Approximate bisimulation for a class of stochastic hybrid systems

(Invited Paper)

Agung Julius, Antoine Girard, George Pappas

***Abstract*— We develop a notion of approximate bisimulation for a class of stochastic hybrid systems, namely, the jump linear stochastic systems (JLSS). The idea is based on the construction of the so called stochastic bisimulation function. With this function, we can quantify the distance between two jump linear stochastic systems. The function is then used to quantify the distance between a given JLSS and its abstraction, and hence quantify the quality of the abstraction. We show that this idea can be applied to simplify safety verification for JLSS. We also show that in the absence of input, and by assuming that the stochastic bisimulation function is of quadratic form, we can pose the construction as a tractable linear matrix inequality problem.**

I. Introduction

Abstraction of dynamical systems has been an active research topic [25], [24], [30], [32], [31]. The main idea of abstracting dynamical systems is, given a dynamical system, we construct a relatively simpler system that is, in some sense, equivalent to the original. The meaning of simpler is rather vague, but usually it is related to some computation (for example, reachability computation) that we want to perform on the system. Thus simpler system usually means a system allows us to spend less computing effort. The equivalence between the original system and its abstraction guarantees that the result of the computation performed on the abstraction can be carried over into the original system.

As systems that we deal with get more complex, abstraction is clearly a necessity, so that the available computation tools will be able to cope with the ever increasing complexity. The need to obtain an abstraction of complex systems leads researchers to develop abstraction theories that allow for more *aggressive* abstraction. One of the ideas is to relax the requirement that the abstraction is equivalent to the original system, and replace with that the abstraction is only *approximately* equal to the original system (see, e.g. [34], [6], [13]). The key ingredient to these theories is a metrics that can quantify the distance between the system and its abstraction, and hence the quality of the abstraction.

The idea of abstraction of stochastic systems using some notion of systems equivalence has also been pursued by researchers for the same motivation. See, for example, [14], [21], [3], [28]. Similarly, there is also an approximate abstraction theory for stochastic systems. For example, [7], [8] discuss the idea of exact and approximate bisimulation for labelled Markov processes.

This research is partially supported by the Région Rhône-Alpes (Project CalCel) and the National Science Foundation Presidential Early CAREER (PECASE) Grant 0132716.

The authors are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 {agung,agirard,pappasg}@seas.upenn.edu

Following a series of previous work on approximate abstraction of dynamical systems [12], [11], [18], we extend the paradigm to handle a class of stochastic hybrid systems, namely, the jump linear stochastic systems (JLSS). Jump linear stochastic systems are widely applied, for example, in manufacturing systems, aircraft control, target tracking, robotics, and power systems [35].

The field of stochastic hybrid systems is a very active research area. The particular class of systems that we use in this paper (JLSS) is only one of the various modelling formalisms available [26]. For example, Hu *et al* [17] discuss a general type of stochastic hybrid systems, where the dynamics within each location (discrete state) is governed by diffusion stochastic differential equations [23], and switches happen when some invariant condition is violated. Piecewise deterministic Markov processes (PDP) [5], [15] is another modelling formalism for stochastic hybrid systems. In this framework, the dynamics within each location is non-stochastic. Stochasticity comes into the picture because switches happen when either a Poisson process generates a point or an invariant condition is violated. In that case, a jump in the state occur according to a certain probabilistic distribution. Continuous time Markov chains [4] can be thought of as a special class of PDP. This framework is extended by including possibility that such processes can communicate through labelled events in [29]. There are also other formalisms such that the polynomial stochastic hybrid systems [16], discrete stochastic hybrid automata [2], switched diffusion processes [10], etc. Research in the field of stochastic hybrid systems has been directed towards various topics, such as, stability analysis [1], control [4], [19], [2], model reduction [35], system identification [33], etc. This list is by no mean exhaustive.

In this paper, our aim is to develop a theory of approximate bisimulation of JLSS, that can be used, for example in safety and reachability analysis of the system. Moreover, we want that the theory allows for tractable computation.

There has been some research in the area of approximate bisimulation of stochastic systems. However, our approach differs from the others, in that we do not partition the state space and use a metric to define distance between probabilistic distributions, for example, as in [7], [8]. The approximate abstraction that we study does not necessarily correspond to an equivalence relation in the state space. There has been also work on H_∞ model reduction of Markovian jump linear systems, for example in [35]. However, as we shall see in the following section, for safety and reachability analysis, our approach that is based on the L_∞ distance between the trajectories is more suitable.

II. JUMP LINEAR STOCHASTIC SYSTEMS

A *jump linear stochastic system* (JLSS) can be modeled as a stochastic system that satisfies the following stochastic differential equation.

$$dx_t = Ax_t\ dt + Bu(t)\ dt + F\ dw_t + Rx_t\ dp_t, \quad (1)$$
$$u(t) \in \mathcal{U}, \forall t \in \mathbb{R}_+, \quad (2)$$
$$y_t = Cx_t. \quad (3)$$

Here, y_t is the observation of the process x_t, the signal $u(t)$ is an input signal taking value in a compact set $\mathcal{U}$, the process w_t is a standard Brownian motion, while p_t is a Poisson process with a constant rate λ.

Notation. We denote the class of locally integrable function taking value in the compact set $\mathcal{U}$ as $\mathfrak{U}$.

A Poisson process with a constant rate λ is a piecewise constant, monotonously nondecreasing process,

$$p_t = \begin{cases} 0, & 0 \leq t \leq t_1, \\ n, & t_n < t \leq t_{n+1}, n \in \mathbb{Z}_+, \end{cases} \quad (4)$$

where $t_1, (t_2 - t_1), (t_3 - t_2), \cdots, (t_{n+1} - t_n), \cdots$ are independent random variables with exponential distribution:

$$P\{t_{n+1} - t_n > \alpha\} = e^{-\lambda\alpha}, n \in \mathbb{Z}_+. \quad (5)$$

The random time instants t_n are called *event times*. Poisson processes are commonly used in modelling stochastic arrival processes (see [5]).

The JLSS described in (1) then can be interpreted as follows. In between the event times generated by the Poisson process p_t, the process behaves like a linear stochastic system

$$dx_t = Ax_t\ dt + Bu(t)\ dt + F\ dw_t, \quad (6)$$
$$y_t = Cx_t. \quad (7)$$

At the event time t_n, the process undergoes a jump

$$\lim_{t \downarrow t_n} x_t = (I + R)x_{t_n}. \quad (8)$$

Notice that with R we can parametrize any linear jump. Hence the name, jump linear stochastic system.

We use a Poisson process to model the occurrences of an event. The effect of an occurrence of the event is expressed as the linear jump (8). However, it is possible that we need to include more than just one kind of event in the model. Thus, generally the model (1) can be slightly extended to be:

$$dx_t = Ax_t\ dt + Bu(t)\ dt + F\ dw_t + \sum_{i=1}^{N} R_i x_t\ dp_t^i. \quad (9)$$

That is, we model N kinds of event whose occurrences are independent one from the others. The matrices R_i, $i = 1, \ldots, N$, parametrize the jump associated with event i. We also assume that the Poisson process p_t^i has the rate of λ_i.

III. ABSTRACTION OF JLSS

Given a JLSS of the following form.

$$dx_t = Ax_t\ dt + Bu(t)\ dt + F\ dw_t + \sum_{i=1}^{N} R_i x_t\ dp_t^i, \quad (10)$$
$$u \in \mathfrak{U}, \quad (11)$$
$$y_t = Cx_t. \quad (12)$$

There are a number of simplifications or model reductions that we can perform on this system. For example, we can

1) Truncate some of the dimensions of the state, to create a JLSS with smaller state space,
2) Neglect the term corresponding to the Brownian motion,
3) Neglect the occurrence of some of the events,

or a combination of them. Simplification of type 1 and 2 above have been discussed in [18] for linear stochastic systems. That is, the same kind of systems as we are discussing here, only without the Poisson processes.

The approach that we take is similar to that of the previous work [13], [12], [11], [18]. Namely, we want to compute a bisimulation function between two given JLSS. Simply said, the bisimulation function is an instrument that measures the distance between the two processes. It also guarantees that both processes satisfy the same reachability and safety properties within a certain bound.

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

$$S_i : \begin{cases} dx_{it} = A_i x_{it} dt + B_i u_i(t) dt + F_i dw_t + \sum_{j=1}^{N} R_{ij} x_t dp_t^j, \\ u_i \in \mathfrak{U}_i, \\ y_{it} = C_i x_{it}. \end{cases} \quad (13)$$

We define the following composite process

$$x_t := \begin{bmatrix} x_{1t} \\ x_{2t} \end{bmatrix}, y_t := y_{1t} - y_{2t},$$
$$u(t) := \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}, A := \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix},$$
$$B := \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix}, F := \begin{bmatrix} F_1 \\ F_2 \end{bmatrix},$$
$$R_j := \begin{bmatrix} R_{1j} \\ R_{2j} \end{bmatrix}, C := \begin{bmatrix} C_1 & -C_2 \end{bmatrix}.$$

Hence we have the following process:

$$S : \begin{cases} dx_t = Ax_t dt + Bu(t) dt + F dw_t + \sum_{j=1}^{N} R_j x_t dp_t^j, \\ u \in \mathfrak{U}_1 \times \mathfrak{U}_2, \\ y_t = Cx_t. \end{cases} \quad (14)$$

Observe that if $u(t)$ and the distribution of the initial state x_0 are known, then x_t is a stochastic process.

Definition 1: A function $\phi(x)$ is called a **stochastic bisimulation function** if
(i) $\phi(x) \geq \|Cx\|^2, \forall x$, and
(iia) for any $u_1 \in \mathfrak{U}_1$ there exists a $u_2 \in \mathfrak{U}_2$ such that the process $\phi(x_t)$ is a supermartingale[1] for any distribution of

[1] i.e. its expectation is monotonously nonincreasing.

the initial state, and
(iib) for any $u_2 \in \mathfrak{U}_2$ there exists a $u_1 \in \mathfrak{U}_1$ such that the process $\phi(x_t)$ is a supermartingale for any distribution of the initial state.

Remark 2: Bisimulation for nonstochastic systems is typically seen as a two-player tracking game [22], [12], [11]. For stochastic systems, one can think of the stochasticity as a third player in the game. In this point of view, there are multiple interpretations about how the game is played. That is, when the third player (stochasticity) makes its decision. There are three possibilities, namely before the other two players, in between, or after them. The definition of bisimulation that we adopt in this paper is based on the interpretation that the third player makes its decision after the other two players. Our choice is mainly based on computation consideration, although we can see later that this choice also leads to a sensible relationship between bisimulation and safety verification. That being said, we make no claim that this choice is better than the other two interpretations. Indeed, to explore the relations between all three interpretations is an interesting research direction.

Remark 3: A function $\phi(x)$ that satisfies conditions (i) and (iia) of Definition 1 is called a stochastic simulation function of S_1 by S_2. Similarly, if it satisfies conditions (i) and (iib) of Definition 1 is called a stochastic simulation function of S_2 by S_1.

Definition 4: A stochastic bisimulation function $\phi(x)$ is **trivial** if its value is $+\infty$ everywhere.

Obviously, we are interested in nontrivial bisimulation functions. The following theorem describes the relation between the bisimulation function and the difference between the observations of S_1 and S_2.

Theorem 5: Given a system described by (14), and $\phi(\cdot)$ a bisimulation function. For any $u_1 \in \mathfrak{U}_1$ there exists a $u_2 \in \mathfrak{U}_2$ such that the following relation holds.

$$P\left\{ \sup_{0\le t<\infty} \|y_t\|^2 \ge \delta \,\middle|\, x_0 \right\} \le \frac{\phi(x_0)}{\delta}. \tag{15}$$

Conversely, for any $u_2 \in \mathfrak{U}_2$ there exists a $u_1 \in \mathfrak{U}_1$ such that (15) holds.

Proof: Following Definition 1, for any $u_1 \in \mathfrak{U}_1$ there exists a $u_2 \in \mathfrak{U}_2$ such that $\phi(x_t)$ is a supermartingale. Since $\phi(x_t)$ is a nonnegative supermartingale, we have the following result [27].

$$P\left\{ \sup_{0\le t<\infty} \phi(x_t) \ge \delta \,\middle|\, x_0 \right\} \le \frac{\phi(x_0)}{\delta}. \tag{16}$$

Moreover, since $\phi(x) \ge \|Cx\|^2$ by construction, we also have that

$$P\left\{ \sup_{0\le t<\infty} \|y_t\|^2 \ge \delta \,\middle|\, x_0 \right\} \le P\left\{ \sup_{0\le t<\infty} \phi(x_t) \ge \delta \,\middle|\, x_0 \right\}. \tag{17}$$

Proving the converse statement is analogous. ■

Theorem 5 tells us that the bisimulation function of JLSS can be used to quantify the distance between the two systems S_1 and S_2. For a better illustration, consider the following corollary.

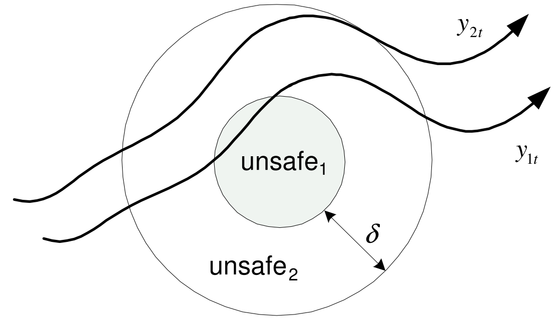

Fig. 1. Safety analysis with approximate bisimulation.

Corollary 6: Given two systems S_1 and S_2 as in (13), and suppose that $\phi(\cdot)$ is a bisimulation function of the composite system. We have the following relations. For any $u_1 \in \mathfrak{U}_1$ there exists a $u_2 \in \mathfrak{U}_2$ such that

$$P\left\{ \sup_{0\le t<\infty} \|y_{1t} - y_{2t}\|^2 \ge 10 \cdot \phi(x_0) \,\middle|\, x_0 \right\} \le 0.1, \tag{18a}$$
$$P\left\{ \sup_{0\le t<\infty} \|y_{1t} - y_{2t}\|^2 \ge 20 \cdot \phi(x_0) \,\middle|\, x_0 \right\} \le 0.05. \tag{18b}$$

Conversely, for any $u_2 \in \mathfrak{U}_2$ there exists a $u_1 \in \mathfrak{U}_1$ such that (18) holds.

Thus, we can say that given the systems can simulate each other (by selecting appropriate input), such that the supremum of the difference in the observations will not exceed $\sqrt{10 \cdot \phi(x_0)}$ and $\sqrt{20 \cdot \phi(x_0)}$ with 90% and 95% confidence respectively.

The idea of approximate bisimulation of JLSS can be used as a tool for abstraction of JLSS that can be used in conjunction with stochastic safety analysis. Given a complex system S_1 and its simpler abstraction S_2. Suppose that $\phi(\cdot)$ is a stochastic bisimulation function between the two systems, and that the initial condition of the composite system is $x_0 = (x_{10}, x_{20})$. Given the unsafe set for the original system S_1, $\texttt{unsafe}_1$, we can construct another set $\texttt{unsafe}_2$, which is the δ neighborhood of $\texttt{unsafe}_1$ for some $\delta > 0$. That is,

$$\texttt{unsafe}_2 = \{ y \mid \exists y' \in \texttt{unsafe}_1, \|y - y'\| \le \delta \}. \tag{19}$$

See Figure 1 for an illustration.

If we define the events $\mathbf{unsafe}_1(v)$ and $\mathbf{unsafe}_2(v)$ as functions of the external input signal, for $i = 1, 2$,

$$\mathbf{unsafe}_i(v) := \{ \exists t \ge 0 \text{ s.t. } y_{it} \in \texttt{unsafe}_i \mid u_i = v \in \mathfrak{U}_i \}, \tag{20}$$

then we have the following theorem holds.

Theorem 7:

$$\sup_{u_1 \in \mathfrak{U}_1} P\{\mathbf{unsafe}_1(u_1)\} \le \sup_{u_2 \in \mathfrak{U}_2} P\{\mathbf{unsafe}_2(u_2)\} + \frac{\phi(x_0)}{\delta^2}. \tag{21}$$

Proof: We start with the following relation. Take any $u_1 \in \mathfrak{U}_1$, let $\tilde{u}_2 \in \mathfrak{U}_2$ be such that $\phi(x_t)$ is a supermartingale (see Definition 1). We then have the following relation.

$$P\{\mathbf{unsafe}_1(u_1)\} = P\{\mathbf{unsafe}_1(u_1) \cap \mathbf{unsafe}_2(\tilde{u}_2)\} + P\{\mathbf{unsafe}_1(u_1) \cap \mathbf{unsafe}_2^C(\tilde{u}_2)\}, \tag{22}$$

where $\mathbf{unsafe}_2^C(\tilde{u}_2)$ denotes the complement of the event $\mathbf{unsafe}_2(\tilde{u}_2)$. Now, notice that

$$\begin{aligned} P\{\mathbf{unsafe}_1(u_1) \cap \mathbf{unsafe}_2(\tilde{u}_2)\} &\le P\{\mathbf{unsafe}_2(\tilde{u}_2)\}, \\ &\le \sup_{u_2 \in \mathfrak{U}_2} P\{\mathbf{unsafe}_2(u_2)\}. \end{aligned} \tag{23}$$

and because of Theorem 5,

$$P\{\mathbf{unsafe}_1(u_1) \cap \mathbf{unsafe}_2^C(\tilde{u}_2)\} \le \frac{\phi(x_0)}{\delta^2}. \tag{24}$$

Thus we have that for any $u_1 \in \mathfrak{U}_1$,

$$P\{\mathbf{unsafe}_1(u_1)\} \le \sup_{u_2 \in \mathfrak{U}_2} P\{\mathbf{unsafe}_2(u_2)\} + \frac{\phi(x_0)}{\delta^2}.$$

■

The term $\sup_{u_1 \in \mathfrak{U}_1} P\{\mathbf{unsafe}_1(u_1)\}$ gives us the risk of unsafety of S_1 in the worst scenario. That is, we choose the input so as to maximize the risk. Similarly, the term $\sup_{u_2 \in \mathfrak{U}_2} P\{\mathbf{unsafe}_2(u_2)\}$ gives us the risk of unsafety of S_2 in the worst scenario. Theorem 7 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.

Remark 8: Notice that the symmetric definition of stochastic bisimulation functions implies that not only S_2 can be used to approximate S_1, but also the converse is true (see Theorem 5). If we only want to have one way approximation, then a stochastic simulation function suffices.

IV. Construction of the Bisimulation Function

In this paper, we assume that a stochastic bisimulation function can be constructed as a quadratic function of the (composite) state, that is, a function of the following form.

$$\phi(x) = x^T M x, \tag{25}$$

where M is symmetric nonnegative definite.

Recall that the composite x_t satisfies (14). The stochastic process $\phi_t := \phi(x_t)$ then satisfies the following stochastic differential equation.

$$\begin{aligned} d\phi_t &= \frac{\partial \phi}{\partial x} dx_t + \frac{1}{2} dx_t^T \frac{\partial^2 \phi}{\partial x^2} dx_t, \\ &= 2x_t^T M \left(Ax_t dt + Bu(t)dt + F dw_t + \sum_{j=1}^N R_j x_t dp_t^j \right) \\ &\quad + \operatorname{trace}\left(F^T M F\right) \; dt \\ &\quad + \sum_{i,j \in \{1,2,\cdots,N\}} x_t^T R_i^T M R_j x_t \; dp_t^i dp_t^j. \end{aligned} \tag{26}$$

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,

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

The expectation of ϕ_t then satisfies the following equation.

$$\begin{aligned} \frac{dE[\phi_t]}{dt} &= 2E\left[x^T \left(MA + \sum_{j=1}^N \lambda_j \left(I + \frac{R_j}{2} \right)^T M R_j \right) x_t \right] \\ &\quad + 2E[x_t^T] M B u(t) + \operatorname{trace}\left(F^T M F\right). \end{aligned} \tag{27}$$

Denote

$$\begin{aligned} Q := M \left(A + \sum_{j=1}^N \lambda_j R_j \right) + \left(A + \sum_{j=1}^N \lambda_j R_j \right)^T M \\ + \sum_{j=1}^N \lambda_j R_j^T M R_j, \end{aligned}$$

then we have that

$$\frac{dE[\phi_t]}{dt} = E\left[x_t^T Q x_t\right] + 2E[x_t^T] M B u(t) + \operatorname{trace}\left(F^T M F\right). \tag{28}$$

Lemma 9: The function ϕ is a stochastic bisimulation function if and only if the following relations are satisfied.

$$M - C^T C \ge 0, \tag{29}$$

and for almost all $t \ge 0$,

$$\sup_{u_1 \in \mathfrak{U}_1} \inf_{u_2 \in \mathfrak{U}_2} E\left[x_t^T Q x_t\right] + 2E[x_t^T] M B u(t) + \operatorname{trace}\left(F^T M F\right) \le 0, \tag{30a}$$

$$\sup_{u_2 \in \mathfrak{U}_2} \inf_{u_1 \in \mathfrak{U}_1} E\left[x_t^T Q x_t\right] + 2E[x_t^T] M B u(t) + \operatorname{trace}\left(F^T M F\right) \le 0, \tag{30b}$$

for any distribution of the initial state x_0.

Proof: (if) Suppose that (29) is satisfied, then we know that condition (i) of Definition 1 is satisfied by ϕ. Moreover, if (30) is satisfied, then from (28) we know that ϕ_t is a supermartingale as required by conditions (iia) and (iib) of Definition 1.

(only if) We can see that (29) is clearly a necessary condition, otherwise condition (i) of Definition 1 will not be satisfied. The condition (30a) is also necessary, because otherwise we can select some input $u_1 \in \mathfrak{U}_1$ such that there is no $u_2 \in \mathfrak{U}_2$ that will make ϕ_t a supermartingale, as required by condition (iia) in Definition 1. Similarly, we can infer the necessity of (30b). ■

In the remaining of the paper, we shall impose the following assumption. The reason being that it will lead to tractable computation that can be used to construct the desired stochastic bisimulation function.

Assumption. Hereafter, we assume that the inputs are absent. That is,

$$B = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} = 0. \quad (31)$$

In this case, ϕ is a stochastic bisimulation function if and only if

$$M - C^T C \geq 0, \quad (32)$$

and for almost all $t \geq 0$,

$$E\left[x_t^T Q x_t\right] + \operatorname{trace}\left(F^T M F\right) \leq 0. \quad (33)$$

Lemma 10: Given two systems S_1 and S_2 as in (13) under the assumption (31),

$$\phi(x) := x^T M x, \quad (34)$$

where M is symmetric nonnegative definite is a bisimulation function if and only if

$$Q \leq 0, \quad (35)$$

$$M - C^T C \geq 0, \quad (36)$$

$$MF = 0. \quad (37)$$

Proof: (only if) Suppose that any of (35)-(37) is not true. We shall show that $\phi(x)$ is not a bisimulation function. Suppose (5) is not true, then there exists an $x_0 \in \mathcal{X}_1 \times \mathcal{X}_2$ such that

$$x_0^T Q x_0 > 0. \quad (38)$$

Therefore, if we assume that the initial state is x_0, $\phi(x_t)$ cannot be a supermartingale. If (7) is not true, then $\phi(x)$ cannot be a bisimulation function (see Definition 1, point (i)). If (37) is not true, then because of nonnegativity of M, we have that

$$\operatorname{trace}\left(F^T M F\right) > 0. \quad (39)$$

Therefore, if we assume that the initial state is 0, $\phi(x_t)$ cannot be a supermartingale.

(if) This part of the proof is trivial. ∎

It can be verified that the problem of constructing a matrix M that satisfies (35)-(37) is a linear matrix inequality problem (LMI) that can be solved using some available tools, such as YALMIP [20].

Remark 11: If we think of the stochastic bisimulation function (34) as a stochastic Lyapunov function, then (35) - (37) guarantees that y_t converges to 0 in probability [9].

A more general construction of the bisimulation function can be achieved by using the so called *truncated quadratic function*. That is, we consider bisimulation functions of the following form.

$$\phi(x) = \begin{cases} x^T M x, & x^T M x > \alpha, \\ \alpha, & x^T M x \leq \alpha, \end{cases} \quad (40)$$

for some value $\alpha \geq 0$. The construction is thus parameterized by M and α. This construction is potentially more powerful than considering only quadratic functions alone. Suppose that there exists a quadratic bisimulation function, the zero level (the kernel) of the function gives us the composite states, from which the output is always zero with probability one (see Theorem 5). This implies perfect tracking, which might be too restrictive. The implication does not hold for truncated quadratic bisimulation functions, since the zero level might be an empty set. Thus, potentially, we can find a stochastic bisimulation function, even if perfect tracking is not possible.

Consider the following problem.

Problem 12: Given a matrices A,F,C and $(R_j)_{j=1,2,\cdots,N}$. Construct a truncated quadratic function $\phi(x)$, as in (40), that satisfies

$$\phi(x) \geq \|Cx\|^2, \quad (41a)$$

$$Q \leq 0, \quad (41b)$$

$$x^T Q x = 0 \text{ only if } x^T M x = 0, \quad (41c)$$

$$x^T Q x + \operatorname{trace}(F^T M F) \leq 0, \text{ if } x^T M x \geq \alpha. \quad (41d)$$

The motivation behind this problem is that if we can construct such a truncated function $\phi(x)$, then we can show that $\phi(x_t)$ is a supermartingale.

Remark 13: Notice that (41b) and (41c) imply that

$$\ker Q = \ker M \bigcap_{1 \leq j \leq N} R_j^{-1}(\ker M), \quad (42)$$

where the set theoretic notation $R_j^{-1}(\ker M)$ means $\{x \mid R_j x \in \ker M\}$.

Let us first discuss the solution to Problem 12. The solution to this problem can be constructed in two steps, namely:

1) Construct a quadratic function $\tilde{\phi}(x) := x^T M x$ that satisfies

$$M - C^T C \geq 0, \quad (43)$$

$$Q \leq 0, \quad (44)$$

$$x^T Q x = 0 \text{ only if } x^T M x = 0. \quad (45)$$

The procedure for this construction can be posed as an LMI problem, and if a solution exists, it can be found using some available LMI tools, such as YALMIP [20].

2) Determine the treshold α for the quadratic function designed in the previous step, so that $\phi(x) := \max(\alpha, \tilde{\phi}(x))$ satisfies (41d). Notice that if α satisfies the requirement, any $\alpha' > \alpha$ will also satisfy the requirement. So we are interested in getting as small α as possible.

Let us now discuss the second step. The smallest α that satisfies (41d) can be expressed as the following optimization problem.

$$\alpha = \max_{x^T Q x + \operatorname{trace}(F^T M F) \geq 0} x^T M x. \quad (46)$$

Notice that since M is nonnegative definite, the optimal solution lies on the boundary of the feasible set. Hence,

$$\alpha = \max_{x^T Q x + \operatorname{trace}(F^T M F) = 0} x^T M x. \quad (47)$$

We can solve this problem, for example, by using Lagrange's multiplier method. The optimality condition is that if the optimal value is attained at $x = \bar{x}$, then

$$Q\bar{x} = \lambda M \bar{x}, \quad (48)$$

for some real number λ. If M is invertible, then by rearranging the equation, we get

$$M^{-1}Q\bar{x} = \lambda\bar{x}. \tag{49}$$

Thus, $\bar{x}$ must be an eigenvector of $M^{-1}Q$. Since $\ker Q \subset \ker M$, Q is a symmetric negative definite matrix. Therefore $M^{-1}Q$ has real eigenvalues. We have reduced the optimization problem (47) to an optimization problem with a finite countable set of feasible points, which is easy to solve.

If M is not invertible, we project the optimization problem to $\operatorname{im} M$, which is the image of M. We denote the dimension of $\operatorname{im} M$ as $\tilde{m}$. Let

$$\Gamma := \begin{bmatrix} \gamma_1 & \gamma_2 & \cdots & \gamma_{\tilde{m}} \end{bmatrix}$$

be such that the vectors $(\gamma_i)_{1\le i\le \tilde{m}}$ span an orthonormal basis for $\operatorname{im} M$. We can find a positive definite matrix $\tilde{M} \in \mathbb{R}^{\tilde{m}\times\tilde{m}}$ such that

$$x^T M x = x^T \Gamma \tilde{M} \Gamma^T x. \tag{50}$$

Thus, we have that

$$\alpha = \max_{v^T\Gamma Q\Gamma^T v + \operatorname{trace}(F^T M F)=0} v^T \tilde{M} v, \ v \in \mathbb{R}^{\tilde{m}}. \tag{51}$$

Notice that (51) has the same form as (47), and $\tilde{M}$ is invertible, so we can use the procedure discussed above to find the optimal α.

Now that we have shown the construction of the truncated quadratic function $\phi(x)$ as required by Problem 12, we are going to prove that $\phi(x_t)$ is a supermartingale.

Theorem 14: Given a truncated quadratic function

$$\phi(x) = \max(\alpha, x^T M x) \tag{52}$$

as required by Problem 12. The stochastic process $\phi(x_t)$ is a supermartingale.

Proof: The process $\phi(x_t)$ satisfies the following stochastic differential equation

$$d\phi(x_t) = \begin{cases} 0, & x_t^T M x_t < \alpha, \\ 2x_t^T M A x_t \ dt + 2x_t^T M F \ dw_t & \\ +2x_t^T M \sum_{j=1}^N R_j x_t \ dp_t^j & \\ + \operatorname{trace}\left(F^T M F\right) \ dt & \\ + \sum_{j=1}^N \lambda_j x_t^T R_j^T M R_j x_t \ dt, & x_t^T M x_t \ge \alpha. \end{cases} \tag{53}$$

Since the expectation of the right hand side is nonpositive by construction, it follows that $\phi(x_t)$ is a supermartingale. ■

Therefore, using Definition 1, we can establish that $\phi(x)$ is a stochastic bisimulation function.

V. Simulation results

In this section we present some simulation results of approximate bisimulation. The original system is a JLSS with sixth order linear dynamics. The system S is given as:

$$S : \begin{cases} dx_t = Ax_t \ dt + F \ dw_t + Rx_t \ dp_t, \\ y_t = Cx_t, \end{cases} \tag{54}$$

where

$$A = \begin{bmatrix} -0.1 & -1 & 0 & 0 & 0 & 0 \\ 1 & -0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.2 & -2 & 0 & 0 \\ 0 & 0 & 2 & -0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.25 \end{bmatrix},$$

$$F = \begin{bmatrix} 0.74 & 0.07 & -0.62 & -0.27 & -0.86 & 0.65 \end{bmatrix}^T,$$

$$C = \begin{bmatrix} 0.84 & -1.03 & 1.07 & -0.88 & 0.5 & 0 \\ -0.60 & -1.35 & -0.26 & -0.27 & 0 & -0.5 \end{bmatrix},$$

$$R = 0.1I.$$

The rate of the Poisson process p_t is 0.5.

We construct three kinds of abstraction, as mentioned earlier in Section III, and for each case, compute a stochastic bisimulation function. We then simulate several realizations of the composite system for the first 500 seconds of the evolution and plot the realizations of the error.

A. Abstraction of the linear dynamics

We construct a JLSS with simpler linear dynamics. Namely, we remove the last two modes of the original linear dynamics and hence create a fourth order linear system. Thus, we compute the stochastic bisimulation function between S and S' where

$$S' : \begin{cases} dx'_t = A'x'_t \ dt + F' \ dw_t + R'x'_t \ dp_t, \\ y'_t = C'x'_t, \end{cases} \tag{55}$$

$$A' = \begin{bmatrix} -0.1 & -1 & 0 & 0 \\ 1 & -0.1 & 0 & 0 \\ 0 & 0 & -0.2 & -2 \\ 0 & 0 & 2 & -0.1 \end{bmatrix}, \ F' = \begin{bmatrix} 0.74 \\ 0.07 \\ -0.62 \\ -0.27 \end{bmatrix},$$

$$C' = \begin{bmatrix} 0.84 & -1.03 & 1.07 & -0.88 \\ -0.60 & -1.35 & -0.26 & -0.27 \end{bmatrix}, \ R' = 0.1I.$$

In the simulation, the initial state of the original system is chosen randomly.

Figure 2 and 3 show the simulation results. In Figure 2 we see a realization of the observed process, y_t and y'_t. In Figure 3 we see ten realizations of $(y_t - y'_t)$. The circle denotes the 90% confidence bound given by the computed stochastic bisimulation function (see Corollary 6).

B. Abstraction of the Poisson process

We construct an abstraction of S with zero R. That is, in the abstraction, we neglect the effect of the Poisson process. We therefore create another system $\tilde{S}$, where

$$\tilde{S} : \begin{cases} d\tilde{x}_t = A\tilde{x}_t \ dt + F \ dw_t, \\ \tilde{y}_t = C\tilde{x}_t. \end{cases}$$

In the simulation, the initial state of the original system is chosen randomly.

Figure 4 and 5 show the simulation results. In Figure 4 we see a realization of y_t and $\tilde{y}_t$. We can see clearly that y_t has jumps corresponding to the Poisson process and $\tilde{y}_t$ does not. In Figure 5, ten realizations of $(y_t - \tilde{y}_t)$ are plotted with the 90% confidence bound (see Corollary 6).

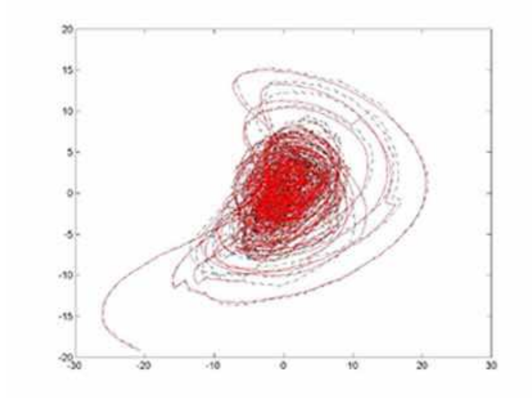

Fig. 2. One realization of y_t (dashed) and y'_t (solid).

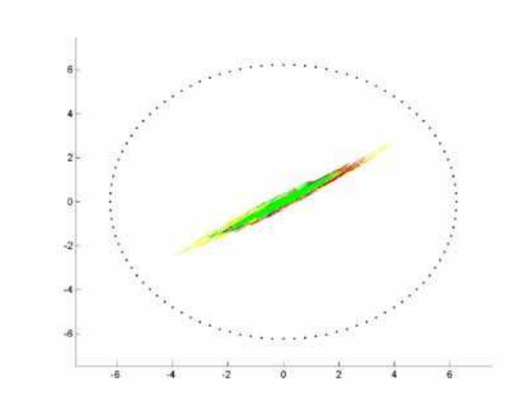

Fig. 3. Ten realizations of $(y_t - y'_t)$. The circle indicates the 90% confidence bound.

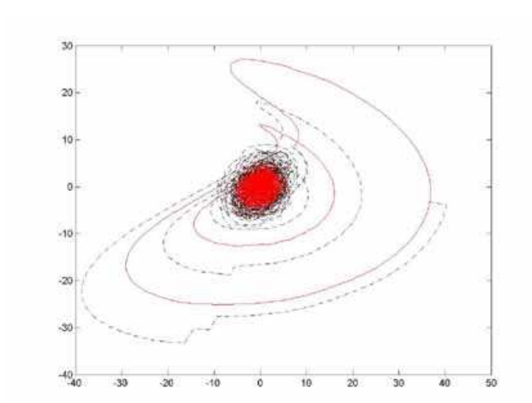

Fig. 4. One realization of y_t (dashed) and $\tilde{y}_t$ (solid).

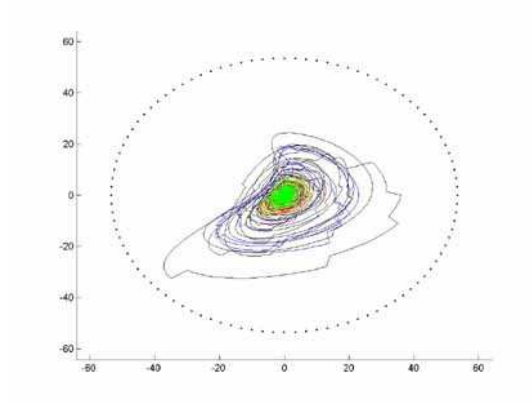

Fig. 5. Ten realizations of $(y_t - \tilde{y}_t)$. The circle indicates the 90% confidence bound.

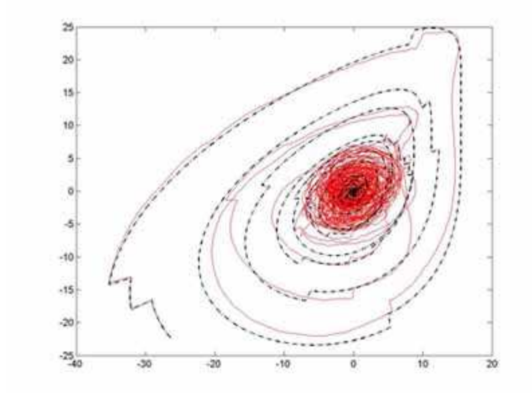

Fig. 6. One realization of y_t (dashed) and $\hat{y}$ (solid).

C. Abstraction of the Brownian motion

We construct an abstraction of S by neglecting the Brownian motion. We therefore create another system $\hat{S}$, where

$$\hat{S} : \begin{cases} d\hat{x}_t = A\hat{x}_t \; dt + R\hat{x}_t \; dp_t, \\ \hat{y}_t = C\hat{x}_t. \end{cases}$$

In the simulation, the initial state of the original system is chosen randomly.

In Figure 6 we see a realization of y_t and $\hat{y}_t$. In Figure 7, ten realizations of $(y_t - \hat{y}_t)$ are shown with the 90% confidence bound (see Corollary 6).

VI. Concluding remarks

In this paper we discuss the idea of approximate bisimulation for a class of stochastic systems, namely, the jump linear stochastic systems. abstraction of jump linear stochastic systems (JLSS). With approximate bisimulation, we can quantify the quality of approximation of the system. We use the tool of stochastic bisimulation function to establish approximate bisimulation between two JLSS The concept of bisimulation function was introduced in [13], [12], [11].

The idea of approximate bisimulation using stochastic bisimulation function amounts to finding a function that satisfies some certain properties (see Definition 1). In the

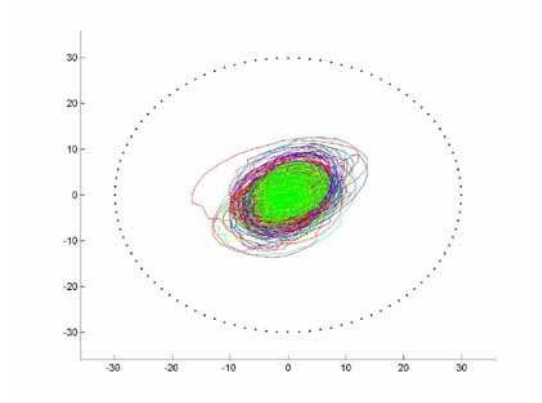

Fig. 7. Ten realizations of $(y_t - \hat{y}_t)$. The circle indicates the 90% confidence bound.

approach discussed in this paper, we restrict our attention to functions in the class of truncated quadratic functions, which can be thought of as a generalization of quadratic functions [11]. For the computation, in this paper, we restrict our attention to systems with no input. In this case, it is shown that the construction of the stochastic bisimulation function can be formulated as a Linear Matrix Inequality (LMI) problem, which can be solved using available LMI tools. However, we have not (yet) formulated necessary and sufficient conditions for the existence for such stochastic bisimulation function. From the theoretical point of view, we identify this problem as an interesting direction for further research. Another interesting research direction is to establish a computation algorithm that can handle systems with nondeterminism (i.e. the presence of inputs).

REFERENCES

[1] A. Abate, L. Shi, S. Simic, and S. Sastry. A stability criterion for stochastic hybrid systems. In *Proc. of Mathematical Theory of Networks and Systems*, Leuven, July 2004.

[2] A. Bemporad and S. Di Cairano. Optimal control of discrete hybrid stochastic automata. In M. Morari and L. Thiele, editors, *Hybrid Systems: Computation and Control*, volume 3414 of *Lecture Notes in Computer Science*, pages 151–167. Springer Verlag, 2005.

[3] M. L. Bujorianu, J. Lygeros, and M. C. Bujorianu. Bisimulation for general stochastic hybrid systems. In M. Morari and L. Thiele, editors, *Hybrid Systems: Computation and Control*, volume 3414 of *Lecture Notes in Computer Science*, pages 198–214. Springer Verlag, 2005.

[4] O. L. V Costa, J. B. R. do Val, and J. C. Geromel. Continuous-time state-feedback H_2-control of markovian jump linear systems via convex analysis. *Automatica*, 35(2):259–268, 1999.

[5] M. H. A. Davis. *Markov models and optimization*. Chapman and Hall, London, 1993.

[6] L. de Alfaro, M. Faella, and M. Stoelinga. Linear and branching system metrics. UCSC-CRL-05-01, School of Engineering, University of California at Santa Cruz, 2005.

[7] J. Desharnais, A. Edalat, and P. Panangaden. Bisimulation for labelled Markov processes. *Information and Computation*, 179(2):163–193, 2002.

[8] J. Desharnais, V. Gupta, R. Jagadeesan, and P Panangaden. A metric for labelled Markov processes. *Theoretical Computer Science*, 318(3):323–354, 2004.

[9] P Florchinger. Lyapunov-like techniques for stochastic stability. *SIAM Journal on Control and Optimization*, 33:1151–1169, 1995.

[10] M. K. Ghosh, A. Arapostathis, and S. Marcus. Ergodic control of switching diffusions. *SIAM Journal on Control and Optimization*, 35(6):1952–1988, 1997.

[11] A. Girard and G. J. Pappas. Approximate (bi)-simulations for linear systems with constrained inputs. manuscript, 2005.

[12] A. Girard and G. J. Pappas. Approximate bisimulation for constrained linear systems. to appear in the Proceedings of the IEEE Conf. Decision and Control, 2005.

[13] A. Girard and G. J. Pappas. Approximation metrics for discrete and continuous systems. MS-CIS-05-10, Dept. Computer and Information Science, University of Pennsylvania, Philadelphia, USA, May 2005.

[14] Holger Hermanns. *Interactive Markov Chains*, volume 2428 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 2002.

[15] J. P. Hespanha. Stochastic hybrid systems: applications to communication networks. In R. Alur and G. Pappas, editors, *Hybrid Systems: Computation and Control*, volume 2993 of *Lecture Notes in Computer Science*, pages 387–401. Springer Verlag, 2004.

[16] J. P. Hespanha. Polynomial stochastic hybrid systems. In M. Morari and L. Thiele, editors, *Hybrid Systems: Computation and Control*, volume 3414 of *Lecture Notes in Computer Science*, pages 322–338. Springer Verlag, 2005.

[17] J. Hu, J. Lygeros, and S. Sastry. Towards a theory of stochastic hybrid systems. In N. Lynch and B. H. Krogh, editors, *Hybrid Systems: Computation and Control*, volume 1790 of *Lecture Notes in Computer Science*, pages 160–173. Springer Verlag, 2000.

[18] A. A. Julius, A. Girard, and G. J. Pappas. Approximation of stochastic linear systems using approximate bisimulation. manuscript, July 2005.

[19] X. D. Koutsoukos. Optimal control of stochastic hybrid systems based on locally consistent Markov decision processes. *International Journal of Hybrid Systems*, 4:301–318, 2004.

[20] J. Löfberg. http://control.ee.ethz.ch/ joloef/yalmip.php.

[21] N. López and M. Núñez. An overview of probabilistic process algebras and their equivalences. In C. Baier, B. R. Haverkort, and H. Hermanns, editors, *Validation of Stochastic Systems: A Guide to Current Research*, volume 2925 of *Lecture Notes in Computer Science*, pages 89–123. Springer, 2004.

[22] R. Milner. *Communication and concurrency*. Prentice Hall International, 1989.

[23] B. Oksendal. *Stochastic differential equations: an introduction with applications*. Springer-Verlag, Berlin, 2000.

[24] G. J. Pappas. Bisimilar linear systems. *Automatica*, 39:2035–2047, December 2003.

[25] G. J. Pappas and S. Simic. Consistent abstractions of affine control systems. *IEEE Transactions on Automatic Control*, 47:745–756, May 2002.

[26] G. Pola, M. Bujorianu, J. Lygeros, and M. D. Benedetto. Stochastic hybrid models: an overview. In *Proc. IFAC Conf. Analysis and Design of Hybrid Systems*, St. Malo, 2003. IFAC.

[27] S. Prajna, A. Jadbabaie, and G. J. Pappas. Stochastic safety verification using barrier certificates. In *Proc. 43rd IEEE Conference on Decision and Control*, Bahamas, 2004. IEEE.

[28] S. Strubbe and A. J. van der Schaft. Bisimulation for communicating piecewise deterministic Markov processes. In M. Morari and L. Thiele, editors, *Hybrid Systems: Computation and Control*, volume 3414 of *Lecture Notes in Computer Science*, pages 623–639. Springer Verlag, 2005.

[29] S. N. Strubbe, A. A. Julius, and A. J. van der Schaft. Communicating piecewise deterministic Markov processes. In *Proc. IFAC Conf. Analysis and Design of Hybrid Systems*, pages 349–354, St. Malo, 2003. IFAC.

[30] P. Tabuada and G. J. Pappas. Abstraction of hamiltonian control systems. *Automatica*, 39:2025–2033, December 2003.

[31] P. Tabuada, G. J. Pappas, and P. Lima. Compositional abstractions of hybrid control systems. *Discrete event dynamic systems*, 14(2):203–238, April 2005.

[32] A. J. van der Schaft. Equivalence of dynamical systems by bisimulation. *IEEE Trans. Automatic Control*, 49(12):2160–2172, December 2004.

[33] R. Vidal, A. Chiuso, and S. Soatto. Observability and idetifiability of jump linear systems. In *Proceedings 41st IEEE Conf. Decision and Control*, pages 3614–3619, Las Vegas, 2002. IEEE.

[34] M. Ying and M. Wirsing. Approximate bisimilarity. In T. Rus, editor, *AMAST 2000*, volume 1816 of *Lecture Notes in Computer Science*, pages 309–322. Springer Verlag, 2000.

[35] L. Zhang, B. Huang, and J. Lam. H_∞ model reduction of markovian jump linear systems. *Systems and control letters*, 50:103–118, 2003.