Automatica 112 (2020) 108634

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Moment analysis of stochastic hybrid systems using semidefinite programming☆

Khem Raj Ghusinga[a], Andrew Lamperski[b], Abhyudai Singh[a,*]

[a] Department of Electrical and Computer Engineering, University of Delaware, Newark, DE, USA
[b] Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA

ARTICLE INFO

Article history:
Received 31 January 2018
Received in revised form 20 August 2019
Accepted 4 October 2019
Available online xxxx

Keywords:
Stochastic systems
Convex programming
Jump process
Optimization
Polynomial models

ABSTRACT

This paper proposes a method based on semidefinite programming for estimating moments of stochastic hybrid systems (SHSs). The class of SHSs considered herein consists of a finite number of discrete states and a continuous state whose dynamics as well as the reset maps and transition intensities are polynomial in the continuous state. For these SHSs, the dynamics of moments evolve according to a system of linear ordinary differential equations. However, it is generally not possible to exactly solve the system since time evolution of a specific moment may depend upon moments of order higher than it. Our methodology recasts an SHS with multiple discrete modes to a single-mode SHS with algebraic constraints. We then find lower and upper bounds on a moment of interest via a semidefinite program that includes linear constraints obtained from moment dynamics and those arising from the recasting process, along with semidefinite constraints coming from the non-negativity of moment matrices. We illustrate the methodology via an example of SHS.

1. Introduction

A Stochastic Hybrid System (SHS) consists of a finite number of discrete states (or modes), stochastic dynamics of a continuous state, and a set of transitions which are specified by transition intensities with corresponding reset maps that define how the states change after each of these transitions (Hespanha, 2006; Teel, Subbaraman, & Sferlazza, 2014). SHSs are applicable to a wide range of phenomena (Hespanha, 2005; Li, Omotere, Qian, & Dougherty, 2017), however their formal analysis (e.g., computing probability density function) is typically challenging. Alternatively, computing or estimating moments of an SHS also provides important insights into its dynamics. It is well-known that for an SHS described via polynomials, the time evolution of its moments is governed by a system of linear ordinary differential equations (Hespanha, 2005). However, except for few special cases, e.g., see Soltani and Singh (2017), the time-evolution of a moment of certain order depends on moments of order higher than it. It is desirable to develop methods that provide approximate values of moments with provable guarantees.

Many methods to approximate moments of polynomial stochastic systems have been proposed (Kuehn, 2016; Socha, 2007). Most of these methods, however, provide point approximations to moments of interest without any guarantee on errors (Kuehn, 2016). We and others have addressed this issue by proposing a semidefinite programming based method for estimating moments of polynomial jump diffusion processes and their special cases (Ghusinga, Vargas-Garcia, Lamperski, & Singh, 2017; Kuntz, Ottobre, Stan, & Barahona, 2016; Lamperski, Ghusinga, & Singh, 2019; Sakurai & Hori, 2018). This method utilizes semidefinite inequalities that moments must satisfy and finds monotonic sequence of lower and upper bounds on a moment of interest. Here, we extend this method to SHSs, which encompass a large class of stochastic systems. We show that an SHS with multiple discrete modes can be transformed to one with single discrete mode and some algebraic constraints. This transformation makes the moment analysis of SHSs amenable to the semidefinite programming setup. Finally, we use an example to illustrate the proposed method.

Notation. We denote random variables in bold. Unless deemed necessary, we omit explicit time dependence of states/moments. $\mathbb{R}^n$ denotes the n-dimensional Euclidean space. $\mathbb{N}$ represents the set of non-negative integers. $\mathbb{E}(\boldsymbol{x})$ is the expectation of a random variable $\boldsymbol{x}$. $\mathbb{1}_{s_i}$ represents the N-dimensional unit vector with 1 in the ith position.

☆ AS is supported by National Science Foundation, USA ECCS-1711548. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Debasish Chatterjee under the direction of Editor Daniel Liberzon.

* Corresponding author.
E-mail addresses: khem@udel.edu (K.R. Ghusinga), alampers@umn.edu (A. Lamperski), absingh@udel.edu (A. Singh).

https://doi.org/10.1016/j.automatica.2019.108634
0005-1098/

2. Background on stochastic hybrid systems

An SHS combines continuous-time evolution with discrete transitions. Let t denote continuous time and let k count the discrete transitions. The state space consists of a continuous state $\boldsymbol{x}(t,k) \in \mathbb{R}^n$ and a discrete state $\boldsymbol{q}(t,k) \in Q = \{s_1, s_2, \dots, s_N\}$. The continuous state evolves as per a stochastic differential equation (SDE)

$$d\boldsymbol{x} = f(\boldsymbol{q},\boldsymbol{x})dt + g(\boldsymbol{q},\boldsymbol{x})d\boldsymbol{w}, \tag{1a}$$

where $f : Q \times \mathbb{R}^n \to \mathbb{R}^n$ and $g : Q \times \mathbb{R}^n \to \mathbb{R}^{n\times l}$ are respectively the drift and diffusion terms, and $\boldsymbol{w}$ is an l-dimensional Wiener process. The state $(\boldsymbol{q},\boldsymbol{x})$ can also change through $\mathcal{R}$ transitions/resets that are characterized by the transition intensities

$$\lambda_r(\boldsymbol{q},\boldsymbol{x}),\ \ \lambda_r : Q \times \mathbb{R}^n \to [0,\infty),\ \ r = 1,2,\dots,\mathcal{R}. \tag{1b}$$

The probability that the r^{th} transition takes place in an infinitesimal time interval $(t, t+dt]$ is given by $\lambda_r(\boldsymbol{q},\boldsymbol{x})dt$. Each transition has an associated reset map

$$(\boldsymbol{q},\boldsymbol{x}) \mapsto (\theta_r(\boldsymbol{q}), \phi_r(\boldsymbol{q},\boldsymbol{x}))\,,$$
$$\theta_r : Q \to Q, \phi_r : Q \times \mathbb{R}^n \to \mathbb{R}^n, \tag{1c}$$

that defines how the pre-transition discrete and continuous states map into the post-transition discrete and continuous states.

For completeness, we sketch how dynamics defined by (1) can be posed as a jump diffusion (Applebaum, 2009) by modifying a construction from Ghosh, Arapostathis, and Marcus (1997) and Yin and Zhu (2009). With slight abuse of notation, identify Q with the integers $\{1,2,\dots,N\}$ and let $\Delta_r(\boldsymbol{q},\boldsymbol{x})$ be intervals of length $\lambda_r(\boldsymbol{q},\boldsymbol{x})$, defined by $\Delta_r(\boldsymbol{q},\boldsymbol{x}) = [\sum_{i=1}^{r-1}\lambda_i(\boldsymbol{q},\boldsymbol{x}), \sum_{i=1}^{r}\lambda_i(\boldsymbol{q},\boldsymbol{x}))$. Let I_S denote the indicator function of a set, S. Let $\boldsymbol{p}(dt,dz)$ be a Poisson random measure over $[0,\infty)\times\mathbb{R}$, which is independent of $\boldsymbol{w}(t)$, with intensity defined by the Lebesgue measure. Then the dynamics can be expressed as

$$d\boldsymbol{q} = \int_{z=0}^{\infty}\sum_{r=1}^{\mathcal{R}}(\theta_r(\boldsymbol{q}) - \boldsymbol{q})I_{\Delta_r(\boldsymbol{q},\boldsymbol{x})}(z)\boldsymbol{p}(dt,dz), \tag{2a}$$

$$d\boldsymbol{x} = f(\boldsymbol{q},\boldsymbol{x})dt + g(\boldsymbol{q},\boldsymbol{x})d\boldsymbol{w} + \int_{z=0}^{\infty}\sum_{r=1}^{\mathcal{R}}(\phi_r(\boldsymbol{q},\boldsymbol{x}) - \boldsymbol{x})I_{\Delta_r(\boldsymbol{q},\boldsymbol{x})}(z)\boldsymbol{p}(dt,dz). \tag{2b}$$

This construction shows that $(\boldsymbol{q}(t),\boldsymbol{x}(t))$ is a Markov process which is adapted to the filtration generated by $(\boldsymbol{w}(t), \int_0^t \boldsymbol{p}(dy,\cdot))$.

An *extended* generator is a commonly used mathematical characterization for the SHS in (1). For a scalar test function $\psi : Q\times\mathbb{R}^n \to \mathbb{R}$ that is twice continuously differentiable with respect to its second argument (i.e., $\boldsymbol{x}$), the extended generator is defined as

$$\lim_{t\downarrow 0}\frac{\mathbb{E}^{q,x}[\psi(\boldsymbol{q}(t),\boldsymbol{x}(t))] - \psi(q,x)}{t} = (\mathcal{L}\psi)(q,x), \tag{3a}$$

where $\mathbb{E}^{q,x}$ denotes the expectation operator conditioned on $\boldsymbol{q}(0) = q$ and $\boldsymbol{x}(0) = x$, while $\mathcal{L}$ is given by Hespanha (2005) and Teel et al. (2014)

$$(\mathcal{L}\psi)(\boldsymbol{q},\boldsymbol{x}) := \frac{\partial\psi(\boldsymbol{q},\boldsymbol{x})}{\partial\boldsymbol{x}}f(\boldsymbol{q},\boldsymbol{x}) + \frac{1}{2}\,\mathrm{Trace}\left(\frac{\partial^2\psi(\boldsymbol{q},\boldsymbol{x})}{\partial\boldsymbol{x}^2}g(\boldsymbol{q},\boldsymbol{x})g(\boldsymbol{q},\boldsymbol{x})^\top\right) + \sum_{r=1}^{\mathcal{R}}\left(\psi\left(\theta_r(\boldsymbol{q}),\phi_r(\boldsymbol{q},\boldsymbol{x})\right) - \psi(\boldsymbol{q},\boldsymbol{x})\right)\lambda_r(\boldsymbol{q},\boldsymbol{x}). \tag{3b}$$

The terms $\frac{\partial\psi(\boldsymbol{q},\boldsymbol{x})}{\partial\boldsymbol{x}}$ and $\frac{\partial^2\psi(\boldsymbol{q},\boldsymbol{x})}{\partial\boldsymbol{x}^2}$ respectively denote the gradient and the Hessian of $\psi(\boldsymbol{q},\boldsymbol{x})$ with respect to $\boldsymbol{x}$. The formulas in (3) can be derived from Itô's lemma and the interlacing method of Applebaum (2009). With the extended generator described above, it is possible to compute time evolution of moments of the SHS that we discuss next.

3. Moment dynamics of SHS

In this section, we describe how the extended generator gives time evolution of its moments. The SHSs considered here are defined over polynomials: for each discrete state $\boldsymbol{q}$, the functions f, g, λ_r, and ϕ_r are polynomials in the continuous state $\boldsymbol{x}$.

3.1. Moment dynamics for SHS with single discrete state

Consider an SHS that has only one discrete state/mode. To simplify notation, we can drop $\boldsymbol{q}$ here. For a given n-tuple $m = (m_1, m_2, \dots, m_n) \in \mathbb{N}^n$, moment dynamics is computed by plugging in a monomial test function

$$\psi(\boldsymbol{x}) = \boldsymbol{x}_1^{m_1}\boldsymbol{x}_2^{m_2}\dots\boldsymbol{x}_n^{m_n} \tag{4}$$

in (3). Here order of the moment $\mathbb{E}(\boldsymbol{x}_1^{m_1}\boldsymbol{x}_2^{m_2}\dots\boldsymbol{x}_n^{m_n})$ is given by $\sum_{i=1}^n m_i$, and there are $\binom{\sum_{i=1}^n m_i+n-1}{n-1}$ moments of order $\sum_{i=1}^n m_i$. The following standard result shows how dynamics of a collection of moments of $\boldsymbol{x}$ evolves over time for a polynomial SHS.

Lemma 1. *Let $f(\boldsymbol{x})$, $g(\boldsymbol{x})$, $\lambda_r(\boldsymbol{x})$ and $\phi_r(\boldsymbol{x})$ be polynomials in $\boldsymbol{x}$. Denoting the vector consisting of all moments up to a specific order of $\boldsymbol{x}$ by $\mathcal{X}$, its time evolution can be compactly written as*

$$\frac{d\mathcal{X}}{dt} = A\mathcal{X} + B\overline{\mathcal{X}} \tag{5}$$

for appropriately defined matrices A, B. Here $\overline{\mathcal{X}}$ is a collection of moments whose order is higher than those stacked up in $\mathcal{X}$.

Proof. Since $f(\boldsymbol{x})$, $g(\boldsymbol{x})$, $\lambda_r(\boldsymbol{x})$ and $\phi_r(\boldsymbol{x})$ are polynomials, the extended generator in (3b) maps monomials of the form $\boldsymbol{x}_1^{m_1}\boldsymbol{x}_2^{m_2}\dots\boldsymbol{x}_n^{m_n}$ to a linear combination of monomials of different orders. Upon collecting all moments up to some order in a vector $\mathcal{X}$, the form in (5) follows from (3a). □

The form of moment dynamics in (5) is well-known (Hespanha, 2005). The matrix B is typically non-zero and the moments contained in $\mathcal{X}$ cannot be computed exactly. This is known as the problem of moment closure (Hespanha, 2005).

3.2. Moment dynamics for SHS with multiple discrete states

Consider an SHS that has a finite, but more than one, discrete states. In this case, it is of interest to know moments of the continuous state given a discrete state and the probability that the system is in the given discrete state. To compute these, we define an N-dimensional state

$$\boldsymbol{b} = (\boldsymbol{b}_1, \boldsymbol{b}_2, \dots, \boldsymbol{b}_N) \in \mathbb{R}^N \tag{6a}$$

such that each $\boldsymbol{b}_i, i = 1,2,\dots,N$ serves as an indicator of the discrete state being $\boldsymbol{q} = s_i$

$$\boldsymbol{b}_i = \begin{cases} 1, & \boldsymbol{q} = s_i, \\ 0, & \text{otherwise.} \end{cases} \tag{6b}$$

For example, when the discrete state $\boldsymbol{q} = s_1$, then we represent it by the tuple $\boldsymbol{b} = (1,0,\dots,0)$. It follows that the following properties hold

$$\sum_{i=1}^N \boldsymbol{b}_i = 1; \quad \boldsymbol{b}_i\boldsymbol{b}_j = 0,\ i \neq j; \quad \boldsymbol{b}_i^2 = \boldsymbol{b}_i. \tag{6c}$$

Furthermore, $\mathbb{E}(\boldsymbol{b}_i)$ is equal to the probability of $\boldsymbol{q} = s_i$, while $\mathbb{E}(\boldsymbol{b}_i\boldsymbol{x}_1^{m_1}\boldsymbol{x}_2^{m_2}\ldots\boldsymbol{x}_n^{m_n})$ is equal to the product of the probability that $\boldsymbol{q} = s_i$ and the moment of $\boldsymbol{x}_1^{m_1}\boldsymbol{x}_2^{m_2}\ldots\boldsymbol{x}_n^{m_n}$, conditioned on $\boldsymbol{q} = s_i$. We can recast the SHS in (1) to the new state space $(\boldsymbol{b}, \boldsymbol{x})$ as described via the following lemma.

Lemma 2. *Consider the SHS described in (1). With $\boldsymbol{b} \in \mathbb{R}^N$ defined in (6), let a single-discrete mode SHS with state space $(\boldsymbol{b}, \boldsymbol{x}) \in \mathbb{R}^{N+n}$ be described by the continuous dynamics*

$$d\begin{bmatrix}\boldsymbol{b}\\ \boldsymbol{x}\end{bmatrix} = \begin{bmatrix}0\\ \sum_{i=1}^N \boldsymbol{b}_i f(s_i, \boldsymbol{x})\end{bmatrix}dt + \begin{bmatrix}0\\ \sum_{i=1}^N \boldsymbol{b}_i g(s_i, \boldsymbol{x})d\boldsymbol{w}\end{bmatrix}, \tag{7a}$$

reset intensities

$$\sum_{i=1}^N \boldsymbol{b}_i\lambda_r(s_i, \boldsymbol{x}), \quad r = 1, 2, \ldots, \mathcal{R}, \tag{7b}$$

and reset maps

$$(\boldsymbol{b}, \boldsymbol{x}) \mapsto \left(\boldsymbol{b} - \sum_{i=1}^N \boldsymbol{b}_i\mathbb{1}_{s_i} + \sum_{i=1}^N \boldsymbol{b}_i\mathbb{1}_{\theta_r(s_i)}, \sum_{i=1}^N \boldsymbol{b}_i\phi_r(s_i, \boldsymbol{x})\right). \tag{7c}$$

Then (7) recasts (1) in $(\boldsymbol{b}, \boldsymbol{x})$ space.

Proof. Let $\boldsymbol{q} = s_j \in Q$. Then (6) implies that dynamics of $\boldsymbol{x}$ in (7a) becomes

$$d\boldsymbol{x} = f(s_j, \boldsymbol{x})dt + g(s_j, \boldsymbol{x})d\boldsymbol{w}, \tag{8}$$

which is same as (1a). Likewise, the reset intensities for both (7) and (1) take the form

$$\lambda_r(s_j, \boldsymbol{x}), \quad r = 1, 2, \ldots, \mathcal{R}. \tag{9}$$

As for the reset maps, (7c) yields

$$\left(\mathbb{1}_{s_j}, \boldsymbol{x}\right) \mapsto \left(\mathbb{1}_{s_j} - \mathbb{1}_{s_j} + \mathbb{1}_{\theta_r(s_j)}, \phi_r(s_j, \boldsymbol{x})\right), \tag{10}$$

which by definition in (6) is same as (1c)

$$\left(s_j, \boldsymbol{x}\right) \mapsto \left(\theta_r(s_j), \phi_r(s_j, \boldsymbol{x})\right). \tag{11}$$

Since we arbitrarily chose $\boldsymbol{q} = s_j \in Q$, the equivalence between the two SHSs will hold true for any $\boldsymbol{q}$. □

To write the moment dynamics of SHS in (7), we can use monomial test functions

$$\psi(\boldsymbol{b}, \boldsymbol{x}) = \boldsymbol{b}_1^{m_1}\boldsymbol{b}_2^{m_2}\ldots\boldsymbol{b}_N^{m_N}\boldsymbol{x}_1^{m_{N+1}}\boldsymbol{x}_2^{m_{N+2}}\ldots\boldsymbol{x}_n^{m_{N+n}}, \tag{12}$$

supplemented with the constraints in (6c). It is worth noting that (7) is a polynomial SHS in $(\boldsymbol{b}, \boldsymbol{x})$ space if the original SHS is polynomial in $\boldsymbol{x}$. The following result provides a general form for the moment dynamics.

Theorem 3. *Consider the SHS in (7). Let f, g, λ_r and ϕ_r be polynomials in $\boldsymbol{x}$. Denoting the vector consisting of all moments up to a specific order of the state $(\boldsymbol{b}, \boldsymbol{x})$ by $\mathcal{X}$, its time evolution can be compactly written as*

$$\frac{d\mathcal{X}}{dt} = A\mathcal{X} + B\bar{\mathcal{X}}, \tag{13a}$$

$$0 = C\mathcal{X} + D\bar{\mathcal{X}} \tag{13b}$$

for appropriately defined matrices A, B, C, D. Here $\bar{\mathcal{X}}$ is a collection of moments whose order is higher than those stacked up in $\mathcal{X}$.

Proof. Since (7) is polynomial in $(\boldsymbol{b}, \boldsymbol{x})$, the form in (13a) follows from Lemma 1. The property $\boldsymbol{b}_i\boldsymbol{b}_j = 0$ in (6c) implies that for a non-zero $m_i \in \mathbb{N}$, all moments except those of the form $\mathbb{E}\left(\boldsymbol{b}_i^{m_i}\boldsymbol{x}_1^{m_{N+1}}\boldsymbol{x}_1^{m_{N+2}}\ldots\boldsymbol{x}_n^{m_{N+n}}\right)$ are zero. Furthermore, $\boldsymbol{b}_i^2 = \boldsymbol{b}_i$ results in

$$\begin{aligned}&\mathbb{E}\left(\boldsymbol{b}_i^{m_i}\boldsymbol{x}_1^{m_{N+1}}\boldsymbol{x}_1^{m_{N+2}}\ldots\boldsymbol{x}_n^{m_{N+n}}\right)\\ &\quad = \mathbb{E}\left(\boldsymbol{b}_i\boldsymbol{x}_1^{m_{N+1}}\boldsymbol{x}_1^{m_{N+2}}\ldots\boldsymbol{x}_n^{m_{N+n}}\right),\end{aligned} \tag{14}$$

for all $m_i \geq 1$. The constraint $\sum_{i=1}^N \boldsymbol{b}_i = 1$ results in

$$\begin{aligned}&\sum_{i=1}^N \mathbb{E}\left(\boldsymbol{b}_i\boldsymbol{x}_1^{m_{N+1}}\boldsymbol{x}_1^{m_{N+2}}\ldots\boldsymbol{x}_n^{m_{N+n}}\right)\\ &\quad - \mathbb{E}\left(\boldsymbol{x}_1^{m_{N+1}}\boldsymbol{x}_1^{m_{N+2}}\ldots\boldsymbol{x}_n^{m_{N+n}}\right) = 0.\end{aligned} \tag{15}$$

These three constraints can be compactly represented by (13b). □

Remark 4. Many of the moments contained in $\mathcal{X}$ and $\overline{\mathcal{X}}$ in Theorem 3 are equal to zero. In practice we do not include them in $\mathcal{X}$ and $\overline{\mathcal{X}}$. Similarly, higher order moments that are equal to lower order moments, as in (14), are not included.

4. Bounding moment dynamics

Although the higher order moments appear in $\overline{\mathcal{X}}$ in (13), they cannot take arbitrary values and must conserve semidefinite properties (Lamperski et al., 2019). The following lemma, adapted from Lamperski et al. (2019), formally states this.

Lemma 5 (*Lamperski et al., 2019*)**.** *Let $\nu_1(\boldsymbol{x}), \ldots, \nu_p(\boldsymbol{x})$ be any collection of polynomials. There is an affine matrix-valued function M such that the following holds:*

$$\left\langle \begin{bmatrix}\nu_1(\boldsymbol{x})\\ \vdots\\ \nu_m(\boldsymbol{x})\end{bmatrix}\begin{bmatrix}\nu_1(\boldsymbol{x})\\ \vdots\\ \nu_m(\boldsymbol{x})\end{bmatrix}^\top\right\rangle = M(\mathcal{X}, \overline{\mathcal{X}}) \succeq 0. \tag{16}$$

Furthermore, if $h_i(\boldsymbol{x}) \geq 0$ and $\nu_1(\boldsymbol{x}), \ldots, \nu_{p_i}(\boldsymbol{x})$ are polynomials, then there is a different affine matrix-valued function M_{h_i} such that

$$\left\langle h_i(\boldsymbol{x})\begin{bmatrix}\nu_1(\boldsymbol{x})\\ \vdots\\ \nu_{p_i}(\boldsymbol{x})\end{bmatrix}\begin{bmatrix}\nu_1(\boldsymbol{x})\\ \vdots\\ \nu_{p_i}(\boldsymbol{x})\end{bmatrix}^\top\right\rangle = M_{h_i}(\mathcal{X}, \overline{\mathcal{X}}) \succeq 0. \tag{17}$$

Utilizing the matrices from Lemma 5, the problem of computing bounds on a moment of interest, i.e., an element of $\mathcal{X}$, can be formulated as an optimal control problem. Suppose that $\mathcal{X}_j$ denotes the jth element of $\mathcal{X}$. A lower bound on $\mathcal{X}_j(\tau) \in \mathcal{X}(\tau)$ can be computed as (Lamperski et al., 2019)

$$\underset{\mathcal{X}(t), \overline{\mathcal{X}}(t)}{\text{minimize}} \quad \mathcal{X}_j(\tau) \tag{18a}$$

$$\text{subjectto} \quad \frac{d\mathcal{X}}{dt} = A\mathcal{X}(t) + B\overline{\mathcal{X}}(t) \tag{18b}$$

$$0 = C\mathcal{X}(t) + D\overline{\mathcal{X}}(t) \tag{18c}$$

$$M(\mathcal{X}(t), \overline{\mathcal{X}}(t)) \succeq 0 \tag{18d}$$

$$M_{h_i}(\mathcal{X}(t), \overline{\mathcal{X}}(t)) \succeq 0 \tag{18e}$$

$$\mathcal{X}(0) = \mathcal{X}_0 \tag{18f}$$

for all $t \in [0, \tau]$. An upper bound can be computed by maximizing the objective function. Moreover, if the number of moments stacked in $\mathcal{X}$ is increased and correspondingly the sizes of M and M_{h_i} are increased, the lower and upper bounds often improve (Lamperski et al., 2019). Theoretically, the increase in sizes of M and M_{h_i} implies that more constraints are added and therefore the bounds cannot get worse. However, in practice they improve and converge to the true moment value.

Note that in (18), the vector $\mathcal{X}(t)$ depends on t but not the transition count, k. This is because transitions are averaged out by the extended generator, (3).

If bounds on only stationary moments are desired, then the semidefinite program becomes simpler. Specifically, if the given SHS has a stationary distribution then lower bound a stationary moment $\mathcal{X}_j \in \mathcal{X}$ can be computed as

$$\underset{\mathcal{X},\bar{\mathcal{X}}}{\text{minimize}} \quad \mathcal{X}_j \tag{19a}$$
$$\text{subjectto} \quad 0 = A\mathcal{X} + B\bar{\mathcal{X}} \tag{19b}$$
$$0 = C\mathcal{X} + D\bar{\mathcal{X}} \tag{19c}$$
$$M(\mathcal{X}, \overline{\mathcal{X}}) \succeq 0 \tag{19d}$$
$$M_{h_i}(\mathcal{X}, \bar{\mathcal{X}}) \succeq 0 \tag{19e}$$

Here we require that the SHS has finite stationary moments. This assumption avoids some pathological cases wherein the stationary moments may not exist (Glynn, Zeevi, et al., 2008). The reader may refer to DeVille, Dhople, Domínguez-García, and Zhang (2016) and Meyn and Tweedie (2012) for details on the existence of stationary distributions.

Remark 6. If the first column of A has all its elements zeros then we obtain trivial lower bounds on stationary moments. This happens because for such systems, the degenerate distribution is always a stationary distribution.

Remark 7. In principle, additional algebraic constraints can be included in the setup in (18) or in (19). Thus, our setup is amenable to moment analysis of non-polynomial SHSs as long as they could be cast as polynomial SHSs with algebraic constraints via augmentation of states. These algebraic constraints have to be linear in $\mathcal{X}$ and $\overline{\mathcal{X}}$.

5. Illustrative example

We illustrate our methodology using an example. To this end, we use a modified version of an SHS model of Transmission Control Protocol (TCP) from Hespanha (2005, 2006).

Example 1 (*TCP On–Off Hespanha, 2005, 2006*)**.** Let $\boldsymbol{v}$ be the continuous state, which represents the congestion window size of the TCP. There are three discrete states/modes, namely, $\{off, ss, ca\}$, which stand for off, slow start, and congestion avoidance, respectively. During these modes, the continuous-state evolves as

$$d\boldsymbol{v} = \begin{cases} (\frac{\log 2}{R}\boldsymbol{v} + \delta)dt, & \boldsymbol{q} = ss \\ \frac{1}{R}dt + \sigma \boldsymbol{v} d\boldsymbol{w}, & \boldsymbol{q} = ca \\ 0\, dt, & \boldsymbol{q} = off \end{cases} \tag{20}$$

Here R is the round trip time, δ is a basal rate of increase in window size, and σ is the noise intensity. We note that in Hespanha (2006), these equations do not have the terms δ and σ.

The transitions between the discrete modes are of three types: drop occurrences ($ss \mapsto ca$, $ca \mapsto ca$), start of new flow ($off \mapsto ss$), and termination of flows ($ss \mapsto off$ and $ca \mapsto off$). These are described via the reset maps

$$\phi_{drop}(\boldsymbol{q}, \boldsymbol{v}) = \begin{cases} \left(ca, \frac{\boldsymbol{v}}{2}\right), & \boldsymbol{q} \in \{ss, ca\} \\ (off, \boldsymbol{v}), & \boldsymbol{q} = off \end{cases} \tag{21a}$$

$$\phi_{start}(\boldsymbol{q}, \boldsymbol{v}) = \begin{cases} (\boldsymbol{q}, \boldsymbol{v}), & \boldsymbol{q} \in \{ss, ca\} \\ (ss, v_0), & \boldsymbol{q} = off \end{cases} \tag{21b}$$

$$\phi_{end}(\boldsymbol{q}, \boldsymbol{v}) = \begin{cases} (off, 0), & \boldsymbol{q} \in \{ss, ca\} \\ (off, \boldsymbol{v}), & \boldsymbol{q} = off \end{cases} \tag{21c}$$

with reset intensities

$$\lambda_{drop}(\boldsymbol{q}, \boldsymbol{v}) = \begin{cases} \frac{p\boldsymbol{v}}{R}, & \boldsymbol{q} \in \{ss, ca\} \\ 0, & \boldsymbol{q} = off \end{cases} \tag{22a}$$

$$\lambda_{start}(\boldsymbol{q}, \boldsymbol{v}) = \begin{cases} 0, & \boldsymbol{q} \in \{ss, ca\} \\ \frac{1}{\tau_{off}}, & \boldsymbol{q} = off \end{cases} \tag{22b}$$

$$\lambda_{end}(\boldsymbol{q}, \boldsymbol{v}) = \begin{cases} \frac{\boldsymbol{v}}{kR}, & \boldsymbol{q} \in \{ss, ca\} \\ 0, & \boldsymbol{q} = off, \end{cases} \tag{22c}$$

where p is packet drop rate.

To recast this SHS as a single-mode SHS, we define the indicator state variables $\boldsymbol{b}_1$, $\boldsymbol{b}_2$, and $\boldsymbol{b}_3$ as in (6b). The equivalent SHS is described via the continuous dynamics

$$d\begin{bmatrix}\boldsymbol{b}_1 & \boldsymbol{b}_2 & \boldsymbol{b}_3 & \boldsymbol{v}\end{bmatrix}^\top = \begin{bmatrix}0 & 0 & 0 & \boldsymbol{b}_1\left(\frac{\log 2}{R}v + \delta\right) + \frac{\boldsymbol{b}_2}{R}\end{bmatrix}^\top dt + \begin{bmatrix}0 & 0 & 0 & \boldsymbol{b}_2\sigma\boldsymbol{v}d\boldsymbol{w}\end{bmatrix}^\top. \tag{23}$$

The reset maps and the reset intensities are given by

$$\phi_{drop}(\boldsymbol{b}, \boldsymbol{v}) = \boldsymbol{b}_1\begin{bmatrix}\boldsymbol{b}_1 - 1 & \boldsymbol{b}_2 + 1 & \boldsymbol{b}_3 & \boldsymbol{v}/2\end{bmatrix}^\top + \boldsymbol{b}_2\begin{bmatrix}\boldsymbol{b}_1 & \boldsymbol{b}_2 & \boldsymbol{b}_3 & \boldsymbol{v}/2\end{bmatrix}^\top, \tag{24a}$$

$$\phi_{start}(\boldsymbol{b}, \boldsymbol{v}) = \boldsymbol{b}_3\begin{bmatrix}\boldsymbol{b}_1 + 1 & \boldsymbol{b}_2 & \boldsymbol{b}_3 - 1 & v_0\end{bmatrix}^\top, \tag{24b}$$

$$\phi_{end}(\boldsymbol{b}, \boldsymbol{v}) = \boldsymbol{b}_1\begin{bmatrix}\boldsymbol{b}_1 - 1 & \boldsymbol{b}_2 & \boldsymbol{b}_3 + 1 & 0\end{bmatrix}^\top + \boldsymbol{b}_2\begin{bmatrix}\boldsymbol{b}_1 & \boldsymbol{b}_2 - 1 & \boldsymbol{b}_3 + 1 & 0\end{bmatrix}^\top, \tag{24c}$$

$$\lambda_{drop}(\boldsymbol{b}, \boldsymbol{v}) = (\boldsymbol{b}_1 + \boldsymbol{b}_2)\frac{p\boldsymbol{v}}{R}, \quad \lambda_{start}(\boldsymbol{b}, \boldsymbol{v}) = \frac{\boldsymbol{b}_3}{\tau_{off}},$$
$$\lambda_{end}(\boldsymbol{b}, \boldsymbol{v}) = (\boldsymbol{b}_1 + \boldsymbol{b}_2)\frac{\boldsymbol{v}}{kR}. \tag{25}$$

To obtain moment dynamics, we can write time evolution of $\mathbb{E}\left(\boldsymbol{b}_1^{m_1}\boldsymbol{b}_2^{m_2}\boldsymbol{b}_3^{m_3}\boldsymbol{v}^{m_4}\right)$. However, algebraic constraints in (6c) and the absence of continuous dynamics in *off* mode imply that we only need dynamics of moments of the form $\mathbb{E}(\boldsymbol{b}_1\boldsymbol{v}^{m_4})$, and $\mathbb{E}(\boldsymbol{b}_2\boldsymbol{v}^{m_4})$. All the other moments are either zero or can be obtained via a linear combination of these. Their dynamics is given by

$$\frac{d\mathbb{E}(\boldsymbol{b}_1\boldsymbol{v}^{m_4})}{dt} = \frac{m_4\log(2)}{R}\mathbb{E}\left(\boldsymbol{b}_1\boldsymbol{v}^{m_4}\right) + m_4\delta\mathbb{E}\left(\boldsymbol{b}_1\boldsymbol{v}^{m_4-1}\right) + \frac{v_0^{m_4}}{\tau_{off}}\mathbb{E}(\boldsymbol{b}_3) - \left(\frac{p}{R} + \frac{1}{kR}\right)\mathbb{E}(\boldsymbol{b}_1\boldsymbol{v}^{m_4+1}), \tag{26a}$$

$$\frac{d\mathbb{E}(\boldsymbol{b}_2\boldsymbol{v}^{m_4})}{dt} = \frac{m_4}{R}\mathbb{E}\left(\boldsymbol{b}_2\boldsymbol{v}^{m_4-1}\right) + \frac{\sigma^2 m_4(m_4-1)}{2}\mathbb{E}\left(\boldsymbol{b}_2\boldsymbol{v}^{m_4}\right) + \frac{p}{2^{m_4}R}\mathbb{E}(\boldsymbol{b}_1\boldsymbol{v}^{m_4+1}) - \left(\frac{p(2^{m_4}-1)}{2^{m_4}R} + \frac{1}{kR}\right)\mathbb{E}(\boldsymbol{b}_2\boldsymbol{v}^{m_4+1}). \tag{26b}$$

Using these moment equations, along with the semidefinite constraints and algebraic constraints arising from the definition of $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3$, a semidefinite program as in (19) is set up. The matrices M_i are generated using the non-negativity of $\boldsymbol{b}_1$, $\boldsymbol{b}_2$, $1 - \boldsymbol{b}_1$, and $1-\boldsymbol{b}_2$. We solved these programs using YALMIP wrapper (Löfberg, 2004), with SDPA-GMP solver (Nakata, 2010). Taking specific values of $R = 2$, $\tau_{off} = 1$, $k = 3$, $p = 0.5$, $v_0 = 1$, $\delta = 0.1$, and $\sigma = 0.01$, we obtain bounds on $\mathbb{E}(\boldsymbol{b}_1)$ and $\mathbb{E}(\boldsymbol{b}_2)$. As expected, incorporating higher order moments improves the moment estimates (see Fig. 1). Recall that $\boldsymbol{b}_3 = 1 - \boldsymbol{b}_1 - \boldsymbol{b}_2$, so bounds on $\mathbb{E}(\boldsymbol{b}_3)$ can be obtained from these.

6. Conclusion

Moments of an SHS are described via infinite dimensional coupled differential equations, which typically cannot be solved

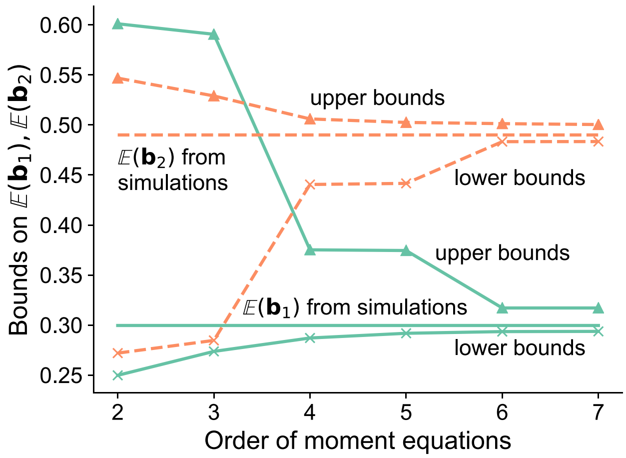

Fig. 1. Bounds on $\mathbb{E}(\boldsymbol{b}_1)$ and $\mathbb{E}(\boldsymbol{b}_2)$ (i.e., the probability that the system is in the mode *ss* and *ca*, respectively). The bounds for both moments improve and converge to their respective true values as the order of moments equations used in the semidefinite program is increased. To numerically verify these bounds, 10 000 simulation runs were averaged.

for a few lower order moments without knowing the higher order moments. In this paper, we presented a semidefinite programming based method to compute exact bounds on the moments of an SHS. Although theoretically our method computes bounds on both transient and stationary moments, its applicability is limited by the scaling of and numerical conditioning of the semidefinite programs. The transient case is particularly challenging because discretization of the time interval leads to large semidefinite programs. On the positive side, whenever the semidefinite program is solvable using current solvers, the estimate on moments is significantly faster than those obtained via simulations. For Example 1, a personal machine (2.9 GHz Intel Core i7 processor, 8 GB 1600 MHz DDR3 RAM) takes less than one second to compute the 7th-order moment bounds whereas it takes several minutes for 10 000 simulations. Our focus of future research would be to improve scalability and robustness of the technique.

References

Applebaum, David (2009). *Lévy processes and stochastic calculus*. Cambridge University Press.

DeVille, Lee, Dhople, Sairaj, Domínguez-García, Alejandro D., & Zhang, Jiangmeng (2016). Moment closure and finite-time blowup for piecewise deterministic markov processes. *SIAM Journal on Applied Dynamical Systems*, *15*, 526–556.

Ghosh, Mrinal K., Arapostathis, Aristotle, & Marcus, Steven I. (1997). Ergodic control of switching diffusions. *SIAM Journal on Control and Optimization*, *35*, 1952–1988.

Ghusinga, Khem Raj, Vargas-Garcia, Cesar A., Lamperski, Andrew, & Singh, Abhyudai (2017). Exact lower and upper bounds on stationary moments in stochastic biochemical systems. *Physical Biology*, *14*, 04LT01.

Glynn, Peter W., Zeevi, Assaf, et al. (2008). Bounding stationary expectations of markov processes. In *Markov processes and related topics: A festschrift for Thomas G. Kurtz* (pp. 195–214). Institute of Mathematical Statistics.

Hespanha, Joao P. (2005). A model for stochastic hybrid systems with application to communication networks. *Nonlinear Analysis. Theory, Methods & Applications*, *62*, 1353–1383.

Hespanha, Joao P. (2006). Modelling and analysis of stochastic hybrid systems. *IEE Proceedings D (Control Theory and Applications)* , *153*, 520–535.

Kuehn, Christian (2016). *Moment closure—a brief review* (pp. 253–271). Springer International Publishing.

Kuntz, Juan, Ottobre, Michela, Stan, Guy-Bart, & Barahona, Mauricio (2016). Bounding stationary averages of polynomial diffusions via semidefinite programming. *SIAM Journal on Scientific Computing*, *38*, A3891–A3920.

Lamperski, Andrew, Ghusinga, Khem Raj, & Singh, Abhyudai (2019). Analysis and control of stochastic systems using semidefinite programming over moments. *IEEE Transactions in Automatic Control*, *64*.

Li, Xiangfang, Omotere, Oluwaseyi, Qian, Lijun, & Dougherty, Edward R. (2017). Review of stochastic hybrid systems with applications in biological systems modeling and analysis. *EURASIP Journal on Bioinformatics and Systems Biology*, *2017*, 8.

Löfberg, J. (2004). Yalmip : A toolbox for modeling and optimization in matlab. In *Proceedings of the CACSD conference*.

Meyn, Sean P., & Tweedie, Richard L. (2012). *Markov chains and stochastic stability*. Springer Science & Business Media.

Nakata, Maho (2010). A numerical evaluation of highly accurate multiple-precision arithmetic version of semidefinite programming solver: Sdpa-gmp, -qd and-dd. In *Computer-aided control system design (CACSD), 2010 IEEE international symposium on* (pp. 29–34). IEEE.

Sakurai, Yuta, & Hori, Yutaka (2018). Optimization-based synthesis of stochastic biocircuits with statistical specifications. *Journal of the Royal Society Interface*, *15*, 20170709.

Socha, Leslaw (2007). *Linearization methods for stochastic dynamic systems (vol. 730)*. Springer Science & Business Media.

Soltani, Mohammad, & Singh, Abhyudai (2017). Moment-based analysis of stochastic hybrid systems with renewal transitions. *Automatica*, *84*, 62–69.

Teel, Andrew R., Subbaraman, Anantharaman, & Sferlazza, Antonino (2014). Stability analysis for stochastic hybrid systems: A survey. *Automatica*, *50*, 2435–2456.

Yin, G. George, & Zhu, Chao (2009). *Hybrid switching diffusions: properties and applications (vol. 63)*. Springer Science & Business Media.

Khem Raj Ghusinga received the B.Tech. and the M.Tech. degrees in Electrical Engineering from the Indian institute of Technology–Bombay, Mumbai, India. He then obtained his Ph.D. in Electrical and Computer Engineering from the University of Delaware, Newark, DE, USA. He is currently a postdoctoral researcher at the University of North Carolina at Chapel Hill, NC, USA. His research interests include dynamics and control of stochastic systems, with applications to systems biology.

Andrew Lamperski received the B.S. degree in biomedical engineering and mathematics in 2004 from the Johns Hopkins University, Baltimore, MD, and the Ph.D. degree in control and dynamical systems at the California Institute of Technology, Pasadena. He held postdoctoral positions in control and dynamical systems at the California Institute of Technology from 2011 to 2013 and in mechanical engineering at the Johns Hopkins University in 2013. From 2012 to 2014, he did postdoctoral work in the Department of Engineering, University of Cambridge, on a scholarship from the Whitaker International Program. In 2014, he joined the Department of Electrical and Computer Engineering, University of Minnesota as an Assistant Professor. His research interests include optimal control and estimation, with applications to neuroscience and robotics.

Abhyduai Singh earned his bachelor's degree in mechanical engineering from the Indian Institute of Technology in Kanpur, India. He received master's degrees in both Mechanical and Electrical & Computer engineering from Michigan State University, and a master's degree in Ecology, Evolution and Marine Biology from University of California Santa Barbara (UCSB). After earning his doctoral degree in Electrical & Computer engineering in 2008, also from UCSB, he completed postdoctoral work in UC San Diegos Department of Chemistry and Biochemistry. He joined the University of Delaware in 2011 as an Assistant Professor of Electrical & Computer Engineering, and Biomedical Engineering, where he is now an Associate Professor since 2017.