Approximate Abstraction of Stochastic Hybrid Automata[*]

A. Agung Julius

Dept. Electrical and Systems Engineering,
University of Pennsylvania,
200 South 33rd Street, Philadelphia PA-19104, USA
agung@seas.upenn.edu

Abstract. This paper discusses a notion of approximate abstraction for linear stochastic hybrid automata (LSHA). The idea is based on the construction of the so called stochastic bisimulation function. Such function can be used to quantify the distance between a system and its approximate abstraction. The work in this paper generalizes our earlier work for jump linear stochastic systems (JLSS). In this paper we demonstrate that linear stochastic hybrid automata can be cast as a modified JLSS and modify the procedure for constructing the stochastic bisimulation function accordingly. The construction of quadratic stochastic bisimulation functions is essentially a linear matrix inequality problem. In this paper, we also discuss possible extensions of the framework to handle nonlinear dynamics and variable rate Poisson processes. As an example, we apply the framework to a chain-like stochastic hybrid automaton.

1 Introduction

Stochastic hybrid systems are widely used to model physical and engineering systems, in which the continuous dynamics has many modes or discontinuities, as well as stochastic behavior [1]. Applications of stochastic hybrid systems can be found in telecommunication networks [2], systems biology [3], air traffic management [4], etc.

There are several available modelling formalisms for stochastic hybrid systems. One of the earliest frameworks is the one in [5], where a general type of stochastic hybrid systems, whose continuous dynamics is described by diffusion stochastic differential equation [6], is presented. Mode switching occurs when some invariant condition in the corresponding mode is violated. Another framework that involves multimodal diffusion equation is the switched diffusion processes [7]. There are also modelling frameworks, where the continuous dynamics is described by ordinary differential equation, such as the piecewise deterministic Markov processes [8], stochastic hybrid systems [2], etc. In these frameworks, the switching is modelled as a Poisson process. For a more thorough

[*] This research is supported by the National Science Foundation Presidential Early CAREER (PECASE) Grant 0132716.

J. Hespanha and A. Tiwari (Eds.): HSCC 2006, LNCS 3927, pp. 318–332, 2006.

survey on the modelling formalisms for stochastic hybrid systems, the interested reader is referred to [1].

Researchers have been working on how to tame the increasing complexity of system analysis. There are two approaches. The first approach is to develop a framework that allows the computation to be performed in a modular fashion. The other approach is to develop a framework that allows abstraction of the complex system. By abstraction we mean building a simpler system that is, in some sense, equivalent to the complex system. The computation is then performed on the simpler system and the equivalence guarantees that the results can be carried over into the complex system. The discussion in this paper pertains to the second approach.

Bisimulation is a concept of system equivalence that is widely used for abstraction of complex systems. Notions of exact bisimulation for some classes of stochastic hybrid systems have been recently developed in [9, 10]. In [9], a category theoretical notion exact bisimulation for general stochastic hybrid systems is discussed, while [10] treats the issue of exact bisimulation for the so called communicating piecewise deterministic hybrid systems. In this paper, we relax the requirement that the abstraction is exactly equivalent to the original system. Instead, we require that they are only *approximately* equivalent [11, 12]. We then need to define a metric, with which we can measure the distance between systems and hence the quality of the abstraction. In [13, 14], the authors develop some metrics for labelled Markov processes and probabilistic transition systems, inspired by the Hutchinson metric, which gives the distance between two distributions of the transition probability. The approach that we take in this paper differs from that, since we use a different kind of metric. The metric that we use is based on the L_∞ distance between the output trajectories of the systems. We develop a theory of approximate bisimulation for a class of stochastic hybrid automata, in which the continuous dynamics is modelled by stochastic differential equations and the switches are modelled as Poisson processes. This class of systems is called the *linear stochastic hybrid automata* (LSHA).

The approach that we take in this paper is by computing the so called *stochastic bisimulation function.* The stochastic bisimulation function is used to quantify the quality of the abstraction. This approach has been used in [15] for jump linear stochastic systems (JLSS). The jump linear stochastic systems are stochastic systems whose dynamics is described by a stochastic differential equation with Poisson jumps in the continuous state. Thus, an LSHA can be thought of as a generalization of JLSS, as in LSHA it is possible to have multiple modes for the continuous dynamics. However, in this paper we also show that it is possible to cast an LSHA as a modified JLSS, and hence we can compute the stochastic bisimulation function for LSHA by modifying the procedure for JLSS. We also demonstrate that the construction of quadratic stochastic bisimulation functions for LSHA can be cast as a tractable linear matrix inequality problem. Further, we also discuss possible extensions of the framework to deal with nonlinear dynamics and variable rate Poisson processes.

2 Linear Stochastic Hybrid Automata

In this paper, we formally define a linear stochastic hybrid automaton (LSHA) as a 5-tuple $\mathcal{A} = (L, n, m, T, F)$, where

- L is a finite set, which is the set of locations or discrete states. The number of locations is denoted by $|L|$.
- $n : L \to \mathbb{N}$, where for every $l \in L$, $n(l)$ is the dimension of the continuous state space in location l,
- $m \in \mathbb{N}$, is the dimension of the output of the automaton $\mathcal{A}$,
- T is the set of random transitions. A transition $\tau \in T$ can be written as a 4-tuple $(l, \lambda_\tau, l', R_\tau)$. This is a transition from location $l \in L$ to $l' \in L$ that is triggered by a Poisson process with intensity $\lambda_\tau \in \mathbb{R}_+$. The matrix $R_\tau \in \mathbb{R}^{n(l') \times n(l)}$ is the linear reset map associated with the transition τ. The number of transitions is denoted by $|T|$.
- F defines the continuous dynamics in each location. For every $l \in L$, $F(l)$ is a triple (A_l, G_l, C_l), where $A_l \in \mathbb{R}^{n(l)\times n(l)}$, $G_l \in \mathbb{R}^{n(l)\times n(l)}$and $C_l \in \mathbb{R}^{m \times n(l)}$.

The state space of the automaton can be written as

$$\mathcal{X} = \bigcup_{i=1}^{|L|} \left(\{l_i\} \times \mathbb{R}^{n(l_i)} \right). \tag{1}$$

We also define the functions $\texttt{source} : T \to L$ and $\texttt{dest} : T \to L$, such that if $\tau \in T$ is $(l, \lambda_\tau, l', R_\tau)$ then

$$\texttt{source}(\tau) = l, \ \texttt{dest}(\tau) = l'. \tag{2}$$

The semantics of the linear stochastic hybrid automaton $\mathcal{A}$ can be explained as follows. The state trajectory $\xi_t = (l_t, x_t)$ of the LSHA $\mathcal{A}$ is inherently a stochastic process. Every state trajectory that the automaton executes is a realization of the process. In each location $l \in L$, the continuous state of the system satisfies the following stochastic differential equation (SDE).

$$dx_{l,t} = A_l x_{l,t}\ dt + G_l x_{l,t}\ dw_t, \tag{3a}$$

$$y_t = C_l x_{l,t}, \tag{3b}$$

$$x_{l,t} \in \mathbb{R}^{n(l)}, y_t \in \mathbb{R}^m. \tag{3c}$$

The process w_t is an $\mathbb{R}$ valued standard Brownian motion, where $E[w_t^2] = t$. The $\mathbb{R}^m$ valued stochastic process y_t is the output/observation of automaton A.

Remark 1. In general, it is possible to incorporate multi dimensional Brownian motions in the framework. In this case, the term $G_l x_{lt}\ dw_t$ in (3a) would be replaced by $\sum_{i=1}^{N} G_{l,i} x_{l,t}\ dw_{i,t}$ to incorporate an N-dimensional Brownian motion. Hereafter, we stick to the one dimensional Brownian motion for simplicity.

Denote the set of outgoing transitions of a location as

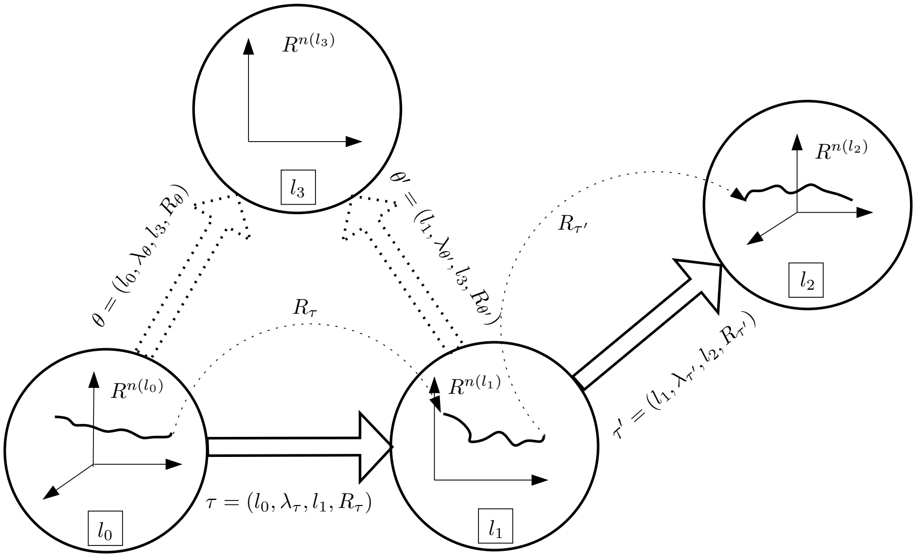
$$\texttt{out} : L \to 2^T,\ \texttt{out}(l) := \{\tau \in T \mid \texttt{source}(\tau) = l\}\,, \tag{4}$$

and $|\texttt{out}(l)|$ as the number of outgoing transitions from location l. While the system is evolving in a location $l \in L$, each transition in $\texttt{out}(l)$ is represented by an active Poisson process. Each of these Poisson processes has a constant rate indicated by the transition. The first Poisson process to generate a point triggers a transition. Suppose that $\tau = (l, \lambda_\tau, l', R_\tau)$ is the transition that corresponds to the first process that generates a point (at time t), then the evolution of the system will switch to location l'. The matrix R_τ defines a linear reset map,

$$x_t = R_\tau x_{t^-}\,, \tag{5}$$

where $x_{t^-} := \lim_{s \uparrow t} x_s$.

Figure 1 illustrates a realization of the execution of an LSHA. In Figure 1, the execution starts in location l_0 by following the SDE that defines the dynamics in the location. The set of outgoing transitions from l_0, $\texttt{out}(l_0) = \{\tau, \theta\}$. In this particular realization, the Poisson process associated with τ generates a point before that of θ. Hence, a transition occurs that brings the trajectory to location $\texttt{dest}(\tau) = l_1$. The continuous state of the trajectory is reset by the linear map



$R^{n(l_3)}$
l_3
$\theta = (l_0, \lambda_\theta, l_3, R_\theta)$
$\theta' = (l_1, \lambda_{\theta'}, l_3, R_{\theta'})$
$R^{n(l_2)}$
$R_{\tau'}$
l_2
R_τ
$R^{n(l_0)}$
$R^{n(l_1)}$
$\tau' = (l_1, \lambda_{\tau'}, l_2, R_{\tau'})$
$\tau = (l_0, \lambda_\tau, l_1, R_\tau)$
l_0
l_1

Fig. 1. An illustration of the execution of an LSHA. The solid bold arrows represent transitions between locations that occur. The dotted bold arrows indicate transitions that do not occur, since the associated Poisson process do not generate a point fast enough. The dotted arrows denote the linear reset maps associated with the transitions that occur.

R_τ. In the new location, the continuous dynamics proceeds with the SDE that defines the dynamics in location l_1. The set $\mathtt{out}(l_1) = \{\tau', \theta'\}$. In this particular realization, the Poisson process associated with τ' generates a point before that of θ'. Hence, a transition occurs that brings the trajectory to location l_2. The continuous state of the system is then subsequently reset by the linear map $R_{\tau'}$.

3 Approximate Abstraction of LSHA

In this paper we will develop the notion of approximate abstraction of linear stochastic hybrid automata. The notion of approximate abstraction is constructed using the concept of stochastic bisimulation functions [15][1].

A stochastic bisimulation function is defined between two LSHA, $\mathcal{A}_i = (L_i, n_i, m, T_i, F_i)$, $i = 1, 2$. Notice that we assume that the outputs of the automata have the same dimension. We denote the state space of $\mathcal{A}_i$ as $\mathcal{X}_i$, $i = 1, 2$. See (1).

Definition 1. *[15] A function $\phi : \mathcal{X}_1 \times \mathcal{X}_2 \to \mathbb{R}_+ \cup \{+\infty\}$ is a* ***stochastic bisimulation function*** *between $\mathcal{A}_1$ and $\mathcal{A}_2$ if the following statements hold.*
(i) Suppose that $\xi_i = (l_i, x_i) \in \mathcal{X}_i, i = 1, 2$, then

$$\phi(\xi_1, \xi_2) \geq \left\| C_{1,l_1} x_1 - C_{2,l_2} x_2 \right\|^2 = \left\| y_1 - y_2 \right\|^2 ,$$

where $\|\cdot\|$ denotes the Euclidean distance in $\mathbb{R}^m$,
(ii) the stochastic process $\phi_t := \phi(\xi_{1,t}, \xi_{2,t})$ is a supermartingale for any distribution of the initial state.

Remark 2. The definition of stochastic bisimulation function in this paper does not exhibit the game theoretic aspect as that in [15]. This is because we do not model disturbance as a source of nondeterminism in this framework. We could add disturbance as another affine term in (3a), and we can see later in Section 4 that the theoretical framework that we develop in this paper can be extended easily to cover this case. However, this would be done at significant computational expense.

The following theorem describes the relation between the stochastic bisimulation function and the difference between the output of $\mathcal{A}_1$ and $\mathcal{A}_2$.

Theorem 1. *(adapted from [15])Given two LSHA, $\mathcal{A}_i = (L_i, n_i, m, T_i, F_i)$, $i = 1, 2$, and $\phi(\cdot)$ a stochastic bisimulation function. The following relation holds.*

$$P\left\{ \sup_{0 \leq t < \infty} \left\| y_{1,t} - y_{2,t} \right\|^2 \geq \delta \,\middle|\, (\xi_{1,0}, \xi_{2,0}) \right\} \leq \frac{\phi(\xi_{1,0}, \xi_{2,0})}{\delta}. \tag{6}$$

Proof. Following Definition 1, $\phi(\xi_{1t}, \xi_{2t})$ is a supermartingale. Since $\phi(\xi_{1t}, \xi_{2t})$ is a nonnegative supermartingale, we have the following result [16].

$$P\left\{ \sup_{0 \leq t < \infty} \phi(\xi_{1,t}, \xi_{2,t}) \geq \delta \,\middle|\, (\xi_{1,0}, \xi_{2,0}) \right\} \leq \frac{\phi(\xi_{1,0}, \xi_{2,0})}{\delta}. \tag{7}$$

[1] The work is inspired by the nonstochastic version in [12].

Moreover, since $\phi(\xi_1, \xi_2) \geq \|y_1 - y_2\|^2$ by construction, we also have that

$$P\left\{ \sup_{0\leq t<\infty} \|y_{1,t} - y_{2,t}\|^2 \geq \delta \middle| (\xi_{1,0}, \xi_{2,0}) \right\} \leq P\left\{ \sup_{0\leq t<\infty} \phi(\xi_{1,t}, \xi_{2,t}) \geq \delta \middle| (\xi_{1,0}, \xi_{2,0}) \right\}. \tag{8}$$

Hence we have (6).

The stochastic bisimulation function can be used to guarantee that the difference between the output of the original system and its abstraction will not exceed a given bound, with a certain probability. The difference between the outputs is measured in the sense of L_∞. This makes this approach particularly suitable for analyzing safety/reachability property of the system, as it is illustrated in the following.

Given a complex system represented by an LSHA $\mathcal{A}_1$ and its simpler abstraction $\mathcal{A}_2$. Suppose that $\phi(\cdot)$ is a stochastic bisimulation function between the two automata, and that the initial condition of the composite system is $(\xi_{1,0}, \xi_{2,0})$. Given the unsafe set for the automaton $\mathcal{A}_1$, $\texttt{unsafe}_1 \subset \mathbb{R}^m$, we can construct another set $\texttt{unsafe}_2 \subset \mathbb{R}^m$, which is the δ neighborhood of $\texttt{unsafe}_1$ for some $\delta > 0$. That is,

$$\texttt{unsafe}_2 = \{y \mid \exists y' \in \texttt{unsafe}_1, \|y - y'\| \leq \delta\}. \tag{9}$$

We define the events $\mathbf{unsafe}_i := \{\exists t \geq 0 \text{ s.t. } y_{i,t} \in \texttt{unsafe}_i\}, i = 1, 2$. The following theorem holds [15].

Theorem 2. *The following relation between the safety properties of the automata holds.*

$$P\{\mathbf{unsafe}_1\} \leq P\{\mathbf{unsafe}_2\} + \frac{\phi(\xi_{1,0}, \xi_{2,0})}{\delta^2}. \tag{10}$$

Theorem 2 tells us that we can get an upper bound of the risk of the complex system by performing the risk calculation on the simple abstraction and adding a factor that depends on the stochastic bisimulation function.

4 Casting LSHA as Jump Linear Stochastic Systems

We have seen that we need to construct a stochastic bisimulation function between an LSHA and its abstraction, to measure the quality of abstraction. In this section, we demonstrate how an LSHA can be cast as a modified jump linear stochastic system (JLSS) [15]. We shall then use the tools that have been developed for JLSS to construct stochastic bisimulation functions for LSHA.

First, we introduce the structure of a jump linear stochastic system. A *jump linear stochastic system* (JLSS) can be modeled as a stochastic system that satisfies the following stochastic differential equation.

$$dx_t = Ax_t\, dt + Gx_t\, dw_t + \sum_{i=1}^{N} Q_i x_t\, dp_t^i, \tag{11a}$$

$$y_t = Cx_t. \tag{11b}$$

Here, y_t is the output of the system, the process w_t is a standard Brownian motion, while p_t^i is a Poisson process with a constant rate λ_i. We assume that the Poisson processes and the Brownian motion are independent of each other.

Remark 3. The model of jump linear stochastic system that we use here is slightly different from that in [15]. The difference is in the fact that the we use a linear diffusion term (i.e. Gx_t), while in [15] a constant term is used. With this modification, we make sure that the origin is an equilibrium with probability 1. That is, $P\{x_t \neq 0, t \geq 0 | x_0 = 0\} = 0$. As we shall see later, this property is exploited to cast LSHA as JLSS.

Given an LSHA $\mathcal{A} = (L, n, m, T, F)$ as in Section 2, the following is an algorithm to define a JLSS, structured as in (11), that represents $\mathcal{A}$.

- The state space of the JLSS has the dimension of $\sum_{i=1}^{|L|} n(l_i)$, $l_i \in L$.
- The A and G matrices of the JLSS has a block diagonal structure, with $|L|$ blocks. That is,

$$A := \begin{bmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_{|L|} \end{bmatrix}, G := \begin{bmatrix} G_1 & 0 & \cdots & 0 \\ 0 & G_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & G_{|L|} \end{bmatrix}. \qquad (12)$$

  where $A_i := A_{l_i}$ and $G_i := G_{l_i}$ are the A and G matrices of the LSHA in location l_i.
- The C matrix of the JLSS is structured as $C := \begin{bmatrix} C_1 \; C_2 \cdots C_{|L|} \end{bmatrix}$, where $C_i := C_{l_i}$ is the C matrix of the LSHA in location l_i.
- There are $|T|$ independent Poisson processes. Thus, $N = |T|$. Each Poisson process represents a transition in T. Denote the transitions as $T = \{\tau_i\}_{1 \leq i \leq |T|}$ and $\tau_i := (loc_i, \lambda_i, loc_i', R_i)$. Then the Poisson process p_t^i has the rate of λ_i, and the matrix Q_i has a block diagonal structure as A and G, where

$$Q_i := \left[\begin{array}{cccccc} 0 \cdots & & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & & & & \vdots \\ 0 & & -I & 0 & & 0 \\ 0 & & R_i & 0 & & 0 \\ \vdots & & & & \ddots & \vdots \\ 0 \cdots & & 0 & 0 & \cdots & 0 \end{array}\right] \begin{array}{l} \cdot \\ \cdot \\ \longleftarrow loc_i \\ \longleftarrow loc_i' \\ \cdot \\ \cdot \end{array}, \qquad (13)$$

  that is, almost all the blocks are zero, except for two blocks:
  (i) the diagonal block associated with loc_i, which is $-I$, and
  (ii) the block whose row is associated with loc_i' and its column with loc_i, which is R_i.

The idea behind this procedure is as follows. We formulate a JLSS with $|L|$ invariant dynamics. That is, the state space can be written as the direct sum of $|L|$ subspaces, each of which is invariant with respect to the following dynamics:

$$dx_t = Ax_t \; dt + Gx_t \; dw_t. \qquad (14)$$

Each invariant subspace represents a location in the LSHA. Further, we can observe that the origin is also invariant with respect to (14) (see Remark 3). As the result, if we start the evolution of the system in one of the invariant subspaces (hence, in one of the locations of the LSHA), the trajectory will remain in the subspace. Let us call the location l. When a Poisson process generates a point, if the process does not correspond to a transition whose source location is l, then the reset map does not change the continuous state of the system. This is due to the construction of (13). If the source location is l and the target is, say, l', then the continuous state is reset to another invariant space that corresponds to the location l'.

One apparent difference between the JLSS realization of the system and the original LSHA is that in the LSHA, only the Poisson processes in the active location are active. However, this difference does not affect the probabilistic properties of the trajectories, since Poisson processes are memoryless [8]. When we enter a location, it does not matter if we assume that the Poisson processes in the location are just started or that they have been running before.

5 Computation of the Stochastic Bisimulation Function

In the previous section we demonstrate how we can cast a linear stochastic hybrid automaton (LSHA) as a jump linear stochastic system (JLSS). In general, we can then exploit the available construction of quadratic stochastic bisimulation function for JLSS [15], and apply it for LSHA. However, since we also modified the definition (see Remark 3), we also need to modify the procedure for constructing a stochastic bisimulation function.

Given two JLSS, for $i = 1, 2$,

$$S_i : \begin{cases} dx_{i,t} = A_i x_{i,t} \; dt + G_i x_{i,t} \; dw_t + \sum_{j=1}^{N} Q_{ij} x_t \; dp_t^j, \\ y_{it} = C_i x_{i,t}. \end{cases} \tag{15}$$

We define the following composite system

$$x_t := \begin{bmatrix} x_{1,t} \\ x_{2,t} \end{bmatrix}, y_t := y_{1,t} - y_{2,t}, A := \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}, G := \begin{bmatrix} G_1 & 0 \\ 0 & G_2 \end{bmatrix}, \tag{16a}$$

$$Q_j := \begin{bmatrix} Q_{1j} & 0 \\ 0 & Q_{2j} \end{bmatrix}, C := \begin{bmatrix} C_1 & -C_2 \end{bmatrix}. \tag{16b}$$

Hence we have the following system:

$$S : \begin{cases} dx_t = A x_t \; dt + G x_t \; dw_t + \sum_{j=1}^{N} Q_j x_t \; dp_t^j, \\ y_t = C x_t. \end{cases} \tag{17}$$

As mentioned above, we want to construct a quadratic stochastic bisimulation function. Thus, we want to find the conditions for a function of the form

$$\phi(x) = x^T M x, \tag{18}$$

to satisfy Definition 1. We can observe that the process $\phi_t := \phi(x_t)$ satisfies the following SDE.

$$d\phi_t = \frac{\partial \phi}{\partial x} dx_t + \frac{1}{2} dx_t^T \frac{\partial^2 \phi}{\partial x^2} dx_t = 2x_t^T M \left(Ax_t \; dt + Gx_t \; dw_t + \sum_{j=1}^{N} Q_j x_t \; dp_t^j \right)$$
$$+ \, x_t^T G^T M G x_t \; dt + \sum_{i,j \in \{1,2,\cdots,N\}} x_t^T Q_i^T M Q_j x_t \; dp_t^i dp_t^j. \tag{19}$$

Using the fact that the Poisson processes are independent from each other, we can establish that the expectation of the last term of the right hand side satisfies the following relation,

$$E\left[x_t^T Q_i^T M Q_j x_t \; dp_t^i dp_t^j\right] = \begin{cases} E\left[x_t^T Q_i^T M Q_j x_t\right] \lambda_i \lambda_j dt^2, & i \neq j, \\ E\left[x_t^T Q_j^T M Q_j x_t\right] (\lambda_j dt + \lambda_j^2 dt^2), & i = j. \end{cases}$$

The expectation of ϕ_t then satisfies the following equation.

$$\frac{dE[\phi_t]}{dt} = E\left[x_t^T \Theta x_t\right], \tag{20}$$

where

$$\Theta := 2MA + 2M \sum_{i=1}^{N} \lambda_i Q_i + G^T M G + \sum_{i=1}^{N} \lambda_i Q_i^T M Q_i. \tag{21}$$

Theorem 3. *The function $\phi(x) = x^T M x$ is a stochastic bisimulation function for the systems in (15) if and only if $M \geq C^T C$, and $\Theta \leq 0$.*

This theorem in an immediate consequence of Definition 1. The problem of finding M such that the conditions in Theorem 3 hold is a linear matrix equality (LMI) problem.

Remark 4. If we see the quadratic stochastic bisimulation function as a stochastic Lyapunov function, then the conditions in Theorem 3 guarantee that y_t converges to 0 in probability. However, in this paper we are not interested in the asymptotic behavior of y_t (the convergence), rather we are interested in the bound on the magnitude of y_t.

6 Extensions of the LSHA

In this section we discuss two possible extensions of the linear stochastic hybrid automata, and the implications of the extensions to the computation of the stochastic bisimulation function.

6.1 Nonlinear Stochastic Hybrid Automata

Consider a linear stochastic hybrid automata $A = (L, n, m, T, F)$. Suppose that instead of the linear dynamics in (3), we assume that the dynamics in location $l \in L$ satisfies a nonlinear SDE of the following form.

$$dx_{l,t} = a_l(x_{l,t})\ dt + g_l(x_{l,t})\ dw_t, \tag{22a}$$

$$y_t = c_l(x_{l,t}), \tag{22b}$$

$$x_{l,t} \in \mathbb{R}^{n(l)}, y_t \in \mathbb{R}^m. \tag{22c}$$

We assume that for all $l \in L$,

$$a_l(0) = 0, g_l(0) = 0. \tag{23}$$

This assumption renders the origin invariant under the dynamics described by (22). In general, we only need to have a point that is invariant under (22).

Furthermore, assume that instead of the linear reset map (5), the reset function of a given transition $\tau \in T$ follows the relation $x_t = r_\tau(x_{t^-})$,where $x_{t^-} := \lim_{s\uparrow t} x_t$.

Analogous to the discussion in Section 4, we can show that the nonlinear version of the stochastic hybrid automata can be cast as a nonlinear version of the jump linear stochastic systems, that is, systems of the form.

$$dx_t = a(x_t)\ dt + g(x_t)\ dw_t + \sum_{i=1}^{N} q_i(x_t)\ dp_t^i, \tag{24a}$$

$$y_t = c(x_t). \tag{24b}$$

Furthermore, given two systems, for $i = 1, 2$,

$$S_i : \begin{cases} dx_{i,t} = a_i(x_{i,t})\ dt + g_i(x_{i,t})\ dw_t + \sum_{j=1}^{N} q_{ij}(x_t)\ dp_t^j, \\ y_{i,t} = c_i(x_{i,t}), \end{cases} \tag{25}$$

we can form a composite system in the form of (24), by following a construction analogous to (16).

Definition 1 is still valid for the nonlinear version of the stochastic hybrid automata. Hence, the results that relate the stochastic bisimulation function with approximate abstraction and safety verification still hold.

Suppose that we are given a smooth function $\phi(\cdot)$ of the state of the composite system (24). It can be verified that the evolution of the expectation of $\phi_t := \phi(x_t)$ can be written as:

$$\frac{dE\left[\phi_t\right]}{dt} = E\left[\frac{\partial \phi}{\partial x} a(x_t)\right] + \frac{1}{2} E\left[g^T(x_t)\frac{\partial^2 \phi}{\partial x^2} g(x_t)\right] + \sum_{j=1}^{N} \lambda_j E[\phi(x_t + q_j(x_t)) - \phi(x_t)]. \tag{26}$$

Define

$$\Theta(x) := \frac{\partial \phi}{\partial x} a(x) + \frac{1}{2} g^T(x) \frac{\partial^2 \phi}{\partial x^2} g(x) + \sum_{j=1}^{N} \lambda_j (\phi(x + q_j(x)) - \phi(x)), \tag{27}$$

then $\frac{dE[\phi_t]}{dt} = E[\Theta(x_t)]$.

Thus, to compute a general stochastic bisimulation function, we need to find a smooth function ϕ such that

$$\phi(x) \geq (c(x))^2, \ \Theta(x) \leq 0. \tag{28}$$

An automatic procedure for constructing such a function ϕ does not exist. However, if we assume that all the functions involved are polynomials, this problem can be cast as a *sum-of-squares* problem. There is a software tool that can be used to solve such problems, that is SOSTOOLS [17].

6.2 LSHA with Variable Rate Poisson Processes

In this subsection, we discuss the LSHA where the rate of the Poisson processes are assumed to be functions of the continuous state. This type of LSHA can still be cast as a JLSS of the form (11). The only difference is that now the Poisson processes $\{p_t^j\}_{1\leq j\leq N}$ have rates that depend on the continuous state, $\lambda_j(x)$ instead of a constant rate. We also assume that for every $j \in \{1,2,\cdots,N\}$, there exist $L_j \geq 0$ and $U_j \geq L_j$ such that for every continuous state x,

$$L_j \leq \lambda_j(x) \leq U_j. \tag{29}$$

Thus, for all x, the vector $\left[\lambda_1(x)\ \lambda_2(x)\ \cdots\ \lambda_N(x)\right]$ is contained in a hyper rectangle defined by the lower and upper bounds in (29). Let $\Gamma \in \mathbb{R}^{2^N\times N}$ be the matrix with all the 2^N vertices of the hyper rectangle. That is,

$$\Gamma := \begin{bmatrix} L_1 & L_2 & \cdots & L_{N-1} & L_N \\ L_1 & L_2 & \cdots & L_{N-1} & U_N \\ L_1 & L_2 & \cdots & U_{N-1} & L_N \\ \vdots & \vdots & & \vdots & \vdots \\ U_1 & U_2 & \cdots & U_{N-1} & U_N \end{bmatrix}.$$

Assuming quadratic stochastic bisimulation function $\phi(x) = x^T M x$, we can show that in the case of variable rate Poisson processes, equations (20) and (21) become

$$\frac{dE[\phi_t]}{dt} = E\left[x_t^T \Theta(x_t) x_t\right], \tag{30}$$

where

$$\Theta(x) := 2MA + 2M\sum_{i=1}^{N}\lambda_i(x)Q_i + G^T MG + \sum_{i=1}^{N}\lambda_i(x)Q_i^T MQ_i. \tag{31}$$

Theorem 4. *Let M be a symmetric matrix that satisfies*

$$M \geq C^T C, \tag{32a}$$

$$\Theta_i := 2MA + 2M\sum_{j=1}^{N}\Gamma_{ij}Q_i + G^T MG + \sum_{j=1}^{N}\Gamma_{ij}Q_i^T MQ_i \leq 0, \tag{32b}$$

for $1 \leq i \leq 2^N$, then $\phi(x) = x^T Mx$ is a stochastic bisimulation function.

Proof. We need to show that (32b) implies that $\phi_t = \phi(x_t)$ is a supermartingale for any distribution of the initial state. Suppose that (32b) holds, then for any x, the matrix $\Theta(x)$ can be written as a convex combination of $\{\Theta_i\}_{1 \le i \le 2^N}$. Therefore, $\Theta(x) \le 0$. From (30) we can infer that ϕ_t is a supermartingale for any distribution of the initial state.

The problem of finding M such that (32) holds can also be cast as a linear matrix inequality problem.

7 Example: Chain-Like Linear Stochastic Hybrid Automata

In this section we present an example, where we apply the framework of approximate abstraction of linear stochastic hybrid automata. The original automaton $\mathcal{A}$ has a chain like structure, with 21 locations. See Figure 2.

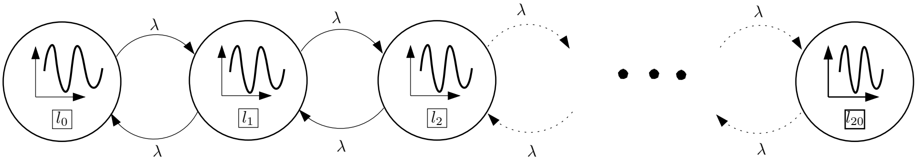


Fig. 2. The chain-like automaton $\mathcal{A}$ with 21 locations

Chain-like automata is a structure that can be found in modelling of systems that involve birth and death process. That is, each location represents the number of a certain object in the system, for example, persons in a queue or molecules in a chemical reaction. Researchers have been working towards approximating such systems in a way that allows for both fast and accurate simulations [18], as well as faster computation [19].

Adjacent locations in the automaton $\mathcal{A}$ are connected by a pair of transitions with constant rate $\lambda = 0.02$. The continuous dynamics of $\mathcal{A}$ is such that the dynamics changes gradually from location l_0 to location l_{20}. The stochastic differential equation that describes the dynamics in location l_i, $0 \le i \le 20$, is as follows.

$$\begin{aligned} dx_{i,t} &= A_i x_{i,t}\ dt + G_i x_t\ dw_t, \\ y_t &= C_i x_{i,t}, \text{ where} \end{aligned}$$

$$\begin{aligned} A_i &= \begin{bmatrix} -0.01 & -0.1(1+\alpha\cdot i) \\ 0.1(1+\alpha\cdot i) & -0.01 \end{bmatrix}, G_i = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}, \\ C_i &= \begin{bmatrix} 0 \ 1 \end{bmatrix}, i = 0 \ldots 20. \end{aligned}$$

We are going to apply the procedure for several values of α.

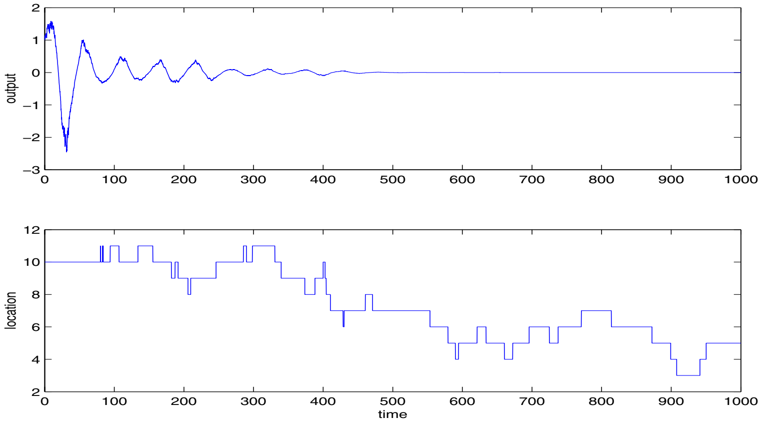


Fig. 3. A realization of the output trajectory (top) and the location (bottom) of the linear stochastic hybrid automaton $\mathcal{A}$

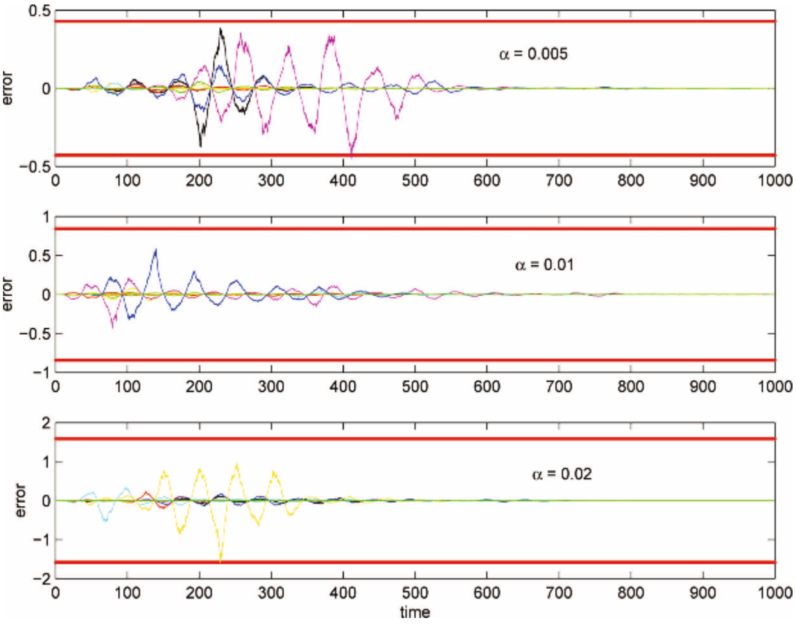


Fig. 4. Ten realizations of the error trajectory for each of the α value. The parallel lines indicate the 90% confidence interval stipulated by the stochastic bisimulation functions.

We can easily observe that the continuous dynamics in each location is a damped 2-dimensional oscillator driven by Brownian motion. A realization of the output of $\mathcal{A}$ is plotted in Figure 3. As we go from location l_0 to l_{20}, the frequency of the oscillation increases. We want to see if we can approximate $\mathcal{A}$ with another automaton $\mathcal{A}'$ that has only one location. The continuous dynamics of $\mathcal{A}'$ is the same as that in location l_{10} of $\mathcal{A}$. Hence we compute a stochastic bisimulation function between $\mathcal{A}$ and $\mathcal{A}'$. The computation is done by solving the linear matrix inequality problem explained in Section 5. We perform the computation using the tool YALMIP [20].

Three different values for α are used, namely 5×10^{-3}, 10^{-2}, and 2×10^{-2}. For these values of α, the ratio between the oscillation frequency in location l_{20} and l_0 are 1.1, 1.2, and 1.4 respectively. We simulate the execution of the original automaton $\mathcal{A}$ and its abstraction $\mathcal{A}'$. In the simulation we use $[1\ 1]^T$ as the initial condition for the continuous dynamics, and assume that automaton $\mathcal{A}$ starts in location l_{10}. With the computed stochastic bisimulation function, we can also compute the 90% confidence interval for the error between the outputs of $\mathcal{A}$ and $\mathcal{A}'$ (see Theorem 1).

In Figure 4 we can see ten realizations of the error trajectory for each of the value of α. The 90% confidence intervals are also shown. We can observe that the quadratic stochastic bisimulation function seems to give a good estimate for the error, as the confidence intervals seem quite tight. We can also observe that as the dynamics in the locations vary more, the error in the approximation becomes larger.

8 Conclusions

In this paper we develop the notion of approximate bisimulation for linear stochastic hybrid automata. The approach is based on the construction of a stochastic bisimulation function that can be used as a tool to quantify the distance between an automaton and its abstraction. We show that this notion of distance relates nicely with the safety properties of the automata (see Theorem 2). An example of the application of the results is provided at the end of the paper, where we evaluate approximate abstraction of a chain-like stochastic hybrid automaton.

We also discuss two possible extensions to the framework, namely when the continuous dynamics is nonlinear, and when the rates of the Poisson processes are not constant. In each case, we show how the computation of the stochastic bisimulation function will be. Future extensions of the work presented in this paper can be highlighted as follows. Issues such as incorporating nondeterminism (see Remark 2) and establishing necessary and sufficient conditions for the existence of the stochastic bisimulation function are possible research direction in the future. Another interesting direction is exploring different construction procedure for the stochastic bisimulation function, for example, using polynomial functions (which are generalization of quadratic functions).

Acknowledgements. The author would like to thank Antoine Girard and George Pappas for the discussions during the preparation of this paper, and Bruce Krogh for a valuable suggestion on relating JLSS and LSHA.

References

1. Pola, G., Bujorianu, M., Lygeros, J., Benedetto, M.D.: Stochastic hybrid models: an overview. In: Proc. IFAC Conf. Analysis and Design of Hybrid Systems, St. Malo, IFAC (2003)
2. Hespanha, J.P.: Stochastic hybrid systems: applications to communication networks. [21] 387 – 401
3. Hu, J., Wu, W.C., Sastry, S.: Modeling subtilin production in bacillus subtilis using stochastic hybrid systems. [21] 417 – 431
4. Glover, W., Lygeros, J.: A stochastic hybrid model for air traffic control simulation. [21] 372 – 386
5. Hu, J., Lygeros, J., Sastry, S.: Towards a theory of stochastic hybrid systems. In Lynch, N., Krogh, B.H., eds.: Hybrid Systems: Computation and Control. Volume 1790 of Lecture Notes in Computer Science., Springer Verlag (2000) 160–173
6. Oksendal, B.: Stochastic differential equations: an introduction with applications. Springer-Verlag, Berlin (2000)
7. Ghosh, M.K., Arapostathis, A., Marcus, S.: Ergodic control of switching diffusions. SIAM Journal on Control and Optimization **35**(6) (1997) 1952–1988
8. Davis, M.H.A.: Markov models and optimization. Chapman and Hall, London (1993)
9. Bujorianu, M.L., Lygeros, J., Bujorianu, M.C.: Bisimulation for general stochastic hybrid systems. [22] 198–214
10. Strubbe, S., van der Schaft, A.J.: Bisimulation for communicating piecewise deterministic Markov processes. [22] 623–639
11. Ying, M., Wirsing, M.: Approximate bisimilarity. In Rus, T., ed.: AMAST 2000. Volume 1816 of Lecture Notes in Computer Science., Springer Verlag (2000) 309–322
12. Girard, A., Pappas, G.J.: Approximate bisimulation for constrained linear systems. to appear in the Proceedings of the IEEE Conf. Decision and Control (2005)
13. Desharnais, J., Gupta, V., Jagadeesan, R., Panangaden, P.: Metrics for labelled Markov processes. Theoretical Computer Science **318**(3) (2004) 323–354
14. van Breugel, F., Worrell, J.: An algorithm for quantitative verification of probabilistic transition systems. In: Proc. of CONCUR, Aalborg, Springer-Verlag (2001) 336–350
15. Julius, A.A., Girard, A., Pappas, G.J.: Approximate bisimulation for a class of stochastic hybrid systems. submitted to the American Control Conference 2006 (2005)
16. Prajna, S., Jadbabaie, A., Pappas, G.J.: Stochastic safety verification using barrier certificates. In: Proc. 43rd IEEE Conference on Decision and Control, Bahamas, IEEE (2004)
17. Prajna, S., Papachristodoulou, A., Seiler, P., Parillo, P.A.: SOSTOOLS and its control application. In: Positive polynomials in control, Springer - Verlag (2005)
18. Neogi, N.A.: Dynamic partitioning of large discrete event biological systems for hybrid simulation and analysis. [21] 463 – 476
19. Hespanha, J.P.: Polynomial stochastic hybrid systems. [22] 322–338
20. Löfberg, J.: (http://control.ee.ethz.ch/∼joloef/yalmip.php)
21. Alur, R., Pappas, G.J., eds.: Hybrid systems: computation and control. Volume 2993 of Lecture Notes in Computer Science., Springer Verlag (2004)
22. Morari, M., Thiele, L., eds.: Hybrid systems: computation and control. Volume 3414 of Lecture Notes in Computer Science., Springer Verlag (2005)