaklis, W. Kohn, M. Lemmon, A. Nerode, and S. Sastry, editors, *Hybrid Systems V*, Lecture Notes in Computer Science. Springer Verlag, New York, 1998. To appear.
- [62] G. Lafferriere, G. J. Pappas, and S. Sastry. O-minimal hybrid systems. Technical Report UCB/ERL M98/29, University of California at Berkeley, Berkeley, CA, April 1998.
- [63] G. Lafferriere, G. J. Pappas, and S. Yovine. Decidable hybrid systems. Technical Report UCB/ERL M98/39, University of California at Berkeley, Berkeley, CA, June 1998.

- [64] G. Lafferriere, G.J. Pappas, and S. Sastry. Hybrid systems with finite bisimulations. Technical Report UCB/ERL M98/15, University of California at Berkeley, Berkeley, CA, April 1998.
- [65] G. Lafferriere, G.J. Pappas, and S. 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.
- [66] C. Loiseaux, S. Graf, J. Sifakis, A. Bouajjani, and S. Bensalem. Property preserving abstractions for the verification of concurrent systems. In *Formal Methods in Systems Design*, volume 6, pages 1–35. Kluwer Academic Publishers, Boston, 1995.
- [67] J. Lygeros. *Hierarchical, Hybrid Control of Large Scale Systems*. PhD thesis, University of California at Berkeley, 1996.
- [68] J. Lygeros, D.N. Godbole, and S. Sastry. Verified hybrid controllers for automated vehicles. *IEEE Transactions on Automatic Control*, 43(4):522–539, April 1998.
- [69] J. Lygeros, C. Tomlin, and S. Sastry. On controller synthesis for nonlinear hybrid systems. In *Proceedings of the 37th IEEE Conference on Decision and Control*, Tampa, FL, 1998.
- [70] John Lygeros, Datta N. Godbole, and Shankar Sastry. A game theoretic approach to hybrid system design. In Rajeev Alur, Thomas A. Henzinger, and Eduardo D. Sontag, editors, *Hybrid Systems III*, number 1066 in LNCS, pages 1–12. Springer Verlag, 1996.
- [71] John Lygeros, Datta N. Godbole, and Shankar Sastry. Optimal control approach to multiagent, hierarchical system verification. In *IFAC World Congress*, pages 389–394, San Fransisco, California, USA, June 30 - July 5 1996.
- [72] N. Lynch, R. Segala, F. Vaandrager, and H.B. Weinberg. Hybrid I/O automata. In *Hybrid Systems III*, volume 1066 of *Lecture Notes in Computer Science*, pages 496–510. Springer-Verlag, 1996.
- [73] A. Macintyre and A.J. Wilkie. On the decidability of the real exponential field. In *Kreiseliana: About and around Georg Kreisel*, pages 441–467. A.K. Peters, 1996.

- [74] O. Maler, editor. *Hybrid and Real-Time Systems*, volume 1201 of *Lecture Notes in Computer Science*. Springer-Verlag, 1997.
- [75] O. Maler, A. Pnueli, and J. Sifakis. On the synthesis of discrete controllers for timed systems. In E.W. Mayr and C. Puech, editors, *STACS 95: Theoretical Aspects of Computer Science*, volume 900 of *Lecture Notes in Computer Science*, pages 229–242. Springer-Verlag, 1995.
- [76] Z. Manna and A. Pnueli. *Temporal Verification of Reactive Systems: Safety*. Springer Verlag, New York, 1995.
- [77] L.S. Martin and P.E. Crouch. Controllability on principal fibre bundles with compact structure group. *Systems & Control Letters*, 5(1):35–40, 1984.
- [78] M.D. Mesarovic. *Theory of hierarchical, multilevel, systems*, volume 68 of *Mathematics in Science and Engineering*. Academic Press, New York, 1970.
- [79] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.
- [80] S. Morse, editor. *Control Using Logic-Based Switching*, volume 222 of *Lecture Notes in Control and information sciences*. Springer-Verlag, 1997.
- [81] A. Nerode and W. Kohn. Models for hybrid systems: Automata, topologies, controllability, observability. In R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, editors, *Hybrid Systems*, pages 317–356. Springer Verlag, New York, 1993.
- [82] H. Nijmeijer and A.J. van der Schaft. *Nonlinear Dynamical Control Systems*. Springer-Verlag, 1990.
- [83] A. Olivero, J. Sifakis, and S. Yovine. Using abstractions for the verification of linear hybrid systems. In *Computer Aided Verification*, volume 818 of *Lecture Notes in Computer Science*, pages 81–94. Springer-Verlag, July 1994.
- [84] C. C. Paige. Properties of numerical algorithms related to computing controllability. *IEEE Transactions on Automatic Control*, AC-26(1):111–129, 1981.
- [85] G. J. Pappas, G. Lafferriere, and S. Sastry. Hierarchically consistent control systems. Technical Report UCB/ERL M98/16, University of California at Berkeley, Berkeley, CA, April 1998.

- [86] G. J. Pappas and S. 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.
- [87] G. J. Pappas and S. Sastry. Straightening out rectangular differential inclusions. *Systems and Control Letters*, 35(2):79–85, September 1998.
- [88] G. J. Pappas, C. Tomlin, J. Lygeros, D. N. Godbole, and S. Sastry. A next generation architecture for air traffic management systems. In *Proceedings of the 36th IEEE Conference on Decision and Control*, pages 2405–2410, San Diego, CA, December 1997.
- [89] G.J. Pappas, G. Lafferriere, and S. Sastry. Hierarchically consistent control systems. In *Proceedings of the 37th IEEE Conference in Decision and Control*. Tampa, FL, December 1998.
- [90] A. Puri, V. Borkar, and P. Varaiya. ϵ -approximation of differential inclusions. In R. Alur, T.A. Henzinger, and E.D. Sontag, editors, *Hybrid Systems III*, volume 1066 of *Lecture Notes in Computer Science*, pages 362–376. Springer-Verlag, 1996.
- [91] A. Puri and P. Varaiya. Decidability of hybrid systems with rectangular differential inclusions. In *Computer Aided Verification*, pages 95–104, 1994.
- [92] A. Puri and P. Varaiya. Verification of hybrid systems using abstractions. In P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, editors, *Hybrid Systems II*, volume 999 of *Lecture Notes in Computer Science*, pages 359–369. Springer-Verlag, 1995.
- [93] 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.
- [94] H.H. Rosenbrock. *State Space and Multivariable Theory*. Jon Wiley, New York, 1970.
- [95] S. Sastry, G. Meyer, C. Tomlin, J. Lygeros, D. Godbole, and G. Pappas. Hybrid control in air traffic management systems. In *Proceedings of the 1995 IEEE Conference in Decision and Control*, pages 1478–1483, New Orleans, LA, December 1995.

- [96] M. Spivak. *A Comprehensive Introduction to Differential Geometry*. Publish or Perish, 1979.
- [97] H. J. Sussmann. Subanalytic sets and feedback control. *Journal of Differential Equations*, 31(1):31–52, January 1979.
- [98] A. Tarski. *A decision method for elementary algebra and geometry*. University of California Press, second edition, 1951.
- [99] L. Tavernini. Differential automata and their discrete simulators. *Nonlinear Analysis: Theore, Methods, and Applications*, 11(6):665–683, 1987.
- [100] W. Thomas. Automata on infinite objects. In *Formal Models and Semantics, volume B of Handbook of Theoretical Computer Science*. Elsevier Science, 1990.
- [101] M. Tittus and B. Egardt. Control design for integrator hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):491–500, April 1998.
- [102] C. Tomlin, G. J. Pappas, and S. Sastry. Conflict resolution for air traffic management : A study in multi-agent hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):509–521, April 1998.
- [103] Claire Tomlin, John Lygeros, and Shankar Sastry. Synthesizing controllers for nonlinear hybrid systems. In S. Sastry and T.A. Henzinger, editors, *Hybrid Systems: Computation and Control*, number 1386 in Lecture Notes in Computer Science, pages 360–373. Springer Verlag, 1998.
- [104] D. van Dalen. *Logic and Structure*. Springer-Verlag, third edition, 1994.
- [105] L. van den Dries. Remarks on Tarski’s problem concerning $(\mathbb{R}, +, \cdot, \exp)$. In G. Lolli, G. Longo, and A. Marcja, editors, *Logic Colloquium '82*, pages 97–121. Elsevier Science Publishers B.V., 1984.
- [106] L. van den Dries. *Tame Topology and o-minimal structures*. Cambridge University Press, 1998.
- [107] L. van den Dries and C. Miller. On the real exponential field with restricted analytic functions. *Israel Journal of Mathematics*, 85:19–56, 1994.

- [108] A.J. van der Schaft and J.M. Schumacher. Complementarity modeling of hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):483–490, April 1998.
- [109] P. Varaiya. Smart cars on smart roads: problems of control. *IEEE Transactions on Automatic Control*, 38(2):195–207, 1993.
- [110] V. Weispfenning. A new approach to quantifier elimination for real algebra. Technical Report MIP-9305, Universität Passau, Germany, July 1993.
- [111] A. J. Wilkie. Model completeness results for expansions of the ordered field of real numbers by restricted pfaffian functions and the exponential function. *Journal of the American Mathematical Society*, 9(4):1051–1094, Oct 1996.
- [112] H.S. Witsenhausen. A class of hybrid-state continuous-time dynamics systems. *IEEE Transactions on Automatic Control*, 11:161–167, February 1966.
- [113] K.C. Wong and W.M. Wonham. Hierarchical control of discrete-event systems. *Discrete Event Dynamic Systems*, 6:241–273, 1995.
- [114] K.C. Wong and W.M. Wonham. Hierarchical control of timed discrete-event systems. *Discrete Event Dynamic Systems*, 6:275–306, 1995.
- [115] H. Wong-Toi. The synthesis of controllers for linear hybrid automata. In *Proceedings of the 36th IEEE Conference on Decision and Control*, San Diego, CA, December 1997.
- [116] W.M. Wonham. *Linear Multivariable Control : A Geometric Approach*, volume 10 of *Applications of Mathematics*. Springer-Verlag, New York, 1985.
- [117] H. Ye, A.N. Michel, and L. Hou. Stability theory for hybrid dynamical systems. *IEEE Transactions on Automatic Control*, 43(4):461–474, April 1998.
- [118] M. Zefran and J. Burdick. Stabilization of systems with changing dynamics. In T. Henzinger and S. Sastry, editors, *Hybrid Systems : Computation and Control*, volume 1386 of *Lecture Notes in Computer Science*, pages 400–415. Springer Verlag, Berlin, 1998.

- [119] H. Zhong and W.M. Wonham. On the consistency of hierarchical supervision in discrete-event systems. *IEEE Transactions on Automatic Control*, 35(10):1125–1134, 1990.

Appendix A

Appendix

A.1 Implementation of Algorithms 5.37 and 5.38

```

function [controllable]=HCA(A,B,k,tol)
%*****
% Hierarchical Controllability Algorithm 5.37
%
% Required Inputs : System Matrices A,B,
%                   Integer 0<= k <= n-1 (k=0 is Algorithm 5.38)
% Optional Inputs : Tolerance threshold tol (used for rank computation)
%*****

n=size(A,1);
if nargin == 3
    tol = n*norm(A,1)*eps;
end
r = rank(B,tol);          %*** Dimension of input space

while ( (n>r) & (r>0) ), %*** If inputs exist and are less than states
    l = min(k,n-1);      %*** Ignore Lie brackets higher than n-1
    W = B;                %*** Compute [B AB ...A^kB]
    for j=1:l,
        W = [B A*W];
    end
    [U,S,V] = svd(W);    %*** Obtain consistent matrix C
    m = rank(S,tol);
    U1 = U(:,1:m) ;
    U2 = U(:,(m+1):n) ;
    C = U2';
    B = C*A*U1;          %*** Obtain consistent abstraction
    A = C*A*C';
    n = size(A,1)        %*** Dimension of abstracted system
    r = rank(B,tol);     %*** Dimension of macroinputs
end

if (n==r) controllable=1;
elseif (r==0) controllable=0;
end

```

```

function [controllable]=HCA(A,B,tol)
%*****
% Hierarchical Controllability Algorithm 5.38
%
% Function Call   : HCA(A,B,tol)
% Required Inputs : System Matrices A,B
% Optional Input  : Tolerance threshold tol
%*****

n=size(A,1);
if nargin == 2
    tol = n*norm(A,1)*eps;
end

[U,S,V] = svd(B);          %*** Dimension of input space
r = rank(S,tol);

while ( (n>r) & (r>0) ),   %*** If inputs exist and are less than states
    U1 = U(:,1:r) ;        %*** Obtain consistent matrix C
    U2 = U(:,(r+1):n) ;
    C = U2';
    B = C*A*U1;           %*** Obtain consistent abstracted system
    A = C*A*C';
    n = size(A,1);        %*** Dimension of abstracted system
    [U,S,V] = svd(B);
    r = rank(S,tol);      %*** Dimension of macroinputs
end

if (n==r)
    controllable=1;
elseif (r==0)
    controllable=0;
end
end

```