A Decompositional Proof Scheme for Automated Convergence Proofs of Stochastic Hybrid Systems*

Jens Oehlerking and Oliver Theel

Department of Computer Science
University of Oldenburg
26111 Oldenburg, Germany
{jens.oehlerking|oliver.theel}@informatik.uni-oldenburg.de

Abstract. In this paper, we describe a decompositional approach to convergence proofs for stochastic hybrid systems given as probabilistic hybrid automata. We focus on a concept called "stability in probability," which implies convergence of almost all trajectories of the stochastic hybrid system to a designated equilibrium point. By adapting classical Lyapunov function results to the stochastic hybrid case, we show how automatic stability proofs for such systems can be obtained with the help of numerical tools. To ease the load on the numerical solvers and to permit incremental construction of stable systems, we then propose an automatable Lyapunov-based decompositional framework for stochastic stability proofs. This framework allows conducting sub-proofs separately for different parts of the automaton, such that they still yield a proof for the entire system. Finally, we give an outline on how these decomposition results can be applied to conduct quantitative probabilistic convergence analysis, i.e., determining convergence probabilities below 1.

1 Introduction

During the previous decade, there has been significant progress in the field of automated stability proofs for feedback control systems. Most importantly, methods for the automatic computation of Lyapunov functions, serving as certificates of the stability property, have been developed [1–4]. A Lyapunov function can be seen as a type of generalized "energy function," ensuring that a system always makes some sort of progress while converging toward a desired equilibrium state. Central tools in this context are *semidefinite programming (SDP)* solvers, which numerically solve the constraint systems arising in Lyapunov function computation. These methods are, in theory, applicable to purely discrete-time (given as difference equations/inclusions or automata), purely continuous-time (given as differential equations/inclusions), and hybrid systems (given as a combination

* This work was partly supported by the German Research Foundation (DFG) as part of the Transregional Research Center "Automatic Verification and Analysis of Complex Systems" (SFB/TR 14/2 AVACS), www.avacs.org.

of both). In the hybrid domain, the presence of both complex discrete structures and continuous dynamics given by differential equations makes the problem hard to solve in practice. In particular, the presence of complex discrete structures (i.e., large and irregular automata) often leads to problems with the numerical solvers: badly conditioned problems, rounding errors, and inaccuracies caused by the optimization algorithms themselves. Furthermore, it is difficult to design a stable hybrid system with complex discrete behavior, as existing analysis methods are only applicable to a complete model of the system. Consequently, they are of only limited help during the design process. Therefore, it is useful to decompose the problem of identifying a suitable Lyapunov function into SDP problems that are as small as possible, while still being able to conduct a convergence proof of the entire system. Apart from a decreased load on the numerical solver, arguments on parts of the hybrid system can be useful for successfully designing a system with the desired convergence property.

This paper extends the classes of hybrid systems that can be dealt with efficiently, by proposing such a decompositional approach for *probabilistic hybrid automata*, i.e., systems that contain probabilistic Markovian transitions between the discrete modes. We presented an automatable decompositional framework for systems without stochastic behavior in [5]. Here, we mainly focus on qualitative ("does a system converge with probability 1?") stability analysis, but also discuss the applicability of the results to quantitative ("with which probability is the system guaranteed to converge?") stability analysis. The result is an automatable decompositional framework allowing for the SDP-based computation of Lyapunov functions for the probabilistic case. As it turns out, probabilistic hybrid automata actually sometimes permit a stronger decomposition than their non-probabilistic counterparts. Furthermore, these results are also, to a certain extent, applicable to systems with *stochastic differential equations* instead of plain differential equations defining the continuous dynamics.

The paper is structured as follows: In Section 2, we formally define a probabilistic hybrid system model and probabilistic stability properties. Section 3 then states non-decompositional Lyapunov theorems that can be used to prove convergence, and details the computational procedure for computing such functions. In Section 4, we give decompositional techniques based on the computational method, yielding several theorems that allow the construction of a convergence proof from separate local computations. Section 5 shows the decomposition of a convergence proof for an example automaton. In Section 6, we discuss the exptension of these results for quantitative convergence analysis, i.e., for deriving stabilization properties that lie below probability 1. We then conclude in Section 7 with a discussion of the implications of our results.

2 Probabilistic Hybrid Systems

The system model we use in this paper is given next. It consists of standard hybrid automata, augmented with discrete probabilistic experiments tied to the discrete transitions. Whenever a transition is taken, we give a probability distri-

bution over possible discrete successor states. Furthermore, we allow for differential inclusions in the mode dynamics instead of just differential equations.

Definition 1 (Probabilistic Hybrid Automaton). *Define $\mathcal{D}_n$ as the set of all n-dimensional, nonempty, convex, closed, and upper semicontinuous differential inclusions $\dot{x} \in F(x)$ on some state space $\mathcal{S}$. Here, $F(x)$ is a set-valued function mapping each x onto a set of possible values for the vector field direction $\dot{x}$.*

A hybrid automaton *H is a tuple $(\mathcal{M}, \mathcal{S}, \mathcal{T}, Flow, Inv, Init)$, where*

- *$\mathcal{M}$ is a finite set of* modes
- *$\mathcal{S} = \mathbb{R}^n$ is the* continuous state space
- *$\mathcal{T}$ is a set of* mode transitions *given as tuples $(m, Target, G, Update)$, where*
  - *$m \in \mathcal{M}$ is the* source mode
  - *$Target : \mathcal{M} \to [0, 1], \sum_{m \in \mathcal{M}} Target(m) = 1$ is the* target mode mapping
  - *$G \subseteq \mathcal{S}$ is the* guard set
  - *$Update : \mathcal{S} \to \mathcal{P}(\mathcal{S})$ is the* update function *for the continuous state*
- *$Flow : \mathcal{M} \to \mathcal{D}_n$ is the* flow function*, mapping each mode onto a continuous evolution given as differential inclusion*
- *$Inv : \mathcal{M} \to \mathcal{P}(\mathcal{S})$ is the* invariant function*, mapping each mode onto a closed subset of the continuous state space*
- *$Init \subseteq \mathcal{M} \times \mathcal{S}$ is the set of combinations of initial discrete and continuous states*

Hybrid automata are finite automata where each node corresponds to a mode of the hybrid system and is labelled with a corresponding differential inclusion via the Flow function (see Figure 1). The continuous state must evolve according to this differential inclusion whenever the system is in this mode. Also, via Inv, each mode has an associated invariant set. A system may only stay in mode m if the current discrete state $x(t) \in \mathcal{S}$ is in $\mathrm{Inv}(m)$. Note that systems with only differential equations in the modes, instead of differential inclusions, are obtained by letting the $F(x)$ be singular sets.

Discrete transitions are driven by the following semantics: whenever the continuous state $x(t)$ reaches a guard set G of a mode transition with current mode m as source mode, the transition *can* be taken. As soon as the invariant set $\mathrm{Inv}(m)$ of the current mode m is left, m must be left immediately, i.e., some applicable transition *must* be taken. If this is not possible, then we will not consider the solution segment with respect to stability. These semantics permit non-deterministic switching in the sense that transitions can be specified to occur somewhere in a certain range of states. Whenever a transition is taken, the function Update is applied to the continuous state, and a new mode is chosen, based on the probability distribution given by Target.

Definition 2 (Trajectory). *A* trajectory *is a solution $x(t)$ for the hybrid automaton, considering only the evolution of the continuous state $x \in \mathcal{S}$. We only consider infinite, non-zeno solutions, i.e., $x(t)$ must be defined for all $t > 0$.* [1] *If*

[1] Finite solution segments can for instance occur if some invariant set is left, but no outgoing transition can be taken.

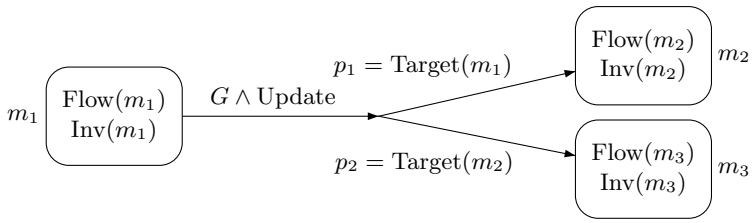


Fig. 1. Graphical representation of probabilistic hybrid automata

discrete updates of the continuous variables occur via Update functions at time t, then we consider $x(t)$ as the state after all such updates. Similarly, $m(t)$ is the discrete mode at time t. The (finite or infinite) mode sequence (m_i) *lists all modes visited by a trajectory, in order.*

The stability notion used in this paper is *global asymptotic stability in probability*. Informally, this term implies that each trajectory converges to an equilibrium point with probability 1, for all resolutions of possible non-determinism. Furthermore, a property similar to Lyapunov stability for the stochastic case is implied. Stronger stability definitions (e.g., almost sure stability), exist in the literature, but we chose global asymptotic stability in probability for two reasons: 1) it already implies convergence with probability 1, and 2) it blends in nicely with Lyapunov theory and allows the direct use of automatic Lyapunov function computation methods.

Definition 3 (Global Asymptotic Stability in Probability). *A probabilistic hybrid automaton H is* globally stable in probability *wrt. an equilibrium state x_e if*

$$\forall \epsilon, \epsilon' > 0 \, \exists \delta > 0 : ||x(0) - x_e|| < \delta \Rightarrow P(\exists t : ||x(t) - x_e|| > \epsilon') < \epsilon$$

and globally attractive in probability (GA-P) *if*

$$P(\lim_{t \to \infty} x(t) = x_e) = 1,$$

where 0 is the origin of $\mathbb{R}^n$. A system that is both globally stable in probability and globally attractive in probability is called globally asymptotically stable in probability (GAS-P).

3 Lyapunov Functions for Probabilistic Systems

Lyapunov functions are a central tool for proving stability properties for various kinds of dynamic systems. In the following, we present a theorem implying that the existence of a Lyapunov function whose value is *expected to decrease* at all time instants is sufficient for proving GAS-P for probabilistic hybrid automata.

Whenever a discrete transition is taken, it is permissible that the Lyapunov function increases, as long as this is not the expected behavior in the long run. Note that Lyapunov function computation, without loss of generality, assumes that the equilibrium of the system lies at the origin of the continuous state space. If one wants to show GAS-P wrt. some other equilibrium state, the system can be "shifted" accordingly.

Definition 4 (Definiteness). *A function $f : S \to \mathbb{R}, S \subseteq \mathbb{R}^n$ is called* positive semidefinite*, if for all $x \in S : f(x) \geq 0$, and* positive definite*, if it is positive semidefinite and $f(x) = 0 \Leftrightarrow x = 0$. A function f is called negative (semi)definite, if $-f$ is positive (semi)definite. A function $f : \mathbb{R}^n \to \mathbb{R}$ is called positive (semi)definite on a set $S \subset \mathbb{R}^n$, if the restricted function $f_{|S}$ is positive (semi)definite. This definition extends to negative (semi)definiteness accordingly.*

Theorem 1 (Discontinuous Lyapunov Functions for Probabilistic Hybrid Systems). *Let H be a probabilistic hybrid automaton. If for each $m \in \mathcal{M}$ there exists a continuously differentiable function $V_m : \mathcal{S} \to \mathbb{R}$ such that*

(1) *$V_m(x) - \alpha||x||$ is positive definite on $Inv(m)$ for some $\alpha > 0$,*
(2) *$\dot{V}_m(x) := \sup\left\{\frac{dV_m}{dx} f(x) \,\middle|\, f(x) \in F_m(x)\right\}$ is negative definite on $Inv(m)$, where F_m is the right hand side of the differential inclusion $Flow(m)$,*
(3) *for each mode transition $(m_1, Target, G, Update) \in \mathcal{T}$: $x \in G \Rightarrow V_{m_1}(x) - \sum_{m \in \mathcal{M}} Target(m) \cdot V_m(Update(x)) \geq 0$,*
(4) *for all $m : V_m(x) \to \infty$ as $||x|| \to \infty$,*

then H is GAS-P wrt. the equilibrium 0. The function V_m is called the local Lyapunov function (LLF) *of H for mode m. The family of the $V_m, m \in \mathcal{M}$ is called the* global (discontinuous) Lyapunov function (GLF) *of H.*

Proof. Let $x(t)$ be a trajectory of H with switching times t_i and associated mode sequence (m_i). For ease of notation define $W(t) := V(x(t))$.
Attractivity: *Per definition of the semantics for probabilistic hybrid automata and condition* (2)*, for all $t_{i+1} \geq t \geq t_i$ the following holds:*

$$W(t) = W(t_i) + \int_{t_i}^{t} \dot{W}(\tau) d\tau \leq W(t_i).$$

This, together with condition (3)*, implies that, for all $t \geq 0$,*

$$E(W(t)) = W(0) + E\left(\sum_i \int_{t_i}^{t_{i+1}} \dot{W}(\tau) d\tau\right) + E\left(\sum_i \Delta_i\right) \leq W(0),$$

where $\Delta_i = V_{m_i}(x(t_i)) - V_{m_{i+1}}(Update(x(t_i)))$. Therefore, for all $t \geq s \geq 0$, we obtain

$$E(W(t) \,|\, \{W(\tau) \,|\, \tau \leq s\}) \leq W(s),$$

i.e., $W(t)$ is a supermartingale[2] *[6, p. 474]. Furthermore,*

$$0 \leq E(W(t)) = W(0) + E\left(\sum_i \int_{t_i}^{t_{i+1}} \dot{W}(\tau)d\tau\right) + E\left(\sum_i \Delta_i\right) \leq W(0) < \infty.$$

Therefore, $E(|W(t)|) = E(W(t)) < \infty$ for all $t \geq 0$ and Doob's martingale convergence theorem [6, p. 505] can be applied, giving us

$$P(\exists x_0 : \lim_{t\to\infty} W(t) = x_0) = 1.$$

Condition (2) *implies that $x_0 = 0$, therefore,*

$$P(lim_{t\to\infty} W(t) = 0) = 1,$$

and per conditions (1) *and* (4)*,*

$$P(lim_{t\to\infty} x(t) = 0) = 1.$$

Stability: *Since $W(t)$ is a supermartingale, the following inequality holds for all $\tilde{\epsilon} > 0$:*

$$P(\exists t : W(t) \geq \tilde{\epsilon}) \leq W(0)/\tilde{\epsilon}.$$

Let $\epsilon, \epsilon' > 0$. Choose $0 < \tilde{\epsilon} < \min\{\epsilon', 1/\epsilon\}$. Then

$$P(\exists t : W(t) > \epsilon') < \epsilon \cdot W(0).$$

Set $\delta := \inf\{||x|| \,|\, V_m(x) \geq 1, m \in \mathcal{M}\} > 0$, then $||x(0)|| < \delta$ implies that

$$P(\exists t : V(x(t)) > \epsilon') < \epsilon,$$

and therefore, per condition (1)

$$P(\exists t : ||x(t)|| > \epsilon') < \epsilon.$$

This theorem can also be adapted to the case where the differential inclusions per mode are replaced by stochastic differential equations $\dot{x} = f(x, \sigma)$. In this case, condition (2) can be replaced by

(2') $E\left(\frac{dV_m}{dx} f(x,\sigma)\right) = \frac{dV_m}{dx} E(f(x,\sigma))$ *is negative definite on Inv(m)*

For non-hybrid systems, a proof outline for this case based on results by Kushner [7] can be found in [8]. For the hybrid case, it can be combined with the proof for Theorem 1 to accommodate for stochastic differential equations in hybrid systems.

For linear dynamics and quadratic Lyapunov function candidates of the form $V(x) = x^T P x, P \in \mathbb{R}^n \times \mathbb{R}^n$, the conditions (1) to (4) can directly be mapped onto a *linear matrix inequality (LMI)* problem [9], which in turn can be solved

[2] *Supermartingales* are stochastic processes for which, given an evolution to time s, the expected value at time $t \geq s$ is never higher than the value at time s.

automatically with nonlinear optimization techniques [10]. The solution of such an LMI problem consists of valuations of the entries of P and some auxiliary variables μ_m^i, ν_m^i, and η_e^i that are used to express the invariants and guards through the so-called $\mathcal{S}$-procedure [10]. Invariants are encoded through an arbitrary number of matrices Q_m^i per mode m that satisfy $x \in Inv(m) \Rightarrow x^T Q_m^i x \geq 0$. The same applies to the guard sets G and the matrices R_e^i. The $\mathcal{S}$-procedure always results in correct over-approximations of invariant and guard sets, although it can be conservative [2]. Denote "$x^T P x$ is positive semidefinite" as "$P \succeq 0$". Assume the dynamics for each mode m are given as the conic hull of a family of linear dynamics, i.e., $\dot{x} \in \text{cone}\{A_{m,1}x, \ldots, A_{m,k}x\}$. Then, the associated LMI problem looks as follows.

Theorem 2 (LMI Formulation). *If the following LMI problem has a solution, then the system is GAS-P wrt.* 0*:*

Find $P_m \in \mathbb{R}^n \times \mathbb{R}^n, \alpha > 0, \mu_m^i \geq 0, \nu_m^i \geq 0, \eta_e^i \geq 0$*, such that*

$$\textit{for each mode } m\textit{:}\ P_m - \sum_i \mu_m^i Q_m^i - \alpha I \succeq 0 \tag{1}$$

$$\textit{for each mode } m \textit{ and each } j\textit{:}\ -A_{m,j}^T P - P A_{m,j} - \sum_i \nu_m^i Q_m^i - \alpha I \succeq 0 \tag{2}$$

for each transition e *and each target mode* m' *with* $Target(m') > 0$*:*

$$P_m - \sum_{m'} Target(m') \cdot P_{m'} - \sum_i \eta_e^i R_e^i \succeq 0 \tag{3}$$

Conditions (1) to (3) directly map onto the same conditions of Theorem 1. Condition (4) of Theorem 1 is already satisfied through the use of quadratic function templates. If the system dynamics are not linear or a non-quadratic Lyapunov function candidate is needed, then the *sums-of-squares decomposition* [3] can be applied to transform the constraints into an LMI problem.

LMI problems are a representation of semidefinite programming (SDP) problems, which in turn form a special class of convex optimization problems [10] that can be solved with dedicated software, e.g., CSDP [11] or SeDuMi [12]. The result of the computation – if it is successful – yields valuations of the matrix variables P_m, and thereby a suitable discontinuous Lyapunov function, completing the stability proof. If no positive result is obtained, then no conclusion about the stability or instability of the system can be drawn, and it is not easy to identify the cause of the problem. It is possible that the system is indeed unstable, or that a different Lyapunov function parameterization (for instance applying the sums-of-squares decomposition [3]) or a different hybrid automaton representation of the system might allow for a solution to the LMI problem. Moreover, the computation might simply fail for numerical reasons, despite the existence of a Lyapunov function. These problems are more likely to occur, the larger the LMI problem grows, i.e., the more complex the hybrid automaton is.

For this reason, we next turn to decompositional proofs of GAS-P, keeping the LMI problems comparatively small, and in case of failure, giving constructive information about the part of the hybrid automaton that is most likely

responsible for the failure. Furthermore, decomposition can also be turned into composition, in the sense that a stable hybrid automaton can be designed step by step, by solving LMI problems for the different sub-automata. These sub-automata can then be composed according to the results in the next section, yielding a new stable automaton.

4 Decompositional Computation of Lyapunov Functions

This section deals with automaton-based decomposition of stability proofs, as opposed to techniques like the composition of input-to-state stable systems [4], which work on the continuous state space. The hybrid automaton is divided into sub-automata, for which LMI problems can either be solved completely independently, or sequentially, but with some information being passed from one computation to the next. In contrast to the non-stochastic setting covered in [5], it turned out that stochastic stability proofs allow for stronger decompositional results, which exploit the knowledge of transition properties. The different levels of decomposition are outlined in the following.

The decompositions take place on the graph structure defined by the automaton. Hence, we will apply graph-theoretic terms to the automaton by viewing the automaton as a hypergraph with some labels. The modes in the hybrid automaton are therefore sometimes referred to as *nodes*, and the transitions as *hyperedges*. Since one hybrid system can be represented by different hybrid automata with potentially different graph structures, some representations of the system might be more amenable to decomposition than others.

The first level of decomposition concerns the *strongly connected components* of the hybrid automaton.

Definition 5 (Strongly Connected Components). *A* strongly connected component (SCC) *of a hypergraph is a maximal subgraph G, such that each node in G is reachable from each other node.*

Note that, here, reachability is exclusively based on the graph structure: continuous dynamics, invariants, and guards are not taken into account. It is well known that every node of a hypergraph belongs to exactly one SCC, and that the reachability relation between the SCCs of a hypergraph forms an acyclic graph structure. This property can be exploited to allow for *completely independent* Lyapunov function computation for the SCCs of a hybrid automaton.

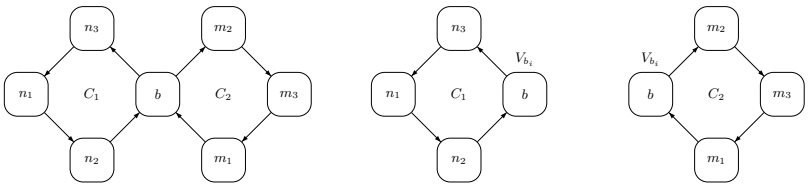
Theorem 3 (Decomposition into Strongly Connected Components). *Let H be a probabilistic hybrid automaton. If all sub-automata pertaining to the SCC of H are GAS-P wrt.* 0*, then so is H.*

The consequence of this theorem is that LMI problems as per Theorem 2 can be solved locally for each SCC, still yielding a proof of GAS-P for the entire system. The proof is a variant of the proof for non-probabilistic hybrid automata stated in [5]. The following corollary is a consequence of this property.

Corollary 1 (Multiple Equilibria). *If all SCCs of probabilistic hybrid automaton H are GA-P, but with respect to different equilibrium states, then each trajectory of H will converge to one of these states with probability* 1.

The second level of decomposition concerns cycles within an SCC and is also described in detail for non-probabilistic systems in [5]. Since it is based on the decompositional properties of Lyapunov functions, and not on the system itself, the result applies to both non-probabilistic and probabilistic hybrid automata. Therefore, we only give a brief summary of the decomposition technique.

These results are based on a theorem that allows the decomposition of LMI computations within an SCC. Consider an automaton consisting of two subgraphs C_1 and C_2, which overlap in exactly one node b (see Figure 2(a)). Again, LMI computations can be conducted separately for C_1 and C_2. However, the computations are not completely independent, but a conic predicate given by a family of local Lyapunov functions V_{b_i} for b is used to "connect" the two Lyapunov function computations for C_1 and C_2. First, a local LMI problem is solved for C_1, computing the V_{b_i} (see Figure 2(b)). These V_{b_i} have the property that, whenever the LLF for node b in a GLF for C_2 is a conic combination of the V_{b_i}, there exists a GLF for the entire system comprising both subgraphs. Therefore, this requirement on the LLF of b is added as an additional constraint to the LMI for C_2 (see Figure 2(c)). If there is a solution to this second local LMI problem, then the system is GAS-P. The probabilistic version of this theorem is given next. Again, a straightforward modification of the proof from [5] yields a variant for the probabilistic case.



(a) Two cycles

(b) Computation of the V_{b_i} for C_1

(c) Computation for C_2, using the V_{b_i}

Fig. 2. Decomposition with separate computations for subgraphs C_1 and C_2

Theorem 4 (Decomposition inside an SCC). *Let H be a probabilistic hybrid automaton consisting of two subgraphs C_1 and C_2 with a single common node b. Let $b, n_1, \ldots, n_j$ be the nodes of C_1 and $b, m_1, \ldots, m_k$ be the nodes of C_2. Let each $V_{b_1}, \ldots, V_{b_m}$ be a LLF for b belonging to a GLF $V_{b_i}, V^i_{n_1}, \ldots, V^i_{n_j}$ for the entire subgraph C_1. If there exists a GLF $V_b, V_{m_1}, \ldots, V_{m_k}$ for subgraph C_2 with $\exists \lambda_1, \ldots, \lambda_m > 0 : V_b = \sum_i \lambda_i V_{b_i}$, then H is GAS-P wrt.* 0.

We employ this theorem for cycle-based decomposition of the stability proof within an SCC. In a slight abuse of terminology, we will use the term "cycle" for the graphs generated by a cyclic path, as defined below.

Definition 6 (Cycle). *A* cycle *of a hypergraph G is a subgraph G', such that there exists a closed path in G', covering all edges and nodes of G'. A cycle C is* simple*, if there exists such a path that only traverses each node once.*

Since each node inside an SCC is part of at least one simple cycle, this theorem allows for local LMI computations on a per-cycle basis. While solving the LMI problem for a cycle C, one can already compute adequate V_{b_i}, which are then taken into account for other cycles that intersect with C. Note, that this decomposition is in general conservative, i.e., some Lyapunov functions are lost in the computation of the V_{b_i}. It is, however, possible to approximate the real set of existing Lyapunov functions of the chosen parameterization arbitrarily close by increasing the number of computed V_{b_i}.

The previous two results stem from analysis of stability properties for non-probabilistic systems. We will now show another decompositional property that is only applicable to probabilistic hybrid automata. However, this decomposition will only preserve global attractivity in probability (GA-P), as opposed to global asymptotic stability in probability (GAS-P). Since attractivity (that is, convergence to the equilibrium) is usually the more interesting property, this is still a useful result.

If the automaton has certain local graph structures, then Lyapunov functions can sometimes be computed *completely independently per mode*, while still allowing for a proof of global attractivity in probability.

We first define *finiteness in probability*, a property of individual cycles. For a simple cycle, this property implies that, with probability 1, it is only possible for a trajectory to traverse the cycle finitely long until the cycle is either not entered again or the trajectory ends up in a mode of the cycle which is not left any more.

Definition 7 (Finiteness in Probability). *Let C be a cycle in a probabilistic hybrid automaton H. If at least one edge in C belongs to a hyperedge e, such that there exists a mode m with $Target_e(m) > 0$ belonging to a different SCC, then C is called* finite in probability.

For an illustration of a cycle that is finite in probability, see Fig. 3. The consequence of this property for stability verification is as follows. A simple cycle that is finite in probability does not need to be mapped onto one LMI problem for the entire cycle. Instead, it is sufficient to provide a local Lyapunov function for each mode of the cycle, with *no constraints spanning several modes.* The existence of such local Lyapunov functions ensures that the system will always stabilize, in case a trajectory "gets stuck in a mode". If it does not get stuck, then finiteness in probability ensures that the cycle is eventually left with probability 1. The following lemma breaks GA-P down into a probability on the cycles of the system and will be used to prove the decomposition theorem.

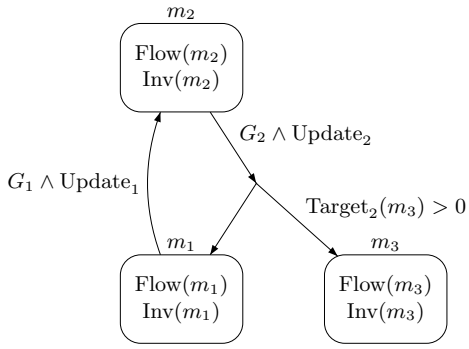


Fig. 3. Finite in probability cycle consisting of m_1 and m_2

Lemma 1 (GA-P and Finite in Probability Cycles). *Let H be a probabilistic hybrid automaton. If for each cycle C of H one of the following two conditions holds, then H is GA-P wrt. 0:*

(1) C is finite in probability and for each $m \in C$ there exists a LLF V_m, or
(2) there exists a GLF for C

Proof. Case 1, *(m_i) is finite: Let m_n be the final element of (m_i). There either exists a LLF V_{m_n} for m_n per condition (1), or per condition (2) as part of a GLF. Therefore, $x(t) \to 0$.*
Case 2, *(m_i) is infinite: Since (m_i) only contains modes belonging to a finite number of SCC $C_1, \ldots, C_m$, we only need to consider the final SCC C_m. If there exist edges originating within C_m, with target states outside of C_m having a probability greater than zero, then the following holds: The probability mass of all trajectories that end up in C_m and take such transitions infinitely often is 0. Summing up over all SCCs, with probability 1, there exists a point in time, after which the system will remain in a subgraph not containing such edges. This subgraph can be covered by a single cycle that is not finite in probability. Per condition (2), there exists a GLF for this cycle, which implies that $P(x(i) \to 0) = 1$.*

By itself, this result is of limited use, because all cycles of the hybrid automaton need to be considered. However, it is used to prove the following theorem, which allows a further decomposition. It implies that nodes lying only on finite in probability *simple* cycles can be treated *completely separately* within an SCC.

Theorem 5 (Decomposition of Lyapunov Functions within an SCC). *Let C be an SCC of a probabilistic hybrid automaton. Let N be the set of nodes that lie only on finite in probability simple cycles. Let C' be the hybrid automaton obtained by removing the nodes of N and all incident transitions from C. If the following two conditions both hold, then C is GA-P wrt. 0:*

(1) for each $m \in N$ there exists a LLF V_m, and

(2) there exists a GLF for C'

Proof. We show that the prerequisites of Lemma 1 are always satisfied. Let D be a cycle in C as required by Lemma 1. There are three cases:
Case 1*, D is simple and finite in probability: there exists a LLF for each node of D, either per condition* (1) *or as part of a GLF per condition* (2)*. Therefore, condition* (1) *of Lemma 1 is satisfied for D.*
Case 2*, D is not simple, but finite in probability: D can be broken down into a family $D_1, \ldots, D_n$ of simple cycles. For each node in a D_i, there exists a LLF either per condition* (1) *or condition* (2)*, and therefore for all nodes in D.*
Case 3*, D is not finite in probability: D cannot contain any simple subcycles that are finite in probability. Therefore, D is a subgraph of C', and a GLF exists for D per condition* (2)*, implying condition* (2) *of Lemma 1.*

To check whether a node is in N or not, an enumeration of only the simple cycles of the automaton is necessary. The result is, that nodes in N can each be treated separately, since they only require the existence of a local Lyapunov function. Once a LLF is found, they can be removed from the automaton, and the cycle-based decomposition procedure from Theorem 4 can be applied to the remainder of the SCC. Next, we will apply these decomposition results to an example automaton.

5 Example

As an example, we present a simple cruise controller system with a probabilistic transition (see Figure 4). The variable v models the difference between actual speed and desired speed v_0 (i.e., the system dynamics are shifted, such that $v = 0$ is the desired speed), and a models the acceleration. Mode $A1$ represents a saturation, enforcing a maximum acceleration, while mode $A2$ is the standard acceleration/deceleration mode that is active whenever a is below the saturation level and v is close to 0. $B1$ and $B2$ represent two different service brake modes, modeling different brake dynamics that are chosen probabilistically with probabilities $p_1 > 0$ and $p_2 > 0$ (for instance depending on the inclination of the track or the weather conditions, which are considered random in this model). Additionally, with a (small) probability $p_3 > 0$, the service brake system might fail altogether, activating an emergency brake mode F. From any mode except F, it is also possible that the vehicle is ordered to stop in a regular manner, e.g., because the destination has been reached. This is modeled in mode E.

By applying the theorems of Section 4, it is possible to decompose the stability proof for this system into a number of subproofs. We want to show that the system will either converge to $v = a = 0$, or the vehicle will come to a stop in modes E or F, which means convergence to $v = -v_0$, where v_0 is the desired speed. The modes E and F each form a separate SCC and can therefore be treated separately through Theorem 3 and Corollary 1. Since the cycles formed by $A2$ and $B1$ and by $A2$ and $B2$ are both finite in probability, the nodes $B1$ and $B2$ can also be analyzed separately per Theorem 5. This yields one LMI

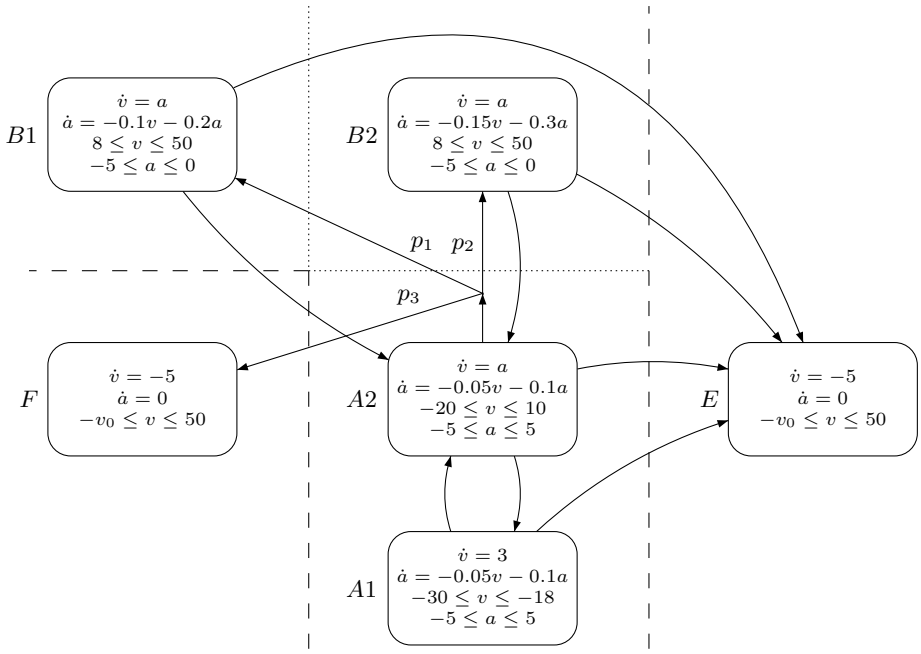


Fig. 4. Example automaton (guards and updates not pictured) and decomposition (dashed lines: Theorem 3, dotted lines: Theorem 5)

problem as per Theorem 2 for E, F, $B1$ and $B2$. Furthermore, one LMI problem for the cycle given by nodes $A1$ and $A2$ must be solved, including the two edges connecting them. If solutions for all of these independent LMI problems can be found, then all SCCs are GA-P, and all trajectories converge to either $v = a = 0$ or $v = -v_0$ with probability 1. In contrast, solving an LMI containing all constraints for all modes and transitions of the whole graph in one step is intractable in practice. The GLF for the SCC consisting of $A1$ and $A2$, as computed by an LMI solver, is given by $V_{A1} = 3.9804v^2 + 2.0001ta + 10.502a^2$ and $V_{A2} = 0.625v^2 + 2va + 10.5a^2$. Examples for LLF for the other nodes are $V_{B1} = 18.496v^2 + 26.314av + 100a^2, V_{B2} = 31.843v^2 + 40.98av + 100a^2$ and $V_E = V_F = v + v_0$.

6 Quantitative Analysis

In this section, we outline how the results of the previous section can be employed for quantitative stability analysis, i.e., the computation of convergence properties that lie below 1. Generally, this type of analysis is oriented along the SCCs of the hybrid system. If the convergence probability is strictly smaller than 1, then

some trajectories need to reach a permanent decision point, where a stabilizing decision is taken with probability $p < 1$ and a non-stabilizing decision is taken with probability $1 - p$. The permanent decision points that are visible in the hybrid automaton are the transitions between SCCs: once an SCC is left, a return is impossible. Therefore, such a "decision" is irreversible for the trajectory. This leads to the conclusion that quantitative stability analysis can be conducted with help of SCCs: those that are "visible in the graph structure" and those that are "hidden in the guards and dynamics" and can be exposed by using alternate hybrid automaton representations of the system. First, we define global attractivity with probability of less than 1, and then we give a theorem on the SCCs that are visible in the graph structure.

Definition 8 (Global Attractivity with Probability $p < 1$). *A probabilistic hybrid automaton H is* globally attractive in probability with probability p (GA-P(p)) *with respect to an equilibrium state x_e, if $P(\lim_{t\to\infty} x(t) = x_e) \geq p$.*

Theorem 6 (Quantitative Analysis). *Let H be a probabilistic hybrid automaton, consisting of an SCC C, that is GA-P wrt. 0 and a number of SCC $C_1, \ldots, C_m$ that are successors of C. Assume that Init only contains hybrid states with nodes of C as their discrete state. Furthermore, assume that each C_i is known to be GA-P(p_i) wrt. 0 for some $0 \leq p_i \leq 1$. Let c_i be a lower bound on the probability that a trajectory of H ends up in C_i Then, the sub-automaton of H consisting of C and the C_i is GA-P(p) wrt. 0, with $p = \sum c_i p_i$.*

Proof. Let (m_i) be a mode sequence belonging to a trajectory $x(t)$ of H. If (m_i) never leaves C, then $x(t)$ must converge to 0 with probability 1 since C is GA-P. If this is not the case, then (m_i) will enter SCC C_i with a probability of at least c_i. Since c_i is GA-P(p_i), $x(t)$ will then converge with probability p_i. Summing up over all C_i, we get a lower bound for the probability of convergence as $p = \sum c_i p_i$. Therefore, H is GA-P(p).

Lower bounds c_i can, for instance, be computed with the help of discrete time Markov decision processes, where the steady-state probability of ending up in an SCC is such a c_i. However, to obtain tight bounds on the stabilization property, it is necessary to have an automaton model of the system where all "branching points" are visible as transitions between SCC in the graph structure. At this point, methods for reachable set computation of hybrid systems can be employed to discover semantically equivalent automata (wrt. the continuous behavior) that have a "finer" SCC structure.

7 Conclusions

In this paper, we presented a scheme for decomposition of proofs of stability in probability for probabilistic hybrid automata. The decomposition results can be used to make automatic stability proofs through Lyapunov function computation more tractable in practice. Furthermore, the results give conditions, under

which stability properties of sub-automata to be composed transfer to the newly obtained larger automaton. Therefore, stable automata can be designed step by step, applying Lyapunov function arguments that are local in the graph during the design process. Furthermore, failure of a Lyapunov function is now less problematic, since it will be visible which computational step – and therefore which part of the automaton – caused the problem. This knowledge allows for an easier diagnosis of the problem that prevented the stability proof from succeeding. In general, we postulate that decompositional reasoning makes it easier to see what makes or breaks stability properties in probabilistic hybrid automata.

References

1. Branicky, M.: Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. IEEE Transactions on Automatic Control **43**(4) (April 1998) 475–482
2. Pettersson, S.: Analysis and Design of Hybrid Systems. PhD thesis, Chalmers University of Technology, Gothenburg (1999)
3. Parrilo, P.A.: Semidefinite programming relaxations for semialgebraic problems. Mathematical Programming, Series B (96) (2003) 293–320
4. Heemels, M., Weiland, S., Juloski, A.: Input-to-state stability of discontinuous dynamical systems with an observer-based control application. In: 10th International Conference on Hybrid Systems: Computations and Control. (2007)
5. Oehlerking, J., Theel, O.: Decompositional construction of Lyapunov functions for hybrid systems. In: 12th International Conference on Hybrid Systems: Computations and Control. Lecture Notes in Computer Science, Springer (2009)
6. Shiryaev, A.N.: Probability. Second edn. Springer (1996)
7. Kushner, H.J.: Stochastic stability. Lecture Notes in Mathematics (249) (1972) 97–124
8. Loparo, K.A., Feng, X.: Stability of stochastic systems. In: The Control Handbook. CRC Press (1996) 1105–1126
9. Boyd, S., El Ghaoui, L., Feron, E., Balakrishnan, V.: Linear Matrix Inequalities in System and Control Theory. Society for Industrial and Applied Mathematics (SIAM) (1994)
10. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2004)
11. Borchers, B.: CSDP, a C library for semidefinite programming. Optimization Methods and Software **10**(1) (1999) 613–623 https://projects.coin-or.org/Csdp/.
12. Romanko, O., Pólik, I., Sturm, J.F.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. (1999) http://sedumi.ie.lehigh.edu.