MODEL CHECKING FOR HYBRID SYSTEMS

by

Mireille Broucke

Memorandum No. UCB/ERL M00/36

22 June 2000

MODEL CHECKING FOR HYBRID SYSTEMS

by

Mireille Broucke

Memorandum No. UCB/ERL M00/36

22 June 2000

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

MODEL CHECKING FOR HYBRID SYSTEMS

MIREILLE BROUCKE

ABSTRACT. The goal of this paper is to extend model checking to hybrid systems with non-trivial continuous dynamics. Our approach is to generalize the results of Alur and Dill by examining the geometric structure of the bisimulation for timed automata. This enables us to present a new methodology for obtaining finite bisimulations. We demonstrate the meaningfulness of the method through several applications: coordinated robots, coordinated aircraft, and hybrid systems with linear dynamics. The method allows the characterization of the symbolic execution theory, making possible the development of algorithms for symbolic model checking.

Keywords: hybrid automata, model checking, bisimulation, coordinated autonomous agents, symbolic model checking

1. INTRODUCTION

The goal of this paper is to extend model checking to hybrid systems. Our approach is to generalize the work of Alur and Dill on timed automata [1] to hybrid systems with non-trivial continuous dynamics. We obtain results on bisimulations of hybrid automata by examining the geometric nature of the bisimulation of timed automata. This gives a new method to construct bisimulations, under a suitable compatibility condition on the enabling and reset conditions and in the process we obtain new decidability results for the examples that are considered: coordinated aircraft, coordinated robots, and hybrid systems with linear dynamics.

First let us briefly review the history of model checking. Verification was introduced for finite state programs to determine automatically if the states of the program satisfy a specification. A *safety* requirement ensures that a system does not exhibit some undesired behavior. The complement is a *liveness* requirement: that the system exhibit some desired behavior. Pnueli proposed the use of temporal logic for the specification of safety and liveness requirements [24]. The algorithmic verification of finite-state systems was started in 1981 by Clarke and Emerson [7] and by Sifakis [26]. The procedure is to convert a finite state program to a finite graph M. Given a temporal logic formula ϕ, the verification question is: do all sequences defined by paths through M satisfy ϕ? The problem is termed *model checking* because we want to know if M is a model of ϕ. Vardi and Wolper [31] showed that one can construct a Buchi automaton that accepts sequences satisfied by formulas of PLTL (Propositional linear-time logic). If the program is viewed as a finite state generator M and the specification ϕ as a finite state acceptor, then the model checking problem is reduced to the automata-theoretic question of whether the language $L(M) - L(\phi)$ is empty, where $L(M)$ is the language generated by M and $L(\phi)$ is the language accepted by ϕ.

For hybrid systems, model checking is performed by *abstracting* the system to obtain a finite quotient system. Bisimulation is the main step in constructing the quotient system.

The need for new results on bisimulation is evident in three areas. First, modeling checking has been announced as a method that can supplant simulation in the design of concurrent systems [8]. In order to make this claim realizable for hybrid systems, model checking must be able to handle non-trivial dynamics. At present, it is incapable of doing so. Our positive results give encouragement to press ahead with a program of model checking for hybrid systems. Second, although state space partitions have been an underlying assumption in several separate research efforts such as [28] and [5], no method to obtain partitions was given. Our results show that a comprehensive methodology is within reach. Finally, model checking and the related problem of controller synthesis are able to address problems that control theory has been unable to address because of a lack of expressiveness of control theoretic models. In particular, temporal logic enables a rich characterization of transient behavior in time, when the system operates in a *reactive* mode with its environment, in contrast with control theoretic specifications which have mostly focused on input-output behavior. Temporal specifications also express communication and coordination requirements of multiple agents.

Inspite of this, few results on obtaining bisimulations or constructing partitions are available. We summarize those works we are aware of. The approach of [22] requires an iterative scheme to compute the bisimulation and is built up from the theory of ominimal structures. While their work is theoretically appealing, we feel there is a simpler way to go about things, using easier concepts and more computationally attractive methods. We obtain an analytical description of bisimulation that can be understood as a gestalt. This has the intuitive benefit that it is an immediate extension of the approach for timed automata. It relies on concepts that are accessible to computer scientists and familiar in geometric control theory [15], namely local coordinate transformations. Moreover the analytical description enables us to define the symbolic execution theory for the hybrid automaton. The method of [13] uses an over-approximation of vector fields by differential inclusions. At present obtaining the inclusions and the region over which it is valid is ad-hoc. The method is easiest to implement in two-dimensions. We take the contrasting view that the vector fields capture important information about the model, which the designer has taken some trouble to identify, but the enabling and reset conditions are design parameters that may be specified to the computational benefit of model checking.

In summary, we believe this work represents a breakthrough for model checking of hybrid systems and we anticipate exciting new possibilities for extending its applications.

The paper is organized as follows. In section 2 we define the syntax and semantics of hybrid automata. Section 3 gives the main results on constructing bisimulations. In section 4 we consider the benefit of using exterior differential systems for determining bisimulations and for parallel composition. We present applications in section 5 and discuss implementation of the method in section 6. Finally, we indicate future directions of research in section 7.

2. Hybrid Automata

Notation x' refers to the updated value of a variable x after a transition is taken. All manifolds, vector fields, curves and maps are of class C^∞. Manifolds are assumed to be connected, paracompact, and Hausdorff. $C^\infty(M)$, $\mathcal{X}(M)$, and $\Omega^k(M)$ denote the sets of smooth real-valued functions, smooth vector fields, and k-forms defined on a manifold M.

A *hybrid system* is a dynamical system consisting of one or more components called hybrid automata. A *hybrid automaton* is a tuple $A = (Q, \Sigma, D, E, I, G, R)$ with the following elements:

State space: $Q = L \times M$ consists of a finite set L of control locations and n continuous variables $x \in M$, where M is an n-dimensional differentiable manifold.

Events: Σ is a finite observation alphabet.

Vector fields: $D : L \to \mathcal{X}(M)$ is a function assigning an autonomous vector field to each location. We will use the notation $D(l) = f_l$. For location l, the dynamics are given by $\dot{x} = f_l(x)$.

Control switches: E is a set of control switches. $e = (l, \sigma, l')$ is a directed edge between a source location l and a target location l' with observation $\sigma \in \Sigma$.

Invariant conditions: $I : L \to 2^M$ is a function assigning for each location an invariant condition on the continuous states. The invariant condition $I(l) \subset M$ restricts the region on which the continuous states can evolve for location l.

Enabling conditions: $G : E \to \{g_e\}_{e \in E}$ is a function assigning to each edge an enabling (or guard) condition $g \subseteq I(l)$. We use the notation $G(e) = g_e$.

Reset conditions: $R : E \to \{r_e\}_{e \in E}$ is a function assigning to each edge a reset condition, $r_e : M \to 2^M$, where we use the notation $R(e) = r_e$. We assume $r_e(g_e) \subseteq I(l')$ for each $e = (l, \sigma, l') \in E$.

2.1. **Semantics.** A state is a pair (l, x) satisfying $x \in I(l)$. $E(l)$ denotes the set of edges possible at $l \in L$. Trajectories of A evolve in *steps* of two types. A *σ-step* is a binary relation $\xrightarrow{\sigma} \subset Q \times Q$ and we write $(l, x) \xrightarrow{\sigma'} (l', x')$ iff (1) $e = (l, \sigma', l') \in E$, (2) $x \in g_e$, and (3) $x' \in r_e(x)$. Define $\phi_t^l(x)$ to be the trajectory in location l, starting from x and evolving for time t. A *t-step* is a binary relation $\xrightarrow{t} \subset Q \times Q$, and we write $(l, x) \xrightarrow{t} (l', x')$ iff (1) $l = l'$ and (2) for $t \geq 0$, $x' = \phi_t(x, \sigma)$, where $\dot{\phi}_t(x) = f_l(\phi_t(x, \sigma), \sigma)$. A *trajectory* π of A is a finite or infinite sequence of the form $\pi : q_0 \xrightarrow{\tau_0} q_1 \xrightarrow{\tau_1} q_2 \xrightarrow{\tau_2} \dots$ where for all $i \geq 0$, $q_i \in Q, \tau_i \in \Sigma \cup \mathbb{R}^+$. We assume the trajectories are *non-Zeno*; that is, every trajectory of A admits a finite number of σ-steps in any bounded time interval.

2.2. **Example.** Consider the hybrid automata of Figure 1. The invariants for locations l_1, l_2, l_3 are $x \geq 1, |x| \leq 1, x \leq -1$, respectively. The dynamics in each location are either affine linear or linear. It has been shown that this hybrid automaton has

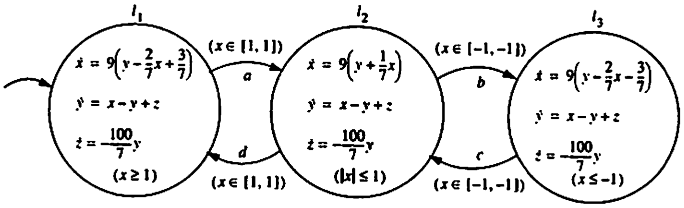
l_1
l_2
l_3
$\dot{x} = 9\left(y - \frac{2}{7}x + \frac{3}{7}\right)$
$\dot{y} = x - y + z$
$\dot{z} = -\frac{100}{7}y$
$(x \geq 1)$
$\dot{x} = 9\left(y + \frac{1}{7}x\right)$
$\dot{y} = x - y + z$
$\dot{z} = -\frac{100}{7}y$
$(|x| \leq 1)$
$\dot{x} = 9\left(y - \frac{2}{7}x - \frac{3}{7}\right)$
$\dot{y} = x - y + z$
$\dot{z} = -\frac{100}{7}y$
$(x \leq -1)$
$(x \in [1, 1])$
a
d
$(x \in [1, 1])$
$(x \in [-1, -1])$
b
c
$(x \in [-1, -1])$

FIGURE 1. Double scroll hybrid automaton.

a homoclinic orbit and by Shilnikov's theorem the system has a Smale horseshoe implying the existence of a chaotic attractor [6].

3. BISIMULATION

The concept of bisimulation was introduced by D. Park [23] in the context of concurrent processes. Let λ represent an arbitrary time passage. Given the hybrid system A, a *bisimulation* of A is a binary relation $\simeq \subset Q \times Q$ satisfying the condition that for all states $p, q \in Q$, if $p \simeq q$ and $\sigma \in \Sigma \cup \{\lambda\}$, then
(1) if $p \xrightarrow{\sigma} p'$, then there exists q' such that $q \xrightarrow{\sigma} q'$ and $p' \simeq q'$, and
(2) if $q \xrightarrow{\sigma} q'$, then there exists p' such that $p \xrightarrow{\sigma} p'$ and $p' \simeq q'$.

Game-theoretic: Bisimulation can be interpreted as a game between an automaton and its environment [11]. In this view, the protagonist and the environment start at two states and take σ- or t- steps, each time recording an *observation.* The environment uses non-determinism advantageously to select a step the protagonist cannot match. If the protagonist matches the observations of the environment, the states are bisimilar.

Topological: Stemming from fundamental work by McKinsey and Tarski [29], bisimulation was interpreted as a form of topological continuity by Jennifer Davoren [9].

Constructive: A constructive view and also a definition is that bisimulation is the *coursest stable refinement* of an *observation equivalence* [19]. One uses a Paige-Tarjan type refinement algorithm [21] such that the fixpoint of the algorithm gives the bisimulation partition. See for instance, [10], [22].

Algebraic: In an algebraic sense, bisimulation is a *congruence*; that is, an equivalence relation closed under concatenation, where by concatenation we mean successive σ- or t-steps of A.

Let $Q_{\simeq}$ be the set of equivalence classes of $\simeq$. A bisimulation is finite if it has a finite number of equivalence classes. Using $\simeq$, a quotient system $A_{\simeq}$ is constructed. If $\simeq$ is finite, the quotient system is the finite automaton

$$A_{\simeq} = (Q_{\simeq}, \Sigma \cup \lambda, E_{\simeq}).$$

$Q_{\simeq} = L \times M/_{\simeq}$ are the cosets of $\simeq$. $q \in Q_{\simeq}$ is written $q = [(l, x)]$ for some $l \in L$, $x \in M$ where $(l, x) \in q$. The transitions of $A_{\simeq}$, defined by $E_{\simeq}$ and denoted $\rightarrow_{\simeq}$, are as follows. For $q = [(l, x)], q' = [(l', x')]$, $q \rightarrow_{\simeq} q'$ iff there exists $(l, y) \in q$ and

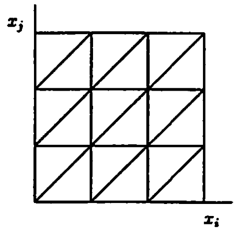

FIGURE 2. Bisimulation for timed automata.

$(l', y') \in q'$ such that $(l, y) \to (l', y')$ is either a t-step or a σ-step of A (for t-steps, q and q' are contiguous). $A_{\simeq}$ is referred to as an *abstraction* of A.

Remark 3.1.

1. The importance of $A_{\simeq}$ is that it captures the salient features of the dynamics of A in a time abstract model. The abstraction of time enables the reduction from an infinite state space to a finite one. If the bisimulation is not finite the reduction can still be done to an infinite automaton, and model checking algorithms can be applied, but they may not terminate.
2. If initial conditions Q^0 or final conditions Q^f are specified with A, then these sets are quotiented by $\simeq$ as well.

Example 3.1. Though timed automata have been written about extensively, it is illuminating to examine their bisimulation with a geometric lens. Timed automata are a subclass of hybrid automata in which the continuous dynamics define the *clock flow*, $\dot{x}_i = 1$. Enabling and reset conditions are built up from finite conjunctions and disjunctions of the formulas $x\%c$ where $\% \in \{<, \leq, =, >, \geq\}$ and $c \in \mathbb{Z}$.

For $y \in \mathbb{R}$, let $\lfloor y \rfloor$ be its integer part and $\langle y \rangle$ its fractional part. Let L be the set of locations, $x \in \mathbb{R}^n$ the clock variables, and m_i the largest integer the ith clock is compared to in an enabling condition. We say $(l, x) \simeq (l', x')$ iff (1) $l = l'$, (2) for all $i = 1, \ldots, n$, $x_i > m_i$ iff $x'_i > m_i$, or $\lfloor x_i \rfloor = \lfloor x'_i \rfloor$, (3) for every $x_i, \leq m_i$ and $x_j \leq m_j$, $\langle x_i \rangle \leq \langle x_j \rangle$ iff $\langle x'_i \rangle \leq \langle x'_j \rangle$ and $\langle x_i \rangle = 0$ iff $\langle x'_i \rangle = 0$.

Figure 2 shows the bisimulation for timed automata projected to the $x_i - x_j$ plane. We observe the following features:

1. The bisimulation is defined on a compact region of the state space where the interesting dynamics occur. Outside this region, the dynamics are sufficiently benign that they can be handled by one equivalence class.
2. The partition is described as a gestalt rather than as an iterative procedure that terminates at a fixpoint.
3. The partition uses hypersurfaces that are either invariants or transversals of the flow to build up equivalence classes.
4. The hypersurfaces are propitiously selected to be compatible with the syntax of the enabling, reset, initial and final conditions. That is, the syntax of timed automata does not imply a further refinement of the "proposed" partition.

5. The hypersurfaces are defined by analytical expressions. The atomic expressions provide an alternative description of the bisimulation [8, p.280], and can be used to define the *symbolic execution theory* [12].

3.1. Stable partitions and compatibility.

In this section we develop the construction of bisimulations for hybrid automata using our geometric insights. First, we show how a concept of stable partitions with respect to a flow combined with a natural compatibility condition on the enabling and reset conditions leads to a bisimulation. This step is rather straightforward. Assuming the compatibility conditions are met, there is only constructing stable partitions.

For each $l \in L$, let $\simeq^l$ be an equivalence relation on $l \times M$ and let P^l be the partition defined by $\simeq^l$. We say P^l is a *stable partition of the flow* ϕ^l or $\simeq^l$ defines a *stable partition of the flow* ϕ^l if $(l,x) \simeq^l (l,x')$ implies that for all $y \in M$, $t \geq 0$, if $y = \phi^l_t(x)$, then there exists $y' \in M$ and $t' \geq 0$ such that $y' = \phi^l_{t'}(x')$ and $(l,y) \simeq^l (l,y')$.

Let $e = (l,\sigma,l') \in E$ and $\mathcal{P} = \{P^l \mid l \in L\}$, a set of stable partitions defined by equivalence relations $\{\simeq^l\}_{l \in L}$. Given $\simeq^l$ at $l \in L$, we say g_e *is compatible with* $\simeq^l$ if $(l,x) \in \{l\} \times g_e$ implies $[(l,x)] \in \{l\} \times g_e$. That is, the enabling condition is a union of cosets of $\simeq^l$. Similarly we say Q^0 *is compatible with* $\simeq^l$ if $(l,x) \in Q^0$ implies $[(l,x)] \in Q^0$. The analogous definition applies to Q^f. For $e = (l,\sigma,l')$ we say that r_e *is compatible with* $\simeq^{l'}$ if $(l',x') \in \{l'\} \times r_e(x)$ implies $[(l',x')] \in \{l'\} \times r_e(x)$, and $[(l,x)] = [(l,x')]$ implies $r_e(x) = r_e(x')$. Finally, we say A *is compatible with* $\{\simeq^l\}$ if for each $e \in E$, g_e and r_e are compatible with $\simeq^l$, $\simeq^{l'}$, respectively, and for each $l \in L$, Ω^l is compatible with $\simeq^l$, and Q^0 and Q^f are compatible with $\{\simeq^l\}$.

Lemma 3.1. *Given hybrid automaton H and $\{\simeq^l\}$ defining a set of stable partitions with respect to the flows of H, suppose H is compatible with $\{\simeq^l\}$. Then $\simeq \subset Q \times Q$ defined by: $(l,x) \simeq (l',x')$ iff (1) $l = l'$, and (2) $(l,x) \simeq^l (l',x')$, is a bisimulation for H.*

Proof. Let $\simeq$ be an equivalence relation satisfying conditions (1) and (2) and suppose $(l,x) \simeq (l',x')$. This implies $l = l'$ and $(l,x) \simeq^l (l,x')$.

Suppose $(l,x) \xrightarrow{t} (l,y)$ is a t-step of A. Because $\simeq^l$ defines a stable partition, there exists $y' \in M$ and $t' \geq 0$ such that $y' = \phi^l_{t'}(x')$ and $(l,y) \simeq^l (l,y')$. Hence $(l,y) \simeq (l,y')$.

Suppose $(l,x) \xrightarrow{\sigma} (\tilde{l},y)$ is a σ-step of A. This implies $x \in g_e$ for $e = (l,\sigma,\tilde{l})$. Since g_e is compatible with $\simeq^l$, $x' \in g_e$. Since r_e is compatible with $\simeq^{l'}$ we can find $y' \in r_e(x')$ such that $[(\tilde{l},y)] = [(\tilde{l},y')]$, since $r_e(x) = r_e(x')$. Hence $(\tilde{l},y) \simeq (\tilde{l},y')$. □

Remark 3.2.

1. The definition of stable partition says two equivalent points *can* visit the same next equivalence class, not that they will. Thus, it applies to differential inclusions as well as vector fields.
2. The compatibility definitions are the natural ones to ensure that bisimulation is preserved over σ-steps. One could also take the view that the enabling and reset

conditions are given in an arbitrary form. For safety controller synthesis starting from *bad* states, to obtain the bisimulation one over-approximates the enabling and reset conditions by compatible ones. For reachability analyses starting from *good* states, one under-approximates. The approximative view is described in [2].

3. If initial and final regions are specified with A, then they must also satisfy a compatibility condition with the partitions.

3.2. Foliations and first integrals.

We build stable partitions using foliations, flow boxes and first integrals. We assume knowledge of some differential geometry (see [32]).

Given an n-dimensional manifold M a smooth *foliation* of dimension p or codimension $q = n - p$ is a collection of disjoint connected subsets $F = \{s_\alpha\}$ whose disjoint union forms a partition of M. The foliation satisfies the property that each point of M has a neighborhood U and a system of coordinates $y : U \to \mathbb{R}^p \times \mathbb{R}^q$ such that for each s_α, the (connected) components of $(U \cap s_\alpha)$ are given by

$$\begin{aligned} y_{p+1} &= c_1 \\ &\vdots \\ y_{p+q} &= c_q \end{aligned}$$

where $c_i \in \mathbb{R}$. Each connected subset is called a *leaf* of the foliation, and each leaf is a submanifold of dimension p in M. See [16] for more background on foliations.

We want foliations whose leaves are regular submanifolds. The Pre-Image theorem [32, p. 31] provides a way to construct regular submanifolds, and, in particular, the pre-image of a submersion defines a foliation with regular leaves. A foliation globally defined by a submersion is called *simple.*

Let $f \in \mathcal{X}(M)$. We define two types of simple co-dimension one foliations with respect to f, called tangential and transversal foliations. For this we require a notion of transversality of foliations. Let TF be the field of tangent spaces to the leaves of F. A map $h : M \to N$ is *transverse* to a foliation F of N if for every $x \in M$, $h_* T_x M + T_{h(x)} F = T_{h(x)} N$, where h_* is the push-forward map of h. A submanifold W on M is *transverse* to foliation F of M if the inclusion map $i : W \to M$ is transverse to F. A foliation F' is said to be *transverse* to F if each leaf of F' is transverse to F. A foliation in general does not admit a transversal foliation, but for each $x \in M$ there exists a neighborhood of x such that F restricted to the neighborhood has a local transversal foliation. A *tangential foliation* F of M is a co-dimension one foliation that satisfies $f(x) \in T_x F, \forall x \in M$; that is, f is a cross-section of the tangent bundle of F. A *transversal foliation* $F_\perp$ of M is a co-dimension one foliation that satisfies $f(x) \notin T_x F, \forall x \in M$.

Let $\{F_i\}$ be a collection of $n - 1$ tangential foliations on $U \subset M$ and one transversal foliation $F_n := F_\perp$ on U, which, additionally, satisfies a *regularity condition:* for each

$x \in M$, $T_xF_1 + \cdots + T_xF_n = \mathbb{R}^n$. For simple foliations the following lemma provides an algebraic test for regularity.

Lemma 3.2. *Let M be an n-dimensional manifold and define $h_i : M \to \mathbb{R}, i = 1, ...n$, a collection of submersions on M. If dh_i are linearly independent on $U \subset M$, then the foliations defined by $h^{-1}(\mathbb{R})$ are independent on U.*

We will not use all of the leaves of a foliation, but a finite subset of them. We *discretize* a foliation as follows. Let $h : M \to \mathbb{R}$ be the submersion of a simple co-dimension one foliation F. Given an interval $[a, b]$, a gridsize $\Delta = \frac{b-a}{2^k} > 0$ with $k \in \mathbb{Z}^+$, define the finite collection of points $C_k = \{a, a + \Delta, \dots, b\}$. Then, $h^{-1}(C_k)$ is the discretization of F on $h^{-1}([a, b])$.

A bisimulation can be constructed using foliations by elaborating the following steps:

1. Find $(n-1)$ simple co-dimension one tangential foliations on $U \subset M$, for each $f_l, l \in L$.
2. Construct either a local or global (on U) transversal foliation for each f_l.
3. Check the regularity condition on U.
4. Discretize the foliations using a gridsize Δ.

To obtain tangential foliations we use local first integrals. A *first integral* of $\dot{x} = f(x)$ is a function $\Psi : M \to \mathbb{R}$ satisfying $L_f\Psi = 0$, where $L_f\Psi$ is the Lie derivative of Ψ along f.

Theorem 3.3 (Flow Box). *Let f be a vector field on M with $f(x) \neq 0$. Then there exist coordinates y defined on a neighborhood V of x such that*

$$f = \frac{\partial}{\partial y_n} \qquad \text{on } V.$$

Here is our main result on stable partitions.

Theorem 3.4. *Given $f \in \mathcal{X}(M)$, compact $U \subseteq M$, and coordinates y, if (y, U) is a flow box for X, there exists a stable partition with respect to f on U.*

Proof. By the Flow Box theorem, there exists a diffeomorphism $h : U \to V \subset \mathbb{R}^n$, where $V = [-1, 1]^n$, such that $\dot{x} = f(x)$ expressed in $y = h(x)$ coordinates is

$$\dot{y}_1 = 0, \dot{y}_2 = 0, ...\dot{y}_n = 1. \tag{3.1}$$

There exist $n - 1$ independent functions $y_1 = c_1, \dots, y_{n-1} = c_{n-1}$ that are first integrals of (3.1), and they define $(n-1)$ independent submanifolds, passing through each $y = (c_1, \dots, c_{n-1}, y_n)$. A submanifold transversal to the flow of (3.1) is given by $y_n = c_n$.

Fix $k \in \mathbb{Z}^+$ and let $\Delta = \frac{1}{2^k}$. Define

$$C_k = \{0, \pm\Delta, \pm 2\Delta, \dots, \pm 1\}. \tag{3.2}$$

Each $y_i = c$ for $c \in C_k$, $i = 1, \dots, n$ defines a hyperplane in $\mathbb{R}^n$ denoted $\tilde{W}_{i,c}$, and a submanifold $W_{i,c} = h^{-1}(\tilde{W}_{i,c})$. The collection of submanifolds is denoted

$$\mathcal{W}_k = \{\ W_{i,c} \mid c \in C_k, i \in \{1, \dots, n\}\ \}. \tag{3.3}$$

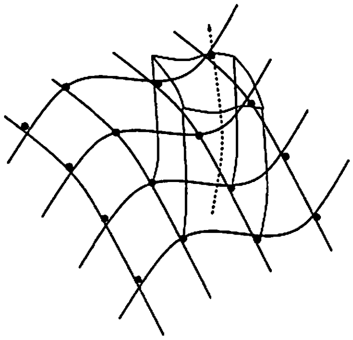

FIGURE 3. Partition for $\simeq$. The leaves of the tangential foliations form boundaries that are invariants of the flow.

$U \setminus \mathcal{W}_k$ is the union of $2^{n(k+1)}$ disjoint open sets $\mathcal{V}_k = \{V_j\}$.

We define an equivalence relation $\simeq^e$ on $\mathbb{R}^n$ as follows. $x \simeq^e x'$ iff
(1) $x \notin V$ iff $x' \notin V$, and
(2) if $x, x' \in V$, then for each $i = 1, \ldots, n$, $x_i \in (c, c + \Delta)$ iff $x' \in (c, c + \Delta)$, and $x_i = c$ iff $x'_i = c$, for all $c \in C_k$.

We define the equivalence relation $\simeq$ on $\{l\} \times M$ by $x \simeq x'$ iff $h(x) \simeq^e h(x')$. $\simeq$ is clearly a stable partition with respect to X^l because the invariant submanifolds enclose trajectories starting at equivalent points so that they can only visit the same next equivalence class. □

Remark 3.3.

1. One can show that the closure of an equivalence class of $\simeq$ is a union of equivalence classes of $\simeq$. This implies that the interior of an equivalence class is either the empty set or the class itself. The picture of $\simeq$ is something like Figure 3. The equivalence classes are the open line segments, points, interiors of cells, etc.
2. Suppose a stable partition has been constructed for a smooth vector field X on M using the steps outlined above. Let Y on N be a smooth vector field topologically conjugate to X; that is, there exists a homeomorphism h taking orbits of X through $x \in M$ to orbits of Y through $h(x) \in N$ and preserving the sense of the orbit. Then h can be used to construct a stable partition with respect to Y. First, if g is a first integral of X then $g \circ h^{-1}$ is a first integral of Y since $L_Y(g \circ h^{-1}) = d(g \circ h^{-1})(h_* X) = dg \cdot X = L_X g$. In this manner, tangential and transversal foliations are mapped through h to tangential and transversal foliations of Y, respectively. If the foliations of X are independent so are the foliations of Y. Also, since h maps fixed points of X to fixed points of Y, a stable partition defined on $U \subset M$ for X non-vanishing on U is well-defined for $h(U) \subset N$ and Y is non-vanishing on $h(U)$.

4. Exterior differential systems

A natural setting for finding first integrals is provided by exterior differential systems [27, 3]. Let $\Omega(M) = \oplus_{k=0}^{\infty}\Omega^k(M)$ with the wedge product $\wedge$ be the exterior algebra on M. $d : \Omega^k(M) \to \Omega^{k+1}(M)$ is the exterior derivative. Recall that $\omega \in \Omega^k(M)$ is *exact* if there exists an $\alpha \in \Omega^{k-1}(M)$ such that $\omega = d\alpha$. A set of independent one-forms $\omega^1, \dots, \omega^n$ generates a *Pfaffian system* $P = \{\omega^1, \dots, \omega^n\} = \{\sum f_k\omega^k | f_k \in C^\infty(M)\}$. The Pfaffian system satisfies the *Frobenius condition* if $d\omega^i$ is a linear combination of $\omega^1, \dots, \omega^n$.

Theorem 4.1 (Frobenius). *Let $P = \{\omega^1, \dots, \omega^n\}$ be a Pfaffian system with one-forms satisfying the Frobenius condition for $i = 1, \dots, n$. Then there exist coordinates $h_1, \dots, h_n$ such that $P = \{dh_1, \dots, dh_n\}$.*

In this case the Pfaffian system is said to be *completely integrable* and the h_i are the first integrals of P. Thus, the Frobenius theorem provides an alternative and equivalent route to existence of local first integrals as the Flow Box theorem. We have found it useful in applications to work with systems in Pfaffian form. Also, it is easy to state results about parallel composition of hybrid automata in terms of the vector fields in Pfaffian form. We give such a result next, but we remark that this is a first step: interesting extensions are possible using the theory of exterior differential systems.

4.1. Parallel composition.

Bisimulation for hybrid systems is, in general, not closed under parallel composition of automata. We give a sufficient condition on the Pfaffian form of the continuous dynamics of each control location so that if two hybrid automata have a finite bisimulation, then so does their parallel composition.

Suppose we have hybrid automata $A_i = (Q_i, \Sigma_i, D_i, E_i, I_i, G_i, R_i)$, $i = 1, 2$ with state spaces $Q_1 = L_1 \times M_1^n$ and $L_2 \times M_2^m$, respectively. We label the components of the continuous variables of A_1, $x_1, \dots, x_n$, and of A_2, $x_{n+1}, \dots, x_{n+m}$. The *parallel composition* of A_1 and A_2 is

$$A_1 \times A_2 = (Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, D, E, I, G, R)$$

$D : L_1 \times L_2 \to \mathcal{X}(M_1 \times M_2)$ assigns vector field $[f_l \; f_{l'}]^T$ to location (l, l'). $I : L_1 \times L_2 \to 2^{M_1 \times M_2}$ assigns invariant $I(l) \times I(l')$ to location (l, l'). $e = ((l_1, l_2), \sigma, (l_1', l_2')) \in E$ if one of the following is true:

1. $\sigma \in \Sigma_1 \setminus \Sigma_2$ and $e_1 = (l_1, \sigma, l_1') \in L_1$.
   Then $g_e = g_{e_1} \times M_2$ and $r_e(x^1, x^2) = [r_{e_1}(x^1) \; x^2]^T$, where $x^1 \in M_1$ and $x^2 \in M_2$.
2. $\sigma \in \Sigma_2 \setminus \Sigma_1$ and $e_2 = (l_2, \sigma, l_2') \in L_2$.
   Then $g_e = M_1 \times g_{e_2}$ and $r_e(x^1, x^2) = [x^1 \; r_{e_2}(x^2)]^T$.
3. $\sigma \in \Sigma_1 \cap \Sigma_2$, $e_1 = (l_1, \sigma, l_1') \in L_1$ and $e_2 = (l_2, \sigma, l_2') \in L_2$.
   Then $g_e = g_{e_1} \times g_{e_2}$ and $r_e(x^1, x^2) = [r_{e_1}(x^1) \; r_{e_2}(x^2)]^T$.

Theorem 4.2 (Parallel Composition). *Given A_1 and A_2, suppose bisimulations exist using the stable partitions method on $U_1 \subseteq M_1$ and $U_2 \subseteq M_2$. If for each pair*

$(l_i, l_j) \in L_1 \times L_2$, there exists a one-form of the Pfaffian system at l_i

$$h(dx_1, \dots, dx_n) - dt = 0,$$

and a one-form of the Pfaffian system at l_j

$$h'(dx_{n+1}, \dots, dx_{n+m}) - dt = 0,$$

such that the one-form

$$h(dx_1, \dots, dx_n) - h'(dx_{n+1}, \dots, dx_{n+m}) = d\alpha_{ij},$$

is exact, and α_{ij} is independent of the first integrals of f_{l_i} on U_1 and f_{l_j} on U_2, then, assuming the appropriate compatibility conditions are satisfied, a bisimulation of $A_1 \times A_2$ can be constructed.

Proof. Since the bisimulations of A_1 and A_2 have been constructed with the stable partitions method, we have $n-1$ first integrals for each f_{l_i}, $l_i \in L_1$ and $m-1$ first integrals for each f_{l_j}, $l_j \in L_2$, giving $n+m-2$ first integrals for the vector field $f = [f_{l_i}\ f_{l_j}]^T$. To construct a stable partition on $U_1 \times U_2$ we require $n+m-1$ independent first integrals and the missing one is supplied by α_{ij}. To see that $L_f \alpha_{ij} = 0$, observe that

$$dt = \frac{dx_i}{f_i(x)} = h(dx_1, \dots, dx_n) = h'(dx_{n+1}, \dots, dx_{n+m})$$

where f_i is the ith component of f and $i \in 1, \dots, n+m$. □

5. Applications

In this section we present several applications. We show how to obtain the bisimulation for timed automata and linear systems in Jordan form. Then we look at problems of coordinated autonomous agents, which make a compelling case for the need for a paradigm shift in control design and verification. Some examples are cooperating automated vehicles, aircraft, underwater vehicles, and mobile robots. We show how bisimulations can be constructed for coordinated aircraft and coordinated mobile robots.

5.1. Timed automata.

A timed automaton has dynamics in Pfaffian form given by

$$\begin{aligned} dx_1 - dt &= 0 \\ &\vdots \\ dx_n - dt &= 0. \end{aligned}$$

There are $n-1$ independent tangential foliations defined by the submersions:

$$\begin{aligned} x_1 - x_2 &= c_1 \\ &\vdots \\ x_{n-1} - x_n &= c_{n-1}. \end{aligned}$$

where $c_i \in \mathbb{R}$. Note that the leaves of each foliation have dimension $n - 1$. A transversal foliation is

$$x_n = d_n,$$

though the partition of [1] uses more transversal foliations because of the nature of the enabling and reset conditions:

$$\begin{aligned} x_1 &= d_1 \\ &\vdots \\ x_n &= d_n. \end{aligned}$$

Each of the leaves of the transversal foliations are transverse to every integral curve. Since the dynamics of each location is the same, the stable partition obtained from the foliations is the same, so the enabling conditions and reset conditions are compatible between locations.

5.2. **Mobile robots.** Consider the coordination problem of two mobile robots A and B, operating in a closed workspace of a factory. The robots are modeled using hybrid automata, with each control location corresponding to an atomic maneuver, such as "move forward", or "change direction". Each location of the automaton has the kinematic model of the associated maneuver. We assume in each automaton location, the control inputs are constant, but they are allowed to change instantaneously upon switching locations. The kinematic model for each robot, converted to chained form [20] is the following:

$$\begin{aligned} \dot{x}_1 &= u_1 \\ \dot{x}_2 &= u_2 \\ \dot{x}_3 &= x_2 u_1 \\ \dot{x}_4 &= x_3 u_1. \end{aligned}$$

There are three tangential foliations given by the equations

$$\begin{aligned} x_2 - \frac{u_2}{u_1} x_1 &= c_2 \\ x_3 - \frac{u_1}{2u_2} x_2^2 &= c_3 \\ x_4 + \frac{1}{3}\left(\frac{u_1}{u_2}\right)^2 x_2^3 - \frac{u_1}{u_2} x_2 x_3 &= c_4. \end{aligned}$$

and a transversal foliation given by:

$$x_1 = c_1.$$

To show these foliations define a bisimulation for each robot, we must check the regularity condition:

$$Dh = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{u_2}{u_1} & 1 & 0 & 0 \\ 0 & -\frac{u_1}{u_2}x_2 & 1 & 0 \\ 0 & -\frac{u_1}{u_2}x_3 + \left(\frac{u_1}{u_2}\right)^2 x_2^2 & -\frac{u_1}{u_2}x_2 & 1 \end{bmatrix}$$

This matrix has full rank so long as $u_1 \neq 0$ and $u_2 \neq 0$. Thus, the partition for each robot is defined globally on $\mathbb{R}^4$.

When we take their parallel composition, an extra tangential foliation is introduced:

$$u_{1B}x_{1A} - u_{1A}x_{1B} = c_{AB}.$$

A calculation similar to the next example shows that a bisimulation for the parallel composition exists.

5.3. **Planar aircraft.** Consider the coordination problem of two aircraft A and B flying at a fixed altitude, which was studied in the hybrid systems context in [30]. Each aircraft is modeled by a hybrid system in which an automaton location corresponds to an atomic maneuver performed with constant control inputs. The control inputs are changed instantaneously upon switching control locations. The state g is an element of the special euclidean group $SE(2)$, and X is an element of its algebra $se(2)$. Assuming the aircraft does not exercise it's pitch control, the kinematic dynamics of aircraft A are given by $\dot{g} = gX$ where

$$g = \begin{bmatrix} \cos\phi & -\sin\phi & x \\ \sin\phi & \cos\phi & y \\ 0 & 0 & 1 \end{bmatrix}$$

and

$$X = \begin{bmatrix} 0 & -u_1 & u_2 \\ u_1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

ϕ is the yaw angle, and the inputs u_1, u_2 control the yaw and velocity, respectively. There are two tangential foliations given by equations

$$\begin{aligned} u_1 x - u_2 \sin\phi &= c_x \\ u_1 y + u_2 \cos\phi &= c_y \end{aligned}$$

and a transversal foliation given by

$$\phi = c_\phi.$$

Letting the state variables and inputs of aircraft B be ϕ_B, x_B, y_B, u_{1B}, and u_{2B}, analogous expressions for the tangential and transversal foliations are obtained for aircraft B. An additional tangential foliation is found for the parallel composition of the two systems given by

$$u_{1B}\phi_A - u_{1A}\phi_B = c_{AB}.$$

We check the regularity condition on the five tangential foliations and either of the two transversal foliations. Namely,

$$Dh = \begin{bmatrix} u_{1A} & 0 & -u_{2A}\cos\phi_A & 0 & 0 & 0 \\ 0 & u_{1A} & -u_{2A}\sin\phi_A & 0 & 0 & 0 \\ 0 & 0 & u_{1B} & 0 & 0 & -u_{1A} \\ 0 & 0 & 0 & u_{1B} & 0 & -u_{2B}\cos\phi_B \\ 0 & 0 & 0 & 0 & u_{1B} & -u_{2B}\sin\phi_B \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

This matrix has full rank so long as $u_{1A}, u_{1B} \neq 0$, so the partition is defined globally on $\mathbb{R}^4 \times \mathbb{T}^2$. If, in addition, $\frac{u_{1A}}{u_{1B}}$ is rational, a finite bisimulation on $U \times \mathbb{T}^2$, for compact $U \subset \mathbb{R}^4$, exists.

5.4. **Linear systems in Jordan form.**

In [2] we presented the analytical description of the bisimulation for hybrid automata with linear dynamics in Brunovksy normal form and in diagonal form. These results can be generalized to Jordan form. For each $l \in L$, the procedure is the following: (1) for each type of elementary Jordan block derive expressions for the local first integrals, defining a set of tangential foliations, (2) for each pair of elementary Jordan blocks derive an expression for the coupling first integral, defining another tangential foliation, and finally, (3) derive an expression for the submersion corresponding to a foliation transverse to the linear flow.

We consider the linear system

$$\dot{x} = Ax \tag{5.1}$$

where $A \in \mathbb{R}^{n\times n}$ is of the form $A = diag(J^r \cdots J^r J^c \cdots J^c)$. J^r and J^c are elementary Jordan blocks corresponding to the real (repeated) eigenvalues and complex (repeated) eigenvalues of A, respectively. Following the proposed procedure, we first derive the local first integrals for J^r and J^c.

5.4.1. *Real Eigenvalues.* Consider the elementary Jordan block $J^r \in \mathbb{R}^{m\times m}$ given by

$$J^r = \begin{bmatrix} \lambda & 1 & & \\ & \ddots & \ddots & \\ & & & 1 \\ & & & \lambda \end{bmatrix} \tag{5.2}$$

where $\lambda \in \mathbb{R}$. The solution of $\dot{x} = J^r x$ with initial condition $c \in \mathbb{R}^m$ is

$$x(t) = e^{\lambda t} \begin{bmatrix} 1 & t & \frac{t^2}{2!} & \cdots & \frac{t^{m-1}}{(m-1)!} \\ & 1 & t & \cdots & \\ & & \ddots & \ddots & \vdots \\ & & & 1 & t \\ & & & & 1 \end{bmatrix} c. \tag{5.3}$$

We obtain $m-1$ first integrals $\Psi^r_1, \ldots, \Psi^r_m$ as follows. From the solution of x_m we find

$$e^{\lambda t} = \frac{x_m}{c_m}. \tag{5.4}$$

The solution of x_{m-1} combined with (5.4) gives

$$t = \frac{x_{m-1}}{x_m} - \frac{c_{m-1}}{c_m}. \tag{5.5}$$

Substituting (5.5) in (5.4) we obtain the first integral

$$\Psi^r_{m-1} := x_m \exp\left(-\lambda \frac{x_{m-1}}{x_m}\right) = d_{m-1} \tag{5.6}$$

where $d_{m-1} \in \mathbb{R}$. The remaining $m-2$ first integrals are found by substituting (5.4) and (5.5) in the solutions for x_1 through x_{m-2}. Carrying out this operation recursively, we obtain the first integrals

$$\Psi^r_{m-2} := \frac{x_{m-2}}{x_m} - \frac{x^2_{m-1}}{2x^2_m} = d_{m-2} \tag{5.7}$$

$$\Psi^r_{m-3} := \frac{x_{m-3}}{x_m} - \frac{x_{m-2}x_{m-1}}{x^2_m} - \frac{x^3_{m-1}}{3x^3_m} = d_{m-3} \tag{5.8}$$

$$\vdots$$

$$\Psi^r_{m-k} := \frac{x_{m-k}}{x_m} - \sum_{j=1}^{k-2} \frac{1}{j!} \frac{x^j_{m-1}}{x^j_m} \Psi^r_{m-(k-j)} - \frac{1}{k!} \frac{x^k_{m-1}}{x^k_m} = d_{m-k} \tag{5.9}$$

where $d_j \in \mathbb{R}$. We show these are first integrals by an inductive argument. First, $D\Psi_{m-2} \cdot J^r x = 0$. Suppose $D\Psi^r_{m-j} \cdot J^r x = 0$ for $j = 2, \ldots, k-1$. Then

$$D\Psi_{m-k} \cdot J^r x = \frac{x_{m-k+1}}{x_m} - \frac{x^{k-1}_{m-1}}{(k-1)! x^{k-1}_m} - \sum_{j=1}^{k-2} \frac{x^{j-1}_{m-1}}{(j-1)! x^{j-1}_m} \Psi^r_{m-k+j} = 0.$$

5.4.2. *Complex Eigenvalues.* Consider the elementary Jordan block $J^c \in \mathbb{R}^{m \times m}$ given by

$$J^c = \begin{bmatrix} D & I_2 & & \\ & \ddots & \ddots & \\ & & & I_2 \\ & & & D \end{bmatrix} \tag{5.10}$$

where

$$D = \begin{bmatrix} a & -b \\ b & a \end{bmatrix}; \qquad I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Following [14], the solution of $\dot{x} = J^c x$ is found by converting to the complex domain. Let $z : \mathbb{R} \to \mathbb{C}^{\frac{m}{2}}$, $i \cdot i = -1$, and consider $\dot{z} = Bz$, where

$$B = \begin{bmatrix} \mu & 1 & & \\ & \ddots & \ddots & \\ & & & 1 \\ & & & \mu \end{bmatrix} ; \qquad \mu = a + ib. \tag{5.11}$$

We identify $\mathbb{C}^{\frac{m}{2}}$ with $\mathbb{R}^m$ by the correspondence

$$(z_1, \ldots , z_{\frac{m}{2}} = (x_1 + ix_2, \ldots , x_{m-1} + ix_m).$$

The solution of $\dot{z} = Bz$ is

$$z_k(t) = e^{\mu t} \sum_{j=k}^{\frac{m}{2}} \frac{t^{j-k}}{(j-k)!} c_j.$$

We obtain $m - 1$ first integrals $\Psi_1^c, \ldots , \Psi_m^c$ as follows. First, from the solutions of x_{m-1} and x_m we derive the useful expressions:

$$\begin{aligned} e^{at} &= \left(\frac{x_{m-1}^2 + x_m^2}{c_{m-1}^2 + c_m^2} \right)^{\frac{1}{2}} && (5.12) \\ e^{at} \cos bt &= \frac{c_{m-1}x_{m-1} + c_m x_m}{c_{m-1}^2 + c_m^2} && (5.13) \\ e^{at} \sin bt &= \frac{c_{m-1}x_m - c_m x_{m-1}}{c_{m-1}^2 + c_m^2} && (5.14) \end{aligned}$$

Let

$$\begin{aligned} X_{k+} &= \frac{x_{m-k}x_{m-1} + x_{m-k+1}x_m}{x_{m-1}^2 + x_m^2} \\ X_{k-} &= \frac{x_{m-k}x_m - x_{m-k+1}x_{m-1}}{x_{m-1}^2 + x_m^2}. \end{aligned}$$

Evaluating X_{3+} gives

$$t = \frac{x_{m-3}x_{m-1} + x_{m-2}x_m}{x_{m-1}^2 + x_m^2} - \frac{c_{m-3}c_{m-1} + c_{m-2}c_m}{c_{m-1}^2 + c_m^2}. \tag{5.15}$$

Equipped with (5.12) - (5.15) we can find $m - 1$ first integrals. Considering the last two equations of $\dot{x} = J^c x$ and using polar coordinates, we obtain a first integral

$$\Psi_{m-1}^c := \sqrt{x_m^2 + x_{m-1}^2} \exp\left(-aX_{3+}\right) = d_{m-1} \tag{5.16}$$

where $d_{m-1} \in \mathbb{R}$. The remaining $m - 2$ first integrals are found by evaluating X_{k+} and X_{k-} for $k = 3, 5, 7, \ldots , m-1$ and substituting (5.12) - (5.15) in the solutions for

x_m to x_1. Considering the evaluation of X_{k-} we obtain the first integrals

$$\begin{aligned}\Psi^c_{m-2} &:= X_{3-} = d_{m-2}\\ &\vdots\\ \Psi^c_{m-k+1} &:= X_{k-} - \sum_{j=1}^{\frac{k-3}{2}} \frac{1}{j!} X^j_{3+} \Psi^c_{m-k+1+2j} = d_{m-k+1}.\end{aligned}$$

Considering the evaluation of X_{k+}, we first obtain the first integral

$$\Psi^c_{m-3} := \frac{x^2_{m-3} + x^2_{m-2}}{x^2_{m-1} + x^2_m} - X^2_{3+} = d_{m-3}.$$

The remaining first integrals for $k = 5, 7, \dots$ are

$$\begin{aligned}\Psi^c_{m-5} &:= X_{5+} - \frac{1}{2} X^2_{3+} = d_{m-5}\\ &\vdots\\ \Psi^c_{m-k} &:= X_{k+} - \sum_{j=1}^{\frac{k-5}{2}} \frac{1}{j!} X^j_{3+} \Psi^c_{m-k+2j} - \frac{1}{p!} X^p_{3+} = d_{m-k}\end{aligned}$$

where $p = \frac{k-1}{2}$. We can verify by a recursive argument analogous to the real repeated case that these are first integrals.

5.4.3. *Coupling integrals.* It remains to find the first integrals describing the coupling between elementary Jordan blocks. We consider the pairs (J^r, J^r), (J^r, J^c), and (J^c, J^c).

For the coupling between a J^r and a J^c block, it suffices to find a coupling first integral for the system

$$\dot{x} = \begin{bmatrix} \lambda & 0 & 0 \\ 0 & a & -b \\ 0 & b & a \end{bmatrix} x. \tag{5.17}$$

Using polar coordinates $x_2 = r\cos\theta$, $x_3 = r\sin\theta$, we have $\dot{r} = ar$, from which it is seen that

$$x_1^a (x_2^2 + x_3^2)^{-\frac{\lambda}{2}} = d$$

where $d \in \mathbb{R}$. For the coupling between two J^r blocks it suffices to find a first integral for the system

$$\dot{x} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} x \tag{5.18}$$

which corresponds to the last row of each J^r block. We obtain

$$\lambda_2 x_1 - \lambda_1 x_2 = d.$$

For the coupling between two J^c blocks it suffices to consider the system

$$\dot{x} = \begin{bmatrix} \begin{bmatrix} a_1 & -b_1 \\ b_1 & a_1 \end{bmatrix} & \\ & \begin{bmatrix} a_2 & -b_2 \\ b_2 & a_2 \end{bmatrix} \end{bmatrix} x. \tag{5.19}$$

Converting to polar coordinates, we have $\dot{\theta}_1 = b_1$ and $\dot{\theta}_2 = b_2$, so

$$b_2 \arctan(\frac{x_2}{x_1}) - b_1 \arctan(\frac{x_4}{x_3}) = d.$$

5.4.4. *Transversal foliation.* A expression for the submersion defining the transversal foliation is found by considering a particular instance of the A matrix. Because of the diagonal structure of the Jordan form, an initial candidate is $\Psi_m := x_m = d_m$, but better candidates are often available which are independent of the first integrals over a larger domain.

In two dimensions there is a canonical choice for the transversal foliation given by the first integral of a complementary vector field. Suppose we have $\dot{x} = A_1 x$ with A_1 non-singular and we want to find A_2 such that for all x, $A_1 x$ and $A_2 x$ are not colinear. That is, there does not exist $\lambda \in \mathbb{R}$ such that $\lambda A_1 x = A_2 x$. Equivalently, $A^{-1} A_2$ has no real eigenvalues (it always involves a rotation). We select

$$A_2 = A_1 \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} .$$

A first integral of $\dot{x} = A_2 x$ defines of a transversal foliation of $\dot{x} = A_1 x$.

Finally, in practice it is often advantageous to introduce extra transversal foliations or submanifolds (as in timed automata) in order to achieve compatibility conditions or to keep the equivalence classes from being too large.

5.4.5. *Decidability of hybrid systems with linear dynamics.* Let $\Psi = \{\Psi_i^l\}_{i \in \{1,\dots,n\}, l \in L}$ be the set of submersions obtained in the steps above.

Theorem 5.1. *Let A be a hybrid automaton with linear dynamics in Jordan form and let Ψ be such that for each $l \in L$, $\{\Psi_1^l, \dots, \Psi_n^l\}$ form a set of euclidean coordinates on $I(l)$. If A is compatible with the equivalence relations $\{\simeq_l\}$ defined using Ψ, then the reachability problem for A is decidable.*

6. Implementation

In this section we discuss the implementation of our method. There are two steps: (1) automatic generation of stable partitions, (2) construction of $A_{\simeq}$. The essence of the first step is to automatically generate local first integrals. We rely on the Prelle-Singer procedure [25], which has been implemented in computer algebra packages [17]. Building the automaton $A_{\simeq}$ involves labeling equivalence classes of the stable partitions, checking compatibility conditions, and defining transitions. Both in this approach and the approximative approach of [2], determining the edges of $A_{\simeq}$ corresponding to σ-steps of A can be stated as a problem of existential quantifier elimination. We are in the process of developing an efficient quantifier elimination algorithm which exploits the structure of foliations and this will be reported on separately. (We have not even touched the computation geometric view of bisimulation, in which the bisimulation partition is a *cell decomposition.*)

6.1. **Automatic generation of first integrals.** Prelle and Singer [25] showed that if a differential equation has an elementary first integral (using elementary functions sin, cos, exp, log, arctan, etc.) they must be of a special form. This lead to a semi-decision procedure for finding first integrals. It's extension to vector fields with transcendental terms was described in [17]. We outline the procedure for nth order differential equations following [18].

Consider the differential equation $\dot{x} = f(x)$, $x \in \mathbb{R}^n$. and define the differential operator $D = \sum_{i=1}^{n} f_i \frac{\partial}{\partial x_i}$. The Prelle-Singer procedure involves the following steps.

(1) Set N = 1.
(2) Find all monic, irreducible polynomials g_i with degrees $\leq N$ such that g_i divides Dg_i.
(3) Let $Dg_i = g_i h_i$. Decide if there are constants n_i not all zero such that $\sum_{i=1}^{m} n_i h_i = 0$. If such n_i exist, then $\Pi_{i=1}^{m} g_i^{n_1}$ is a first integral. If no such n_i exist then go to the next step.
(4) Increase N by 1.

The procedure is a semi-decision procedure because an effective bound on N is unknown. Step (2) is the most involved and is discussed in [17].

6.2. **Symbolic model checking.** The size of the automaton $A_{\simeq}$ is exponential in the number of parallel components of the hybrid system and the dimension of the continuous state space. Therefore, rather than *enumerating* all the states of $A_{\simeq}$, the *symbolic* approach explores only the parts of the state space that are relevant and it does so using a symbolic representation of the state space. This approach has reported remarkable results for hardware verification [4]. Symbolic model checking involves computing a fixpoint of a functional on the state space. The symbolic analysis is performed by iterating on a *Pre* (*Post*) operator and uses a set of formulas to represent regions of the state space.

Let $\mathcal{S}$ be a set of formulas in the variables $q \in Q$. A *region* is a set of states $R \subseteq Q$. Let $\sharp R$ denote a set of formulas that define R. We define the operators $Pre : 2^Q \times \Sigma \cup \{\lambda\} \to 2^Q$ and $Post : 2^Q \times \Sigma \cup \{\lambda\} \to 2^Q$ by

$$\begin{aligned} Pre(R,\sigma) &= \{q \in Q \mid \exists q' \in R \,.\, q \xrightarrow{\sigma} q'\} \\ Post(R,\sigma) &= \{q' \in Q \mid \exists q \in R \,.\, q \xrightarrow{\sigma} q'\}. \end{aligned}$$

Following [10], A is *effective* if there is a class of formulas $\mathcal{S}$ which permits the symbolic analysis of A; namely

1. the emptiness problem for each predicate of $\mathcal{S}$ is decidable,
2. $\mathcal{S}$ is closed under boolean operations and *Pre* and *Post* operations,
3. $\sharp Q^f, \sharp Q^0 \in \mathcal{S}$.

Suppose the tangential and transversal foliations on U for each $l \in L$ are defined by submersions $\Psi^l_i(x) = c_i$. Let $\mathcal{S}$ be the class of formulas

$$\Psi^l_i(x)\%c_i$$

with $\% = \{\leq, <, =, >, \geq\}$, $l \in L$, $i = 1, \ldots, n$, and all finite conjunctions and disjunctions of these expressions.

Theorem *A with $\mathcal{S}$ is effective.*

Proof. We observe that: (1) the regions Q^0, Q^f can be represented as predicates of $\mathcal{S}$ by the compatibility assumption, (2) $\sharp Pre(R,\sigma), \sharp Post(R,\sigma) \in \mathcal{S}$ for $\sharp R \in \mathcal{S}$, by the compatibility of g_e and r_e and the stable partitions construction, (3) the emptiness problem for $\mathcal{S}$ is decidable. Indeed, consider a predicate defining a closed region: $\exists x.(c_1 \leq \Psi_1(x) \leq d_1) \wedge \cdots \wedge (c_n \leq \Psi_n(x) \leq d_n)$. This predicate is equivalent to the quantifier free expression $(c_1 \leq d_1) \wedge \cdots \wedge (c_n \leq d_n)$. □

7. Conclusion

This paper contributes a new methodology for model checking of hybrid systems under natural compatibility conditions of the enabling and reset conditions, which represents a breakthrough from current capabilities that have been limited to timed and linear rate automata. Implied by our results are decidability results for the examples considered: coordinated aircraft, coordinated robots, and linear hybrid systems.

Model checking may provide a vast improvement in efficiency over simulation-based approaches for validating hybrid system performance. Nevertheless, there is some way to go before the theory can be turned to practical benefit. Algorithm development and attention to applications in embedded systems design and coordinated autonomous agents where this method applies are obvious next steps in our research.

The paper suggests some areas for future theoretical investigations. First, the paper develops a local geometric theory of bisimulation. A global theory is needed. "Inspired adhoccery" suggests patching together local partitions similar to the gluing of coordinate neighborhoods in a manifold using diffeomorphisms. Another approach is to use Lie group symmetries. An exciting new horizon is to present a unified theory of hybrid systems with symmetries by treating at once the group symmetries at the automaton level [8, Ch. 14] and the group symmetries of the vector fields. We plan to report on these directions in future papers.

Acknowledgements: The author is grateful to Marika di Benedetto and Alberto Sangiovanni-Vincentelli for their feedback and guidance.

References

[1] R. Alur, D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, no. 126, pp. 183-235, 1994.

[2] M. Broucke. A geometric approach to bisimulation and verification of hybrid systems. In *Hybrid Systems: Computation and Control*, LNCS 1569, p. 61-75, Springer-Verlag, 1999.

[3] R. Bryant, S. Chern, R. Gardner, H. Goldschmidt, P. Griffiths. *Exterior Differential Systems.* Springer-Verlag, New York, 1991.

[4] J. Burch, E. Clarke, K. McMillan, D. Dill, Hwang. Symbolic Model Checking: 10^{20} states and beyond. *Information and Computation*, 98(2), p. 142-70, 1992.

[5] P. Caines and Y. Wei. On dynamically consistent hybrid systems. *Hybrid Systems II*, pp. 86-105, Springer-Verlag, 1995.

[6] L.O. Chua, M. Komuro, and T. Matsumoto. The double scroll family - part I: rigorous proof of chaos. *IEEE Transactions on Circuits and systems* vol. 33, no. 11, pp. 1072-1097, November, 1986.

[7] E.M. Clarke, E.A. Emerson, and A.P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, vol. 8, no. 2, pp. 244-263, 1986.

[8] E.M. Clarke, O. Grumberg, D.A. Peled. *Model Checking*. Princeton University Press, 1999.

[9] J.M. Davoren. *Modal Logics for Continuous Dynamics*. PhD. Dissertation, Cornell University, 1998.

[10] T.A. Henzinger. Hybrid automata with finite bisimulations. In *"Proc. 22nd ICALP: Automata, Languages and Programming*, LNCS 944, pp. 324-335, Springer-Verlag, 1995.

[11] R. Alur and T.A. Henzinger. *Computer-Aided Verification. An Introduction to Model Building and Model Checking for Concurrent Systems*. Draft, 1998.

[12] T. Henzinger. The theory of hybrid automata. In *Proc. 11th IEEE Symposium on Logic in Computer Science*, pp. 278-292, New Brunswick, NJ, 1996.

[13] T. Henzinger and H. Wong-Toi. Linear phase-portrait approximations for nonlinear hybrid systems. *Hybrid Systems III*, LNCS 1066, pp. 377-388, Springer-Verlag, 1996.

[14] M. Hirsch, S. Smale. *Differential equations, dynamical systems, and linear algebra*. Academic Press, 1974.

[15] V. Jurdjevic. *Geometric Control Theory*. Cambridge Studies in Advanced Mathematics; 52. Cambridge University Press, 1997.

[16] H. B. Lawson. The Quantitative theory of foliations. *Regional Conference Series in Mathematics*, no. 27. American Mathematical Society, Providence, 1977.

[17] Y.-K. Man. Computing closed-form solutions of first-order ODE's using the Prelle-Singer procedure. *Journal of Symbolic Computation*, no. 16, p. 423-443, 1993.

[18] Y.-K. Man. First integrals of autonomous systems of differential equations and the Prelle-Singer procedure. *Journal of Physics A: Mathematical and general*, no. 27, pp. 329-332, 1994.

[19] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[20] R. Murray and S. Sastry. Nonholonomic motion planning: steering using sinusoids. *IEEE Transactions on Automatic Control*, vol.38, no.5, pp. 700-16, May, 1993.

[21] R. Paige and R.E. Tarjan. Three partition refinement algorithms. *SIAM Journal on Computing*, vol.16, no.6, pp. 973-89, December 1987.

[22] G. Lafferriere, G. Pappas, S. Yovine. A new class of decidable hybrid systems. *Hybrid Systems: Computation and Control*, LNCS 1569, p. 137-151, Springer-Verlag, 1999.

[23] D.M.R. Park. Concurrency and automata on infinite sequences. *Fifth GI Conference on Theoretical Computer Science*, pp. 167-183, Springer, 1981.

[24] A. Pnueli. The temporal logic of programs. *Proc. of the 18th Annual Symposium on Foundations of Computer Science*, pp. 46-57, IEEE COmputer Society Press, 1977.

[25] M.J. Prelle and M.F. Singer. Elementary first integrals of differential equations. *Transactions of the American Mathematical Society*, vol. 279, no. 1, September 1983, pp. 215-229.

[26] J. Queilli and J. Sifakis. Specification and verification of concurrent systems in CESAR. *Fifth International Symposium on Programming*, LNCS 137, pp. 337-351, Springer-Verlag, 1981.

[27] W. Sluis. *Absolute Equivalence and its Applications to Control Theory*. Ph.D. thesis, University of Waterloo, 1992.

[28] J. Stiver, P. Antsaklis, M. Lemmon. A logical DES approach to the design of hybrid control systems. *Mathematical and computer modelling*. vol. 23, no. 11-1, pp. 55-76, June, 1996.

[29] J.C. McKinsey and A. Tarski. The algebra of topology. *Annals of Mathematics*, 45, pp. 141-191, 1944.

[30] C. Tomlin, G. Pappas, J. Lygeros, D. Godbole, and S. Sastry. Hybrid control models of next generation air traffic management. *Hybrid Systems IV*, LNCS 1273, pp. 378-404, Springer-Verlag, 1997.

[31] M.Y. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification. *Proceedings of the First Annual Symposium on Logic in Computer Science*, pp. 322-331, IEEE Computer Society Press, 1986.

[32] F. Warner. *Foundations of Differential Manifolds and Lie Groups*. Springer-Verlag, New York, 1983.

Department of Electrical Engineering and Computer Sciences, 211 Cory Hall, University of California, Berkeley, CA 94720

E-mail address: mire@eecs.berkeley.edu