Moment Propagation of Polynomial Systems Through Carleman Linearization for Probabilistic Safety Analysis

Sasinee Pruekprasert [a] , Jeremy Dubut [a] , Toru Takisaka [b] , Clovis Eberhart [c,e] ,
Ahmet Cetinkaya [d]

[a]*National Institute of Advanced Industrial Science and Technology, Tokyo, Japan*

[b]*University of Electronic Science and Technology of China, Chengdu, China*

[c]*National Institute of Informatics, Tokyo, 101-8430, Japan*

[d]*Shibaura Institute of Technology, Tokyo, 135-8548, Japan*

[e]*Japanese-French Laboratory of Informatics, IRL 3527, CNRS, Tokyo, Japan*

Abstract

We develop a method to approximate the moments of a discrete-time stochastic polynomial system. Our method is built upon Carleman linearization with truncation. Specifically, we take a stochastic polynomial system with finitely many states and transform it into an infinite-dimensional system with linear deterministic dynamics, which describe the exact evolution of the moments of the original polynomial system. We then truncate this deterministic system to obtain a finite-dimensional linear system, and use it for moment approximation by iteratively propagating the moments along the finite-dimensional linear dynamics across time. We provide efficient online computation methods for this propagation scheme with several error bounds for the approximation. Our results also show that precise values of certain moments at a given time step can be obtained when the truncated system is sufficiently large. Furthermore, we investigate techniques to reduce the offline computation load using reduced Kronecker power. Based on the obtained approximate moments and their errors, we also provide hyperellipsoidal regions that are safe for some given probability bound. Those bounds allow us to conduct probabilistic safety analysis online through convex optimization. We demonstrate our results on a logistic map with stochastic dynamics and a vehicle dynamics subject to stochastic disturbance.

Key words: Stochastic systems, nonlinear systems, probabilistic safety analysis, moment computation, Carleman linearization

1 Introduction

Uncertainty is one of the critical issues that make safety assurance of cyber-physical systems a difficult task. Handling uncertainty in automated driving systems is especially challenging, as motion planning algorithms are required to take

account of uncertainty in both the measurement and the actuation mechanisms (Bry and Roy, 2011). Depending on the dynamical variations in the environment and the vehicle-controller interaction, the actual trajectory taken by a vehicle may differ substantially from a trajectory that is precomputed based on nominal dynamics of the vehicle. Most motion planning algorithms address this issue by assessing the collision risk for a set of possible future trajectories of the vehicle and not just for the nominal trajectory. Moreover, there are different approaches regarding how the uncertainty is modeled. A well known approach is based on considering known deterministic bounds on the disturbance and the measurement noise (see Althoff and Dolan (2014), and the references therein). Another approach is to use stochastic system models to describe the effects of uncertainty.

The effect of uncertainty on the state of a stochastic system can be quantified through the so-called *mean and covari-*
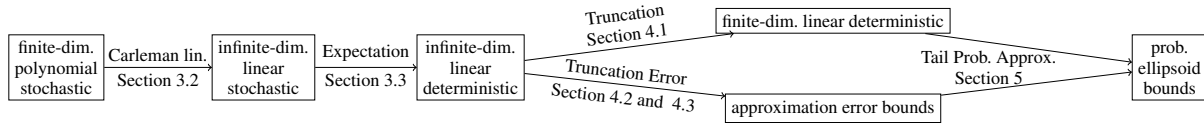
Fig. 1. The different steps of the proposed method

ance propagation method. This method is especially effective for systems with linear dynamics where the uncertainty is modeled as additive Gaussian noise. For those systems, if the initial state distribution is Gaussian, the distribution of the state at any future time remains Gaussian. Moreover, the mean and the covariance of the state can be computed iteratively using linear equations. Previously, Bry and Roy (2011) and Banzhaf *et al.* (2018) used this together with Kalman filters in motion planning tasks.

While mean and covariance propagation is an effective method for linear Gaussian systems, many cyber-physical systems involve nonlinearities and the noise cannot always be modeled to be an additive noise or a Gaussian one. Our goal is to develop a moment propagation method for nonlinear systems where the noise can be non-additive and non-Gaussian. Specifically, our propagation method can approximate the mean, covariance, and higher moments of such systems. We consider discrete-time stochastic nonlinear systems with polynomial dynamics, where the coefficients are randomly varying. Our model can describe both additive and multiplicative noise. Our propagation method is based on Carleman linearization with truncation (Steeb and Wilhelm, 1980). By following the Carleman linearization technique, we obtain a new dynamical system that describes how the Kronecker powers of the system state evolve. This system is linear, but it is infinite-dimensional. The expectation of the states of this linear system correspond to all moments of the original nonlinear system. By truncating the expected dynamics of the linear system at a certain truncation limit, we obtain a finite-dimensional linear system. Using the equations of this finite-dimensional system, our method iteratively computes the approximate moments of the nonlinear system up to the provided truncation limit.

The mean, covariance, and higher moments of a system can provide useful statistical information regarding possible future state trajectories and can be used in safety verification. In this paper, we use our moment approximation method as a basis to develop a stochastic safety verification framework. Our framework uses tail probability bounds (Gray and Wang, 1991). We provide hyperellipsoidal safety region, for which the probability of the system state to be outside that region is within a given probability bound. Such safety regions are characterized using positive-semidefinite matrices. Previously, a scenario-based approach was used by Shen *et al.* (2020) to obtain similar safety regions for vehicles. This scenario-based approach is different from our moment-based approach: it relies on generating sufficiently many sample state trajectories using the information of the distributions of random elements in the dynamics. In our approach, we do not need to generate sample trajectories.

In this paper, we propose a novel tail probability-based safety verification framework consisting of two phases. The first one is an offline computation phase to compute the approximate moment dynamics and compute the upper bounds for their errors. The second phase is a fast online computation to propagate the moments and their errors through the truncated dynamics, then use them to compute probabilistic ellipsoid bounds. A key ingredient in our framework is to efficiently compute bounds on moment approximation errors. First, we present the exact approximation error for a particular moment as a sum. However, for the exact computation, the number of terms is a quickly growing function of the dimension of the system and the time. We provide efficient online computation methods for this propagation scheme with several error bounds for the approximation. Furthermore, we develop novel techniques to reduce the offline computation load using reduced Kronecker power.

Remark that, similarly to Bry and Roy (2011), this paper focuses on discrete-time systems. We could also take into account the error due to going from continuous-time to discrete-time dynamics by incorporating Lagrange remainders in our analysis, but we decided against it to keep the presentation of our method simple. Remark also that extending the analysis of Forets and Pouly (2017) on continuous-time systems by adding general noise with both additive and multiplicative noise terms is not easily achievable, which motivated the present work.

We present two case studies to demonstrate various aspects of our method. Our first case study is on a scalar stochastic logistic map. In particular, we consider uniformly distributed growth/decay rates in the dynamics. We run moment approximation and tail probability-based safety analysis. In the safety analysis, we check the conservativeness of our tail-probability bounds through Monte Carlo simulations. We observe that our obtained safety regions are quite tight for small durations. We also observe that our method is advantageous in terms of speed. It does not require drawing random variables, and as a result, it is much faster than Monte Carlo simulations even with small number of samples. Another advantage of our method is that it provides a theoretical guarantee for obtaining the safety regions.

Our second case study is on a vehicle with kinematic bicycle model. First, we use second-order Taylor expansion to obtain a discrete-time nonlinear polynomial system describing the vehicle. Then we use our approximate moment propagation approach to study how the vehicle's future states are affected by the uncertainty in the acceleration and in the initial vehicle state. Our approach uses the moments of the

initial state of the vehicle and the moments for the acceleration to approximate future moments of the state. Here the uncertainty in the initial state is caused by the measurement noise and thus initial state moments are derived from the statistical properties of the measurement error. On the other hand uncertainty in the acceleration is related to actuators and the environment. In a typical control loop, our approach can be applied in a receding horizon fashion. In particular, computation is done by first obtaining new state measurements which are used for specifying the moments of the new "initial" state; then these moments are propagated to approximately compute the moments of the state associated with future time instants.

The rest of the paper is organized as follows (see also Figure 1). Section 2 presents related work on Carleman linearization. In Section 3, we start from a finite-dimensional stochastic polynomial system, on which we apply the Carleman linearization technique to get an infinite-dimensional linear stochastic system. Then, we turn the linear stochastic system into a deterministic one by taking expectation. We present our truncation-based moment approximation method and study the error it introduces in Section 4. Then, in Section 5, we provide probabilistic ellipsoid bounds based on approximations of the first and second moments. Section 6 presents techniques to reduce the offline computation load using the concept of reduced Kronecker power. We provide two numerical examples to demonstrate the applicability of our approach in Section 7. Finally, in Section 8, we conclude the paper.

2 Related Work

The characterization of the dynamics under truncated Carleman linearization and the truncation error analysis in Section 3 appeared previously in our preliminary conference report (Pruekprasert *et al.*, 2020).

In this paper, we present new results on tail-probability analysis and a two-step approximation procedure for confidence ellipsoid bounds of the system state in Section 5. We investigate techniques to improve the computational efficiency of our methods in Section 6. We also provide a new upper bound for the approximation error from truncation in Section 4.3.3, using partial exact computation with indices on moment coordinates. We illustrate these new results and techniques in Section 7 through new simulations for a case study on vehicle dynamics.

The Carleman linearization technique is well-known in the nonlinear systems literature. For deterministic systems, it has been used for approximation of nonlinear models by linear ones (Bellman and Richardson, 1963; Steeb and Wilhelm, 1980; Al-Tuwaim *et al.*, 1998). Recently, Amini *et al.* (2019) used a Carleman linearization approach to design state feedback controllers for continuous-time nonlinear polynomial systems, Hashemian and Armaou (2019) used Carleman linearization in model predictive control of continuous-time

deterministic nonlinear systems, and Amini *et al.* (2021) obtained error bounds for Carleman linearization with truncation. Furthermore, Amini *et al.* (2020) proposed control frameworks based on Carleman linearization. There, an approach using Carleman linearization with truncation allows the authors to approximately represent the Hamilton-Jacobi-Bellman equation (arising in optimal control of nonlinear systems) as an operator equation in quadratic form. This representation is then used to derive the approximate value function of the optimal control system as a quadratic Lyapunov function. The main advantage of the method is that it yields an iterative procedure to approximate the value function that converges to the true value. In our work, we do not investigate the optimal control problem and we use Carleman linearization with truncation to approximate moments of stochastic systems instead of value functions.

There are fewer results on Carleman linearization for stochastic systems. Specifically, Wong (1983) investigated bilinear noise terms in Ito-type stochastic differential equations and used Carleman linearization to describe how the moments of the stochastic differential equations evolve. Furthermore, Rauh *et al.* (2009) considered continuous-time nonlinear systems with additive noise and used a Carleman linearization technique in conjunction with a series expansion approach to approximate such systems with discrete-time linear systems. They then demonstrated that a Kalman filter can be used on the obtained linear systems for state estimation. In addition, Cacace *et al.* (2014) and Cacace *et al.* (2019) proposed Carleman linearization-based sampled-data filters for nonlinear stochastic differential equations driven by Wiener noise. More recently, Jasour *et al.* (2021) used a technique akin to Carleman linearization in moment propagation of a class of discrete-time mixed trigonometric polynomial systems with probabilistic disturbance. They show that several systems of interest are mixed trigonometric polynomial of degree 1, for which moment propagation without error is computationally feasible. The truncation error analysis in our paper is partly motivated by the work of Forets and Pouly (2017), which derives tight approximation error bounds for the truncated Carleman linearization of deterministic continuous-time systems. We note that deriving linear representations of finite-dimensional nonlinear systems can also be achieved through the use of Koopman operators (Goswami and Paley, 2017; Mesbahi *et al.*, 2019). A Koopman operator approach has recently been used in model predictive control of vehicles with deterministic nonlinear dynamics (Cibulka *et al.*, 2020).

Notations. We use $\mathbb{R}$ and $\mathbb{N}$ for the sets of real numbers and non-negative integers, respectively. We write 1 for the (1×1)-matrix $\begin{bmatrix} 1 \end{bmatrix}$. We denote by $M_1 \otimes M_2$ the Kronecker product of two matrices M_1 and M_2. We use $M^{[k]}$ to denote the kth Kronecker power of the matrix M, given by $M^{[0]} = 1$ and $M^{[k]} = M^{[k-1]} \otimes M$ for $k > 0$. For a matrix M of random variables, we write $\mathbb{E}[M]$ for its expectation.

3 Carleman Linearization for Stochastic Polynomial Systems

In this section, we present the Carleman linearization technique to transform a finite-dimensional discrete-time stochastic polynomial system into an infinite-dimensional one, then take expectation to get a deterministic system that describes the evolution of all moments of the system state.

3.1 Discrete-Time Stochastic Polynomial Systems

We consider the following finite-dimensional discrete-time stochastic polynomial system

$$x(t+1) = \sum_{i=0}^{\nu} F_i(t) x^{[i]}(t), \quad t \in \mathbb{N},$$
$$x(0) = x_{\text{ini}}, \tag{1}$$

where $x(t) \in \mathbb{R}^n$ is the state vector and $F_i(t) \in \mathbb{R}^{n \times n^i}$, $i \in \{0, 1, \ldots, \nu\}$, are matrix-valued stochastic processes. More precisely, we assume given a probability space Ω, then x_{ini} (*resp.* $x(t)$, $F_i(t)$) is a measurable function from Ω to $\mathbb{R}^n$ (*resp.* $\mathbb{R}^n$, $\mathbb{R}^{n \times n^i}$). We consider the scenarios where all $F_i(t)$'s have known distributions.

Systems of the form (1) can be used to model dynamics with both additive and multiplicative noise terms. The vector $F_0(t) \in \mathbb{R}^n$ represents additive noise, as $x^{[0]}(t) = 1$, while the matrices $F_1(t), \ldots, F_\nu(t)$ characterize the effects of multiplicative noise. Then, we define

$$\mathfrak{F}(t) \triangleq \begin{bmatrix} F_0(t) & F_1(t) & \cdots & F_\nu(t) \end{bmatrix} \in \mathbb{R}^{n \times \sum_{i=0}^{\nu} n^i}, \tag{2}$$

which is a matrix with n rows and $\sum_{i=0}^{\nu} n^i$ columns.

We make the following assumptions concerning the coefficient matrices and the random initial state x_{ini}.

Assumption 3.1. *The vector x_{ini} and the matrices $\mathfrak{F}(t)$, $t \in \mathbb{N}$, are all independent.*

Assumption 3.2. *The matrices $\{\mathfrak{F}(t) \mid t \in \mathbb{N}\}$ are identically distributed.*

Note that Assumptions 3.1 and 3.2 are not overly restrictive and they hold in fairly general situations as we discuss in Section 7. Notice also that under Assumption 3.1, matrices $F_i(t)$ and $F_j(t)$ are still allowed to statistically depend on each other. Moreover, Assumption 3.2 allows us to obtain a "time-invariant" method to compute moments, further yielding computational advantage.

Example 1. Consider

$$\begin{aligned} x_1(t+1) &= a(t) x_1(t) x_2(t), & x_1(0) &= x_{\text{ini},1}, \\ x_2(t+1) &= a(t)(x_1(t) + x_2(t)), & x_2(0) &= x_{\text{ini},2}, \end{aligned}$$

where $x_{\text{ini},1}$, $x_{\text{ini},2}$, and $a(t)$, $t \in \mathbb{N}$, are independent and identically distributed random variables. By using $x(t) = \begin{bmatrix} x_1(t) & x_2(t) \end{bmatrix}^\mathsf{T}$, $F_0(t) = \begin{bmatrix} 0 & 0 \end{bmatrix}^\mathsf{T}$, $F_1(t) = \begin{bmatrix} 0 & 0 \\ a(t) & a(t) \end{bmatrix}$, and

$F_2(t) = \begin{bmatrix} 0 & a(t) & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$, we can rewrite the system as in the form of (1), i.e.,

$$x(t+1) = F_0 x^{[0]}(t) + F_1 x^{[1]}(t) + F_2 x^{[2]}(t). \tag{3}$$

Our objective is to approximate the moments of the state of (1), which we will use in Section 5 to compute probabilistic safety areas for the state $x(t)$ at a given time t. For Example 1, given j and t, our objective is to approximate the moment $\mathbb{E}[x^{[j]}(t)]$. Note that the initial moments $\mathbb{E}[x^{[j]}(0)]$ can be computed from the probability distribution of x_{ini}.

3.2 Carleman Linearization

We use Carleman linearization to obtain an infinite-dimensional linear system that describes the evolution of the Kronecker powers of the state vector $x(t)$. By defining

$$y_k(t) \triangleq \begin{bmatrix} x^{[0]}(t)^\mathsf{T} & x^{[1]}(t)^\mathsf{T} & \cdots & x^{[k]}(t)^\mathsf{T} \end{bmatrix}^\mathsf{T}, \tag{4}$$

we can rewrite the dynamical system (1) using (2) and (4) as

$$\begin{aligned} x(t+1) &= \begin{bmatrix} F_0(t) & \cdots & F_\nu(t) \end{bmatrix} \begin{bmatrix} x^{[0]}(t)^\mathsf{T} & \cdots & x^{[\nu]}(t)^\mathsf{T} \end{bmatrix}^\mathsf{T} \\ &= \mathfrak{F}(t) \, y_\nu(t). \end{aligned}$$

Therefore, for all $j \in \mathbb{N}$, we have

$$x^{[j]}(t+1) = (\mathfrak{F}(t) \, y_\nu(t))^{[j]}. \tag{5}$$

Then, we introduce Theorem 3.3 to compute $(\mathfrak{F}(t) \, y_\nu(t))^{[j]}$.

Theorem 3.3. *Consider $\mathfrak{F}(t)$ and $y_\nu(t)$ given in equations* (2) *and* (4), *respectively. We have*

$$\begin{aligned} &\left(\mathfrak{F}(t) \, y_\nu(t) \right)^{[j]} \\ &\quad = \sum_{k=0}^{j\nu} \left(\sum_{(i_1, \ldots, i_l) \in H_{j,k}} F_{i_1}(t) \otimes \cdots \otimes F_{i_j}(t) \right) x^{[k]}(t), \end{aligned}$$

for $j \in \mathbb{N}$, where

$$H_{j,k} \triangleq \left\{ (i_1, \ldots, i_j) \, \middle| \, \sum_{l=1}^{j} i_l = k \text{ and } 0 \le i_l \le \nu \right\}.$$

4

The proof of Theorem 3.3 is presented in Appendix A.

By (5) and Theorem 3.3, we obtain the infinite-dimensional linear system

$$x^{[j]}(t+1) = \sum_{k=0}^{j\nu} \left(\sum_{(i_1,\ldots,i_j)\in H_{j,k}} F_{i_1}(t) \otimes \cdots \otimes F_{i_j}(t) \right) x^{[k]}(t),$$
$$x^{[j]}(0) = x_{\text{ini}}^{[j]}, \tag{6}$$

which describes the evolution of all Kronecker powers $x^{[0]}(t), x^{[1]}(t), \ldots$ of the state $x(t)$.

In order to give a simpler description of the system, we introduce the matrices $A_{j,k}(t) \in \mathbb{R}^{n^j \times n^k}$ given by

$$A_{j,k}(t) \triangleq \sum_{(i_1,\ldots,i_j)\in H_{j,k}} F_{i_1}(t) \otimes \cdots \otimes F_{i_j}(t). \tag{7}$$

Note in particular that $H_{0,0}$ only contains the empty tuple and $A_{0,0}(t) = 1$, as it is the identity of $\otimes$. We also have

$$A_{j,k}(t) = 0 \text{ if } k > j\nu, \tag{8}$$

since $H_{j,k}$ is then empty. We then introduce, for all $N, M \in \mathbb{N}$, the matrix $\mathfrak{A}_{N,M}(t)$ defined by blocks as:

$$\mathfrak{A}_{N,M}(t) \triangleq \begin{bmatrix} A_{0,0}(t) & \cdots & A_{0,M}(t) \\ \vdots & \ddots & \vdots \\ A_{N,0}(t) & \cdots & A_{N,M}(t) \end{bmatrix}, \tag{9}$$

which is a matrix of $\sum_{i=0}^{N} n^i$ rows and $\sum_{i=0}^{M} n^i$ columns. From (6), for any $k \in \mathbb{N}$, we have

$$\begin{bmatrix} x^{[0]}(t+1) \\ \vdots \\ x^{[k]}(t+1) \end{bmatrix} = \begin{bmatrix} A_{0,0}(t) & \cdots & A_{0,k\nu}(t) \\ \vdots & \ddots & \vdots \\ A_{k,0}(t) & \cdots & A_{k,k\nu}(t) \end{bmatrix} \begin{bmatrix} x^{[0]}(t) \\ \vdots \\ x^{[k]}(t) \\ \vdots \\ x^{[k\nu]}(t) \end{bmatrix},$$

which can be rewritten using (4) and (9) as

$$y_k(t+1) = \mathfrak{A}_{k,k\nu}(t)\, y_{k\nu}(t). \tag{10}$$

Notice that is not closed: we need $y_{k\nu}(t)$ in order to compute $y_k(t+1)$. As a result, the linear model describing the system cannot be written using a finite state space. However, in Section 4, we can restrict this system to a finite dimensional one as we are interested in a finite number of moments on a finite time horizon.

Example 2. For the system (3) in Example 1,

$$x^{[2]}(t+1) = \left(F_0(t)x^{[0]}(t) + F_1(t)x^{[1]}(t) + F_2(t)x^{[2]}(t)\right)^{[2]}.$$

By (10), we have

$$y_2(t+1) = \mathfrak{A}_{2,4}(t)y_4(t), \tag{11}$$

where

$$\mathfrak{A}_{2,4}(t) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ F_0 & F_1 & F_2 & 0 & 0 \\ F_0^{[2]} & \begin{matrix}F_0\otimes F_1 \\ +F_1\otimes F_0\end{matrix} & \begin{matrix}F_0\otimes F_2+F_1^{[2]} \\ +F_2\otimes F_0\end{matrix} & \begin{matrix}F_1\otimes F_2 \\ +F_2\otimes F_1\end{matrix} & F_2^{[2]} \end{bmatrix}.$$

Note that we omitted time dependence of F_0, F_1 and F_2 for brevity. Since $F_0(t) = \begin{bmatrix} 0 & 0 \end{bmatrix}^\mathsf{T}$, (11) can also be written as

$$\begin{bmatrix} 1 \\ x^{[1]}(t+1) \\ x^{[2]}(t+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & F_1 & F_2 & 0 & 0 \\ 0 & 0 & F_1^{[2]} & \begin{matrix}F_1\otimes F_2 \\ +F_2\otimes F_1\end{matrix} & F_2^{[2]} \end{bmatrix} \begin{bmatrix} 1 \\ x^{[1]}(t) \\ x^{[2]}(t) \\ x^{[3]}(t) \\ x^{[4]}(t) \end{bmatrix}.$$

Recall that our objective is to approximate $\mathbb{E}[x^{[j]}(t)]$ for given j and t. In the next section, we will consider a vector $\mathbb{E}[y_k(t)]$ which includes all moments $\mathbb{E}[x^{[j]}(t)]$ where $j \leq k$.

3.3 Moment Equations

We now derive the deterministic system that describes the evolution of the moments of $x(t)$ by taking expectation in (10). This gives

$$\mathbb{E}[y_k(t+1)] = \mathbb{E}[\mathfrak{A}_{k,k\nu}(t)y_{k\nu}(t)].$$

By iteration of (10), we get that $y_{k\nu}(t)$, $t \in \mathbb{N}$, is given by

$$y_{k\nu}(t) = \mathfrak{A}_{k\nu,k\nu^2}(t-1)\cdots\mathfrak{A}_{k\nu^t,k\nu^{t+1}}(0) \cdot y_{k\nu^{t+1}}(0). \tag{12}$$

It follows from Assumption 3.1 that $\mathfrak{A}_{k,k\nu}(t)$ and $y_{k\nu}(t)$ in (10) are mutually independent. To see this, observe that $\mathfrak{A}_{k,k\nu}(t)$ is composed of the matrices $F_i(t)$, which are independent of x_{ini} and $\mathfrak{F}(t-1), \ldots, \mathfrak{F}(0)$, which determine $y_{k\nu}(t)$ as given by (12). It then follows that

$$\mathbb{E}[y_k(t+1)] = \mathbb{E}[\mathfrak{A}_{k,k\nu}(t)]\mathbb{E}[y_{k\nu}(t)].$$

Here again, to give a simpler description of the system, we introduce new matrices. Notice that the coefficients of the

matrix $\mathbb{E}[\mathfrak{A}_{N,M}(t)]$ are products of the moments of coefficients of $\mathfrak{F}(t)$ and thus independent of t by Assumption 3.2.

Thereby, we use the notations:

$$E_{i,j} = \mathbb{E}[A_{i,j}(t)] \in \mathbb{R}^{n^i \times n^j},$$
$$\mathfrak{E}_{N,M} = \mathbb{E}[\mathfrak{A}_{N,M}(t)] \in \mathbb{R}^{\sum_{i=0}^{N} n^i \times \sum_{i=0}^{M} n^i}, \tag{13}$$

emphasizing the fact that both are independent of the time.

Then, from (6), we can obtain the following system.

$$\mathbb{E}[x^{[j]}(t+1)] = \sum_{k=0}^{j\nu} E_{j,k}\mathbb{E}[x^{[k]}(t)], \quad t \in \mathbb{N}, \tag{14}$$
$$\mathbb{E}[x^{[j]}(0)] = \mathbb{E}[x_{\mathrm{ini}}^{[j]}].$$

From (7), the matrices $E_{j,k}$ only depend on the moments $\mathbb{E}[F_i(\cdot)]$, which are independent of t as discussed above. As a result, the matrices $E_{j,k}$ can be computed offline.

4 Moment Approximation through Truncation

We will now introduce an approach that allows us to compute *approximations* of the moments of $x(t)$. This truncation approach is critical, as an exact computation of the moments is impossible from a practical point of view. Indeed, by (4), computing the first k moments at time t amounts to computing $\mathbb{E}[y_k(t)]$. So, by iteration of (14),

$$\mathbb{E}[y_k(t+1)] = \mathfrak{E}_{k,k\nu}\cdots\mathfrak{E}_{k\nu^{t-1},k\nu^t}\mathbb{E}[y_{k\nu^t}(0)]. \tag{15}$$

As this indicates, exact computation of the first k moments requires the knowledge of matrices $\mathfrak{E}_{k,k\nu}$, $\mathfrak{E}_{k\nu,k\nu^2}$, ..., $\mathfrak{E}_{k\nu^{t-1},k\nu^t}$ of exponentially increasing dimensions, making any practical computation unrealistic.

Thus, in the next section, we compute approximations of the moments of $x(t)$ by trucating the system (14).

4.1 Approximate Moments and the Truncated System

In this section, we define the system that we use to compute approximations of the moments of $x(t)$.

We fix the truncation limit $N_\mathrm{T} \in \mathbb{N}$, and define approximate moments $\widetilde{x}^{(i)}(t) \in \mathbb{R}^{n^i}$, $i \in 1, \ldots, N_\mathrm{T}$, by

$$\left[1 \ \widetilde{x}^{(1)}(t)^{\mathsf{T}} \ \cdots \ \widetilde{x}^{(N_\mathrm{T})}(t)^{\mathsf{T}} \right]^{\mathsf{T}}$$
$$= \mathfrak{E}_{N_\mathrm{T},N_\mathrm{T}}^t \left[1 \ \mathbb{E}[x_{\mathrm{ini}}]^{\mathsf{T}} \ \cdots \ \mathbb{E}[x_{\mathrm{ini}}^{[N_\mathrm{T}]}]^{\mathsf{T}} \right]^{\mathsf{T}}. \tag{16}$$

Notice that (16) follows the same pattern as (15), but use the square matrix $\mathfrak{E}_{N_\mathrm{T},N_\mathrm{T}}$. Here, the vector $\widetilde{x}^{(i)}(t)$ represents an approximation of the moment $\mathbb{E}[x^{[i]}(t)]$ that is computed using only our knowledge of the first N_T moments of x_{ini}. Note that the superscript (i) is only for notation purpose, and it has no relation to the Kronecker power.

By letting

$$\widetilde{y}(t) \triangleq \left[1 \ \widetilde{x}^{(1)}(t)^{\mathsf{T}} \ \widetilde{x}^{(2)}(t)^{\mathsf{T}} \ \cdots \ \widetilde{x}^{(N_\mathrm{T})}(t)^{\mathsf{T}} \right]^{\mathsf{T}}, \tag{17}$$

we obtain what we call the "truncated system", which is a discrete-time linear time-invariant system given by

$$\widetilde{y}(t+1) = \mathfrak{E}_{N_\mathrm{T},N_\mathrm{T}}\widetilde{y}(t), \quad t \in \mathbb{N},$$
$$\widetilde{y}(0) = \left[1 \ \mathbb{E}[x_{\mathrm{ini}}(t)]^{\mathsf{T}} \ \cdots \ \mathbb{E}[x_{\mathrm{ini}}^{[N_\mathrm{T}]}(t)]^{\mathsf{T}} \right]^{\mathsf{T}}. \tag{18}$$

The truncated system allows us to iteratively compute approximations of the moments of $x(t)$ at consecutive time instants. Moreover, the approach only requires an offline computation of the matrix $\mathfrak{E}_{N_\mathrm{T},N_\mathrm{T}}$.

Example 3. By considering $N_\mathrm{T} = 2$, we may approximate the first and and the second moments of the system in Example 2 by the following truncated system.

$$\widetilde{y}(t+1) = \left[1 \ \widetilde{x}^{(1)}(t+1)^{\mathsf{T}} \ \widetilde{x}^{(2)}(t+1)^{\mathsf{T}} \right]^{\mathsf{T}}$$
$$= \mathbb{E}\left[\begin{bmatrix} 1 & 0 & 0 \\ 0 & F_1(t) & F_2(t) \\ 0 & 0 & F_1(t) \otimes F_1(t) \end{bmatrix} \right] \begin{bmatrix} 1 \\ \widetilde{x}^{(1)}(t)^{\mathsf{T}} \\ \widetilde{x}^{(2)}(t)^{\mathsf{T}} \end{bmatrix}$$
$$= \mathfrak{E}_{2,2}\,\widetilde{y}(t),$$

where $\widetilde{x}^{(1)}(0)^{\mathsf{T}} = \mathbb{E}[x_{\mathrm{ini}}(t)]^{\mathsf{T}}$ and $\widetilde{x}^{(2)}(0)^{\mathsf{T}} = \mathbb{E}[x_{\mathrm{ini}}^{[2]}(t)]^{\mathsf{T}}$.

4.2 Computation of Truncation Errors

We now consider the error due to the truncation. We consider $j_0 \in \{0, \ldots, N_\mathrm{T}\}$, and let $e^{(j_0)}(t) \in \mathbb{R}^{n^{j_0}}$ denote the error of the j_0-th moment, that is,

$$e^{(j_0)}(t) \triangleq \mathbb{E}[x^{[j_0]}(t)] - \widetilde{x}^{(j_0)}(t). \tag{19}$$

First, by (14), we have

$$\mathbb{E}[x^{[j_0]}(t)] = \sum_{j_1=0}^{j_0\nu} E_{j_0,j_1}\mathbb{E}[x^{[j_1]}(t-1)]$$
$$= \sum_{j_1=0}^{j_0\nu} E_{j_0,j_1} \sum_{j_2=0}^{j_1\nu} E_{j_1,j_2}\mathbb{E}[x^{[j_2]}(t-2)]$$
$$= \sum_{j_1=0}^{j_0\nu} E_{j_0,j_1} \sum_{j_2=0}^{j_1\nu} E_{j_1,j_2} \cdots \sum_{j_t=0}^{j_{t-1}\nu} E_{j_{t-1},j_t}\mathbb{E}[x_{\mathrm{ini}}^{[j_t]}],$$

and similarly, by (16),

$$\widetilde{x}^{(j_0)}(t) = \sum_{j_1=0}^{N_\mathrm{T}} E_{j_0,j_1} \sum_{j_2=0}^{N_\mathrm{T}} E_{j_1,j_2} \cdots \sum_{j_t=0}^{N_\mathrm{T}} E_{j_{t-1},j_t} \mathbb{E}[x_\mathrm{ini}^{[j_t]}].$$

From (8) and (13), we can observe that $E_{j,k} = 0$ for $k > j\nu$. Therefore, we obtain from the two equations above,

$$
\begin{aligned}
e^{(j_0)}(t) &= \mathbb{E}[x^{[j_0]}(t)] - \widetilde{x}^{(j_0)}(t) \\
&= \sum_{j_1=N_\mathrm{T}+1}^{j_0\nu} E_{j_0,j_1} \sum_{j_2=0}^{j_1\nu} E_{j_1,j_2} \cdots \sum_{j_t=0}^{j_{t-1}\nu} E_{j_{t-1},j_t} \mathbb{E}[x_\mathrm{ini}^{[j_t]}] \\
&\quad + \sum_{j_1=0}^{N_\mathrm{T}} E_{j_0,j_1} \sum_{j_2=N_\mathrm{T}+1}^{j_1\nu} E_{j_1,j_2} \cdots \sum_{j_t=0}^{j_{t-1}\nu} E_{j_{t-1},j_t} \mathbb{E}[x_\mathrm{ini}^{[j_t]}] \\
&\quad + \cdots \\
&\quad + \sum_{j_1=0}^{N_\mathrm{T}} E_{j_0,j_1} \cdots \sum_{j_{t-1}=0}^{N_\mathrm{T}} E_{j_{t-2},j_{t-1}} \sum_{j_t=N_\mathrm{T}+1}^{j_{t-1}\nu} E_{j_{t-1},j_t} \mathbb{E}[x_\mathrm{ini}^{[j_t]}]. \quad (20)
\end{aligned}
$$

As an immediate consequence, we get the following:

Proposition 4.1. *Consider the truncated approximation of the moments of system* (1) *with truncation limit* $N_\mathrm{T} \in \mathbb{N}$. *If* $j_0\nu^t \leq N_\mathrm{T}$, *then* $\widetilde{x}^{(j_0)}(t) = \mathbb{E}[x^{[j_0]}(t)]$.

Proof. We show that, if $j_0\nu^t \leq N_\mathrm{T}$, then $e^{(j_0)}(t) = 0$ and the proposition holds. To this end, it is enough to show that all lines of (20) are equal to 0 for any sequence $j_1, \ldots, j_t$ of relevant indices. Let us pick any ith line and any sequence $j_1, \ldots, j_t$. It is enough to show that there exists $k \in \{1, \ldots t\}$ where $j_k > j_{k-1}\nu$, so that $E_{j_{k-1},j_k} = 0$ by (8), which also makes the ith line equal to 0. For the sake of contradiction, we assume that $j_k \leq j_{k-1}\nu$ for all $k \in \{1, \ldots t\}$, which makes $j_{i-1} \leq j_{i-2}\nu \leq j_{i-3}\nu^2 \leq \ldots \leq j_0\nu^{i-1} \leq j_0\nu^t$. However, notice that we have $j_{i-1} \geq N_\mathrm{T} + 1 > N_\mathrm{T} \geq j_0\nu^t$, which is a contradiction that proves the desired result. $\square$

This shows that, for large values of truncation limit N_T, the proposed method computes *exact* moments $\mathbb{E}[x^{[j_0]}(t)]$ for small enough j_0 and t. Note that this is due to the *discrete-time* nature of the finite-dimensional polynomial system (1). In the continuous-time case, approximation errors cannot be avoided in general (Forets and Pouly, 2017).

Since $\mathbb{E}[x^{[j_0]}(t)] = \widetilde{x}^{(j_0)}(t) + e^{(j_0)}(t)$, if $e^{(j_0)}(t)$ could efficiently be computed, then so would $\mathbb{E}[x^{[i]}(t)]$, and there would be no need in using the truncated system. However, the exact value of $e^{(j_0)}(t)$ is generally hard to compute. Therefore, in the following section, we provide upper bounds for $e^{(j_0)}(t)$.

4.3   Approximation of Error Bounds

We now investigate the approximation error introduced by truncation. For a given $j_0 \in \{0, \ldots, N_\mathrm{T}\}$, our goal is to obtain an upper bound on $\left\| e^{(j_0)}(t) \right\|_\infty$, which is the error on the j_0-th moment introduced by the truncation (where $\|\cdot\|_\infty$ denotes the infinity norm).

Bounds on $\left\| e^{(i)}(t) \right\|_\infty$ allow us to use various techniques to study the distribution of the state at future time steps. We illustrate this in Section 5 by using tail probability approximations to compute a safety area, for which we know that the probability of the system landing outside that area is bounded by a predefined constant.

4.3.1   Global Error Bound

First, let $\xi(t) \triangleq \max\limits_{0 \leq j \leq j_0\nu^t} \left\| \mathbb{E}[x_\mathrm{ini}^{[j]}] \right\|_\infty$. Using $\xi(t)$, we can derive bounds for $e^{(j_0)}(t)$ by first reorganising (20) according to the moments of x_ini, which gives

$$e^{(j_0)}(t) = \sum_{j=0}^{j_0\nu^t} \widetilde{E}_j \mathbb{E}[x_\mathrm{ini}^{[j]}], \qquad (21)$$

where each $\widetilde{E}_j \in \mathbb{R}^{n^{j_0} \times n^j}$ is a simple sum of products of $E_{n,m}$'s, obtained by this reorganisation of (20). All $\widetilde{E}_j$ can thus be computed offline (note that $\widetilde{E}_j$ is dependent on t, but we keep this implicit for readability).

From this, we can derive a global bound

$$\left\| e^{(j_0)}(t) \right\|_\infty \leq \xi(t) \sum_{j=0}^{j_0\nu^t} \left\| \widetilde{E}_j \right\|_\infty, \qquad (22)$$

where $\sum_{j=0}^{j_0\nu^t} \left\| \widetilde{E}_j \right\|_\infty$ can be computed offline.

Observe that $\xi(t)$ can be efficiently computed in some cases. An obvious situation is when the position x_ini is determined, in which case we have $\xi(t) = \max\{1, \|x_\mathrm{ini}\|_\infty^{j_0\nu^t}\}$. Another case is when x_ini obeys a well-known distribution whose moments are easy to compute, such as uniform or normal distributions. Another case is when the system satisfies $x(t) \in \mathbb{R}$ and $x_\mathrm{ini} \in [0, 1]$, in which case $\left\| \mathbb{E}[x_\mathrm{ini}^{[j]}] \right\|_\infty$ is decreasing and we have $\xi(t) = \left\| \mathbb{E}[x_\mathrm{ini}^{[0]}] \right\|_\infty = 1$.

4.3.2 Error Bound using Partial Exact Computation with Block Indices on Moments

We can further refine (21) to consider a single line $i \leq n^{j_0}$ of the equation, for which we get

$$e_i^{(j_0)}(t) = \sum_{j=0}^{j_0 \nu^t} v_{j,i} \mathbb{E}[x_{\text{ini}}^{[j]}], \tag{23}$$

where $v_{j,i} \in \mathbb{R}^{1 \times n^j}$ is the ith row of $\widetilde{E}_j$. By repeated application of triangle and Cauchy-Schwarz inequalities, we have

$$|e_i^{(j_0)}(t)| \leq \xi(t) \sum_{j=0}^{j_0 \nu^t} \|v_{j,i}\|_\infty, \tag{24}$$

where $\sum_{j=0}^{j_0 \nu^t} \|v_{j,i}\|_\infty$ can also be computed offline.

This bound can, however, be crude in practice as the norm gets distributed over all sums and products. We alleviate this problem in Proposition 4.2 where we compute tighter bounds while maintaining a reasonable computational cost.

Proposition 4.2. *For any subset $J \subseteq \{0, \ldots, j_0 \nu^t\}$, we have the bound*

$$|e_i^{(j_0)}(t)| \leq \Big| \sum_{j \in J} v_{j,i} \mathbb{E}[x_{\text{ini}}^{[j]}] \Big| + \xi(t, J) \sum_{j \notin J} \|v_{j,i}\|_\infty,$$

where $\xi(t, J) = \max_{j \notin J} \left\| \mathbb{E}[x_{\text{ini}}^{[j]}] \right\|_\infty$.

The idea is that J is a set of indices where one should avoid distributing the norm over the sum. One should pick J to consist of those indices where the distribution is too crude and makes the error bound loose. In order for this method to be computationally efficient, one should pick J that is of relatively small size, e.g., $|J| = \mathcal{O}(t)$. One possible way to choose J is to fix a size $\widehat{j}$ and return the set of $\widehat{j}$ indices j where $\|v_{j,i}\|$ are the largest; another way to return the set of $\widehat{j}$ indices j such that $\left\| \mathbb{E}[x_{\text{ini}}^{[j]}] \right\|_\infty$ are the largest. For example, suppose x_{ini} is drawn from a truncated normal distribution over the interval $[0, 1]$. Then $\xi(t) = 1$ and $\xi(t, J) = \left\| \mathbb{E}[x_{\text{ini}}^{[j_{\min}]}] \right\|_\infty$, where $j_{\min}$ is the smallest number that is not in J. If J is chosen as the set of first $\widehat{j}$ indices, then $J = \{0, \ldots, \widehat{j} - 1\}$ and $\xi(t, J) = \left\| \mathbb{E}[x_{\text{ini}}^{[\widehat{j}]}] \right\|_\infty$.

4.3.3 Error Bound using Partial Exact Computation with Regular Indices on Moment Coordinates

We can further refine (21) and (23) by considering each element of matrix $\widetilde{E}_j$ and vector $\mathbb{E}[x_{\text{ini}}^{[j]}]$. Let $\tilde{v} =$

$\left[\widetilde{E}_0 \;\; \widetilde{E}_1 \;\; \cdots \;\; \widetilde{E}_{j_0 \nu^t} \right]$ and $\tilde{y} = \mathbb{E}\left[x_{\text{ini}}^{[0]\mathsf{T}} \;\; x_{\text{ini}}^{[1]\mathsf{T}} \;\; \cdots \;\; x_{\text{ini}}^{[j_0 \nu^t]\mathsf{T}} \right]^\mathsf{T}$. In the same way as in (24), we have

$$|e_i^{(j_0)}(t)| \leq \sum_{k=0}^{m} |\tilde{v}_{i,k}| |\tilde{y}_k| \leq \max_{k \leq m} |\tilde{y}_k| \cdot \sum_{k=0}^{m} |\tilde{v}_{i,k}|,$$

where $\tilde{y}_k$ is the kth row of $\tilde{y}$ and and $\tilde{v}_{i,k}$ is the element at ith row and kth column of $\tilde{v}$. Then, in the same way as in Proposition 4.2, we obtain the following proposition.

Proposition 4.3. *For any subset $K \subseteq \{0, \ldots, m\}$, we have*

$$|e_i^{(j_0)}(t)| \leq \Big| \sum_{k \in K} \tilde{v}_{i,k} \tilde{y}_k \Big| + \max_{k \notin K} |\tilde{y}_k| \sum_{k \notin K} |\tilde{v}_{i,k}|.$$

Again, one possible option of the set K is to fix a size $\widehat{k}$ and then choose $\widehat{k}$ indices k where $|\tilde{y}_k|$ are the largest. In the next section, we will use these error bounds for probabilistic safety analysis.

5 Ellipsoid Bounds for Probabilistic Safety Analysis

Everything that we have computed up to this point can be computed offline, as it does not depend on the actual system state. In this section, we show how to do online computation of probabilistic safety regions using tail probability analysis. It crucially relies on approximations of the first and second moments of the dynamics of the system.

5.1 Tail Probability Approximation

In this section, we use the bounds on the error introduced by the truncated system, which are derived in Section 4.3, to give a lower bound on the probability of the system being inside a given ellipsoid region after t time steps. For any $j_0 \leq N_{\text{T}}$ and $i \leq n^{j_0}$, let $\epsilon_i^{(j_0)}(t)$ be an upper bound of $|e_i^{(j_0)}(t)|$ obtained by any of the methods in Section 4.3.

We define the region we are interested in terms of positive-semidefinite matrices. A matrix $P \in \mathbb{R}^{n \times n}$ is positive-semidefinite if, for all vectors $x \in \mathbb{R}^n$, $x^\mathsf{T} P x \geq 0$. Such a P defines a seminorm, called the P-seminorm, by $\|x\|_P = (x^\mathsf{T} P x)^{1/2}$. The region defined by $\|x\|_P \leq r$ is an ellipsoid (possibly of infinite radius in some dimensions).

Recall that n is the dimension of the state vector $x(t)$ of the system (1). Let

$$x(t) = \left[x_1(t) \; x_2(t) \; \ldots \; x_n(t) \right]^\mathsf{T} \in \mathbb{R}^n,$$

where x_i's are scalars. Notice that, for any $i, j \in \{1, \ldots, n\}$, the $(ni + j)$-th row of $x^{[2]}(t)$ is

$$x_{ni+j}^{[2]}(t) = x_i(t) x_j(t).$$

8

Hence, we can approximate the expectations $\mathbb{E}[x_i(t)x_j(t)]$ using $\widetilde{x}^{(2)}(t)$, which is the approximation of $\mathbb{E}[x^{[2]}(t)]$ computed with the methods in Section 4.3, as well as its error bound $\epsilon^{(2)}(t)$. For the sake of readability, we use the notation

$$\widetilde{x}^{(2)}_{(i,j)}(t) = \widetilde{x}^{(2)}_{ni+j}(t),$$

as it is the approximation of $\mathbb{E}[x_i(t)x_j(t)]$. Similarly, we use

$$\epsilon^{(2)}_{(i,j)}(t) = \epsilon^{(2)}_{ni+j}(t),$$

which is the respective approximation error bound. We also use $\widetilde{x}^{(1)}_i(t)$ and $\epsilon^{(2)}_i(t)$ to denote the i-th row of $\widetilde{x}^{(1)}(t)$ and $\epsilon^{(1)}(t)$, respectively.

Now, we assume that we know a global error bound

$$\|\widetilde{x}^{(1)}(t) - \mathbb{E}[x(t)]\|_P \leq \varepsilon_{P,t}. \tag{25}$$

We will discuss the technique to get this bound $\varepsilon_{P,t}$ later at the end of this section.

We then introduce Proposition 5.1, which gives a mean to bound the probability of the system being outside of an ellipsoid centered around $\widetilde{x}^{(1)}(t)$.

Proposition 5.1. *For given positive-semidefinite matrix P and $\alpha > \varepsilon_{P,t}$,*

$$\mathbb{P}(\|x(t) - \widetilde{x}^{(1)}(t)\|_P \geq \alpha)$$
$$\leq \frac{\sum_{i,j=1}^n p_{ij}(\mathbb{E}[x_i(t)x_j(t)] - \mathbb{E}[x_i(t)]\mathbb{E}[x_j(t)])}{(\alpha - \varepsilon_{P,t})^2}, \tag{26}$$

where p_{ij} is the element on the ith row and the jth column of the matrix P.

Proof. Since $\alpha > \varepsilon_{P,t}$, we get by Markov's inequality:

$$\mathbb{P}(\|x(t) - \widetilde{x}^{(1)}(t)\|_P \geq \alpha)$$
$$\leq \mathbb{P}(\|x(t) - \mathbb{E}[x(t)]\|_P \geq \alpha - \varepsilon_{P,t})$$
$$= \mathbb{P}(\|x(t) - \mathbb{E}[x(t)]\|_P^2 \geq (\alpha - \varepsilon_{P,t})^2)$$
$$\leq \frac{\mathbb{E}[\|x(t) - \mathbb{E}[x(t)]\|_P^2]}{(\alpha - \varepsilon_{P,t})^2}$$
$$= \frac{\sum_{i,j=1}^n p_{ij}(\mathbb{E}[x_i(t)x_j(t)] - \mathbb{E}[x_i(t)]\mathbb{E}[x_j(t)])}{(\alpha - \varepsilon_{P,t})^2}. \quad \square$$

For a fixed matrix P and known values of the approximations $\widetilde{x}^{(1)}_i(t)$, $\widetilde{x}^{(2)}_{(i,j)}(t)$, and the error bounds $\epsilon^{(1)}_i(t)$ and $\epsilon^{(2)}_{(i,j)}(t)$, each term of the sum in (26) can be bounded. The derived bound depends on the signs of p_{ij}, $\epsilon^{(1)}_i(t) - \widetilde{x}^{(1)}_i(t)$, and

$\epsilon^{(1)}_j(t) - \widetilde{x}^{(1)}_j(t)$. For example, if $p_{ij} > 0$, $\widetilde{x}^{(1)}_i(t) \geq \epsilon^{(1)}_i(t)$, and $\widetilde{x}^{(1)}_j(t) \geq \epsilon^{(1)}_j(t)$, then we have

$$p_{ij}(\mathbb{E}[x_i(t)x_j(t)] - \mathbb{E}[x_i(t)]\mathbb{E}[x_j(t)]) \leq$$
$$p_{ij}(\widetilde{x}^{(2)}_{(i,j)}(t) + \epsilon^{(2)}_{(i,j)}(t) - \tag{27}$$
$$(\widetilde{x}^{(1)}_i(t) - \epsilon^{(1)}_i(t))(\widetilde{x}^{(1)}_j(t) - \epsilon^{(1)}_j(t))).$$

Similar bounds can easily be found in all other cases by using the following bounds:

$$|\mathbb{E}[x_i(t)x_j(t)] - \widetilde{x}^{(2)}_{(i,j)}(t)| \leq \epsilon^{(2)}_{(i,j)}(t),$$
$$|\mathbb{E}[x_i(t)] - \widetilde{x}^{(1)}_i(t)| \leq \epsilon^{(1)}_i(t).$$

In particular, if we know exact values for $\mathbb{E}[x_i(t)]$ and $\mathbb{E}[x_i(t)x_j(t)]$ (i.e., if $N_T \geq 2\nu^t$), we have Corollary 5.2.

Corollary 5.2. *Consider a positive-semidefinite matrix P. If $N_T \geq 2\nu^t$, for any $\alpha > \varepsilon_{P,t}$,*

$$\mathbb{P}(\|x(t) - \widetilde{x}^{(1)}(t)\|_P \geq \alpha)$$
$$\leq \frac{\sum_{i,j=1}^n p_{ij}(\widetilde{x}^{(2)}_{(i,j)}(t) - \widetilde{x}^{(1)}_i(t)\widetilde{x}^{(1)}_j(t))}{(\alpha - \varepsilon_{P,t})^2}, \tag{28}$$

where p_{ij} is the element on the ith row and the jth column of the matrix P.

One part that we have left open is how to compute a value for $\varepsilon_{P,t}$. In Section 4.3, we have given bounds on $\|\mathbb{E}[x(t)] - \widetilde{x}^{(1)}(t)\|_\infty$. This directly gives us bounds on $\|\mathbb{E}[x(t)] - \widetilde{x}^{(1)}(t)\|_P$, using the fact that $\|x\|_P \leq \lambda_n^{1/2}\|x\|_\infty$, where λ_n is the greatest eigenvalue of P. Therefore, $\varepsilon_{P,t} \leq \lambda_n^{1/2} \|e^{(1)}(t)\|_\infty$. Recall that we can compute an upper bound of $\|e^{(1)}(t)\|_\infty$ using methods in Section 4.3.

5.2 Computation of Probabilistic Ellipsoid Bounds

Using the bounds above, we explain how to compute probabilistic ellipsoid bounds, i.e., ellipsoid areas in which we know the system will be with at least a given probability.

The problem we are interested in is the following: given a probabilistic system as in (1), approximations $\widetilde{x}^{(1)}(t)$ and $\widetilde{x}^{(2)}(t)$ of first and second moments of the system at time t, bounds on the errors of these approximations, and a constant $b \in (0, 1)$, find an ellipsoid in which we know the system state $x(t)$ will be with probability at least $(1 - b)$. The ellipsoid is preferred to be as small as possible, to give a precise bound. The problem to find the smallest ellipsoid

can be formulated as the following optimization problem:

$$\underset{P}{\text{maximize}} \quad \det(P)$$

subject to P is positive-definite,
$$\frac{\sum_{i,j=1}^{n} p_{i,j}(\mathbb{E}[x_i(t)x_j(t)] - \mathbb{E}[x_i(t)]\mathbb{E}[x_j(t)])}{(\alpha - \varepsilon_{P,t})^2} \leq b, \tag{29}$$

for some fixed $\alpha > 0$, say 1, and $p_{i,j}$ is the element on the ith row and the jth column of P. From Proposition 5.1, we have that the system state at time t is in the ellipsoid $\left\{ x \in \mathbb{R}^\nu \,\middle|\, \|x - \widetilde{x}^{(1)}(t)\|_P \leq \alpha \right\}$ with probability at least $(1 - b)$.

Remark 1. The problem (29) cannot be solved by convex optimization methods – and therefore not solved online – because $\varepsilon_{P,t}$ depends on P and the presence of $\varepsilon_{P,t}$ makes it unclear whether the constraint is convex. Moreover, computing $\varepsilon_{P,t}$ cannot be done online repeatedly.

Hence, we propose a three-step approximate solution for efficiency. We first solve the problem above, but assume that all errors are 0 to obtain the matrix Q:

$$\underset{Q}{\text{maximize}} \quad \det(Q)$$

subject to Q is positive-definite,
$$\frac{\sum_{i,j=1}^{n} q_{i,j}(\widetilde{x}^{(2)}_{(i,j)}(t) - \widetilde{x}^{(1)}_i(t)\widetilde{x}^{(1)}_j(t))}{\alpha^2} \leq b, \tag{30}$$

where $q_{i,j}$ is the element on the ith row and the jth column of Q. This makes the problem convex (Boyd and Vandenberghe, 2004), so (30) can be solved online. Note that changing the value of $\alpha > 0$ does not change the volume of the ellipsoid $\left\{ x \in \mathbb{R}^\nu \,\middle|\, \|x - \widetilde{x}^{(1)}(t)\|_Q \leq \alpha \right\}$, since the volume is inversely proportional to $\det(Q)$. So, we may use $\alpha = 1$.

Then, as a second step, we compute a matrix $P = sQ$ where s is a positive real value that is obtained by solving the following optimization problem.

$$\underset{s}{\text{maximize}} \quad \det(sQ)$$

subject to $s \in \mathbb{R}, s > 0$,
$$\frac{\sum_{i,j=1}^{n} s q_{i,j} u_{i,j}(\widetilde{x}^{(1)}(t), \widetilde{x}^{(2)}(t), \epsilon^{(1)}(t), \epsilon^{(2)}(t))}{\alpha^2} \leq b, \tag{31}$$

where $u_{i,j}$ is a function of the moment approximations and truncation error bounds that satisfies

$$\mathbb{E}[x_i(t)x_j(t)] - \mathbb{E}[x_i(t)]\mathbb{E}[x_j(t)]$$
$$\leq u_{i,j}(\widetilde{x}^{(1)}(t), \widetilde{x}^{(2)}(t), \epsilon^{(1)}(t), \epsilon^{(2)}(t)).$$

Notice that $u_{i,j}$ can always be computed as discussed in Section 5.1 (e.g., (27)).

The goal of this second step is to enlarge the ellipsoid of Q obtained from the first step by using a scalar value s. This
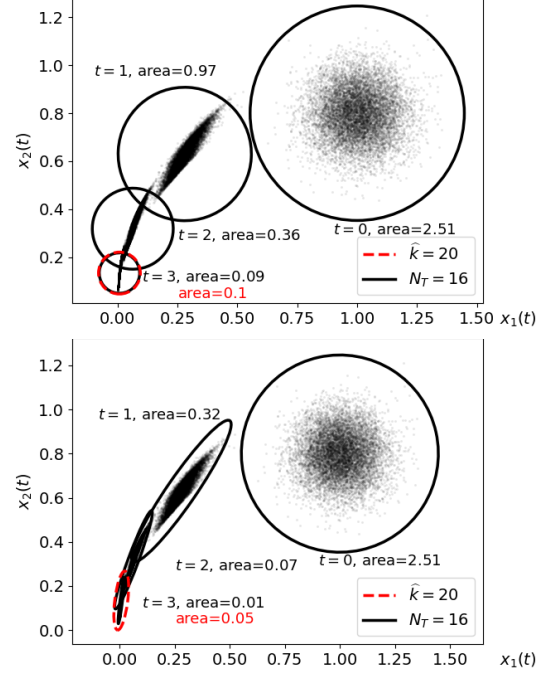

Fig. 2. Safe ball (top) and ellipsoid (bottom) bounds for safe regions with probability 0.9 for the system in Example 4. The black bounds are tight bounds computed using $N_\mathrm{T} = 16$. The red dashed bound is computed using an approximate dynamics from Proposition 4.3 with $N_\mathrm{T} = 8$ and $\widehat{k} = 20$.

results in a new ellipsoid $P = sQ$ that satisfies the constraint of (29) if $\varepsilon_{P,t} = 0$. Note that we could also consider an optimization problem for P that satisfies the constraint of (29) (if $\varepsilon_{P,t} = 0$) directly, but the problem has too many constraints and is difficult to solve efficiently in practice.

Let $P = sQ$ be obtained from solving (31). For the final step, we compute $\varepsilon_{P,t}$ or its over-approximation using the methods in the previous sections. Recall that, from (25), we have $\|\widetilde{x}^{(1)}(t) - \mathbb{E}[x(t)]\|_P \leq \varepsilon_{P,t}$. Hence, we enlarge the ellipsoid of P by using the bound $\varepsilon_{P,t}$, and take the ellipsoid

$$\left\{ x \in \mathbb{R}^\nu \,\middle|\, \|x - \widetilde{x}^{(1)}(t)\|_P \leq \alpha + \varepsilon_{P,t} \right\}. \tag{32}$$

By (25), Proposition 5.1, and (31), the system state $x(t)$ will be in the ellipsoid (32) with probability at least $(1 - b)$.

Example 4. Figure 2 shows the safe regions with probability 0.9 for the system in Example 1 at time steps $t \in \{0, 1, 2, 3\}$. More precisely, we defined $x_1(0)$ (*resp.* $x_2(0)$) as a Gaussian random variable with mean 1 (*resp.* 0.8) and standard deviation 0.1, and $a(t)$ as independent and identically distributed random variables following a uniform distribution on $[0.3, 0.4]$. The safe regions are plotted in comparison to 10000 Monte Carlo simulations. Both plots are computed as described above; however, the regions in the top one are computed by restricting to $p_{11} = p_{22}$ and $p_{12} = p_{21} = 0$. (i.e., ball-shaped regions). The black solid bounds are computed using $N_\mathrm{T} = 16$; therefore, they are tight bounds where

the error $e^{(1)}(t) = 0$ (see Proposition 4.1). The red dashed bound at $t = 3$ is computed using a truncated dynamics and the upper bound of the moment approximation errors as in Proposition 4.3, where $N_T = 8$ and the set K consists of $\widehat{k} = 20$ indices k where $|\tilde{y}_k|$ are the largest. Notice that the bound computed using the truncated dynamics (the red one) is an over-approximation of the tight bound (the black one). Notice also that the areas of the ellipsoids are smaller than those of the ball-shaped regions (e.g., at $t = 2$, the area of the ball is 0.36, while that of the ellipsoid is 0.07).

Remark 2. Note that we are in general not interested in optimizing all dimensions of P. For example, if a system has (x, y, θ) as coordinates, we may be only interested in finding a bound for (x, y). Maximizing $\det(P)$ under the constraints in (30) can lead to a P with a large unit ball in the θ dimension, but small in the x and y dimensions, while there may be a better P when considering just (x, y). If $I \subseteq \{1, \ldots, n\}$ is the set of dimensions of interest, our goal is to maximize $\det(P_I)$ under the constraints above, where P_I is the submatrix of P whose indices are in I.

6 Smaller matrices with reduced Kronecker powers

The main bottleneck of our method is the size of the matrix E we compute offline. One reason for this is due to duplications of computations. Indeed, the Kronecker power of a vector contains several times the same element. For example, the Kronecker square of the vector $x = \begin{bmatrix} a & b \end{bmatrix}^\mathsf{T}$ is given by $x^{[2]} = \begin{bmatrix} a^2 & ab & ba & b^2 \end{bmatrix}^\mathsf{T}$ and $ab = ba$ appears twice. In this section, we describe a *reduced Kronecker power* whose elements are the same as the normal Kronecker power, but without any duplication. For example, the reduced Kronecker square of x above will be $x^{\langle 2 \rangle} = \begin{bmatrix} a^2 & ab & b^2 \end{bmatrix}^\mathsf{T}$.

It should be noted that the notion of reduced Kronecker powers was also presented by Bellman (1970) and Carravetta *et al.* (1996) but used for different problems. In this work, we need to efficiently represent those reduced Kronecker powers, and also accommodate the previous sections with this notion. This requires the development of an operation corresponding to matrix multiplication (see (35) below) in order to propagate moments. Formally, we need to manipulate polynomials in a clever way, and this is one of our novelties compared to the literature.

Fix a vector $x = \begin{bmatrix} x_1 & \ldots & x_n \end{bmatrix}^\mathsf{T} \in \mathbb{R}^n$. Each element of a reduced Kronecker power of x will correspond to a n-tuple of natural number $(m_1, \ldots, m_n)$, this element being given by $x_1^{m_1} \ldots x_n^{m_n}$. The degree of such an n-tuple is given by the sum of its elements $\sum_{i=1}^{n} m_i$. Denote the set of n-tuples of degree m by $\mathcal{I}_{n,m}$. This set can be totally ordered by lexicographic order, that is, $(m_1, \ldots, m_n) < (m'_1, \ldots, m'_n)$ if there is k such that $m_k < m'_k$ and for all $j < k$, $m_j = m'_j$. For example, $\mathcal{I}_{2,2} = \{(2, 0) > (1, 1) > (0, 2)\}$.

The *m-th reduced Kronecker power of x* is then given by:

$$x^{\langle m \rangle} = \begin{bmatrix} x_1^{m_1} \ldots x_n^{m_n} \mid (m_1, \ldots, m_n) \in \mathcal{I}_{n,m} \end{bmatrix}^\mathsf{T}. \quad (33)$$

This means that the first element of $x^{\langle m \rangle}$, namely x_1^m, is given by the largest element of $\mathcal{I}_{n,m}$, namely $(m, 0, \ldots, 0)$, the second element, namely $x_1^{m-1} x_2$, is given by the second largest element of $\mathcal{I}_{n,m}$, namely $(m-1, 1, 0, \ldots, 0)$, and so on. As claimed earlier, the reduced square of $x = \begin{bmatrix} a & b \end{bmatrix}^\mathsf{T}$ is indeed $x^{\langle 2 \rangle} = \begin{bmatrix} a^2 & ab & b^2 \end{bmatrix}^\mathsf{T}$.

Using this reduced Kronecker power, we can describe our original system in the form

$$x(t+1) = \sum_{i=0}^{\nu} \widehat{F}_i(t) x^{\langle i \rangle}(t), \quad t \in \mathbb{N},$$
$$x(0) = x_{\text{ini}}. \quad (34)$$

Compared to (1), where F_i is a matrix of size $n \times n^i$, $\widehat{F}_i$ is of size $n \times |\mathcal{I}_{n,i}|$ obtained from F by summing columns. For example, if (1) is of the form:

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \end{bmatrix} \begin{bmatrix} x_1(t)^2 \\ x_1(t)x_2(t) \\ x_2(t)x_1(t) \\ x_2(t)^2 \end{bmatrix},$$

Then (34) is of the form:

$$\begin{bmatrix} x_1(t+1) \\ x_2(t+1) \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} + a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} + a_{2,3} & a_{2,4} \end{bmatrix} \begin{bmatrix} x_1(t)^2 \\ x_1(t)x_2(t) \\ x_2(t)^2 \end{bmatrix}. \quad (35)$$

Everything we described in the previous sections can be accommodated with this new power, reducing the size of the vectors and the matrices involved. More details on the saved space will be given in the experiment section.

In terms of implementation, this power relies on manipulating and generating the elements of $\mathcal{I}_{n,m}$ on the lexicographic order. This can be done by representing the n-tuples as monomials and most calculations can be done using abstract polynomials operations. In our implementation, we heavily used the python package numpoly [1].

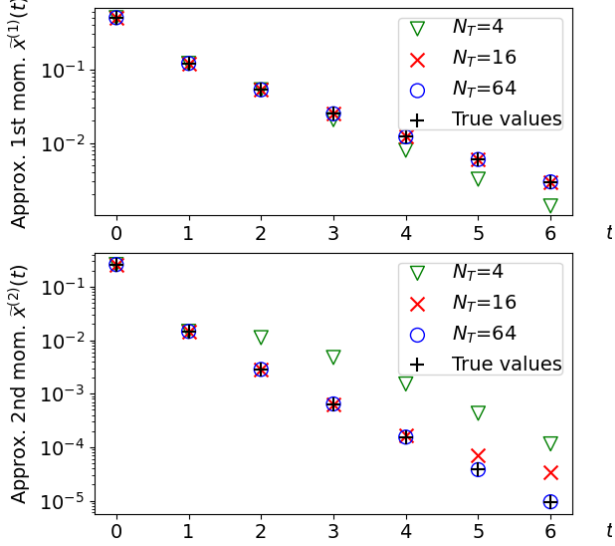[1] https://pypi.org/project/numpoly/

Fig. 3. Moment approximations for stochastic logistic map.

7 Experimental Results

In this section, we provide two numerical examples and experimental results to illustrate our techniques. All experiments are run on a standard laptop computer (MacBook Pro, M1 chip, 16G memory).

7.1 Stochastic Logistic Map

Consider the stochastic logistic map as studied by Athreya and Dai (2000), which is given by

$$x(t + 1) = r(t)x(t)(1 - x(t)), \quad t \in \mathbb{N},$$
$$x(0) = x_0,$$

where $x_0, r(0), r(1), \ldots$ are mutually independent random variables; x_0 takes values in $[0, 1]$ and $r(t)$ all take values in $[0, 4]$. The scalar $x(t) \in [0, 1]$ represents the population of a species subject to growth rate $r(t)$. This system can equivalently be represented by (1) with $\nu = 2$, $F_0(t) = 0$, $F_1(t) = r(t)$, and $F_2(t) = -r(t)$. For experiments, we chose all $r(t)$ to be uniformly distributed over the interval $[0.4, 0.6]$, and x_0 to follow a normal distribution of mean 0.5 and standard deviation 0.1 truncated to $[0, 1]$.

7.1.1 Moment Approximation via Truncated System

We first compare our moment approximations for different truncation limits to the true value of the moments (computed using our method with $N_{\mathrm{T}} = 256$, which gives the true value by Proposition 4.1). In Figure 3, we plot the first and second moments of the truncated system with different truncation limits N_{T}, and larger N_{T} is required to obtain good approximations of higher moments. This is a natural consequence, as truncation discards more information on the dynamics of higher moments.
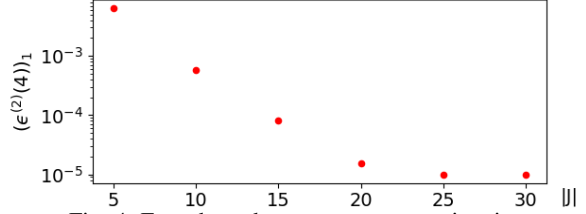

Fig. 4. Error bound on moment approximations.

7.1.2 Error Bound on Moment Approximations

Next, we evaluate our approximation method of error bounds for moments. Figure 4 shows error bounds given by Proposition 4.2 with different sizes of J, with parameters $N_{\mathrm{T}} = 16$, $t = 4$, and $j_0 = 2$. The set J contains the indices j where $\|\mathbb{E}[x_0^{[j]}]\|$ are the largest. We observe that the error bound quickly decreases as $|J|$ increases. This supports our expectation that we can use our error bounds for tail probability analysis with larger parameters (t, N_{T}, and ν). Note that we cannot expect to get much more precise than the bound for $|J| = 30$, since we get the exact error bound for $|J| = 33$.

7.1.3 Tail Probability Analysis

Lastly, we provide a result on tail probability analysis via the method in Section 5. We computed the error bound for $0 \leq t \leq 5$ using Proposition 4.2 with $N_{\mathrm{T}} = 16$ and $|J| = 6t$, where J contains the indices j where $\|\mathbb{E}[x_0^{[j]}]\|$ are the largest. Figure 5 summarizes the results of the analysis. Red intervals indicate the 95%-probability neighborhoods of $\widetilde{x}^{(1)}(t)$ computed by using Proposition 5.1. Blue intervals with a solid line indicate the region where 95% of 10000 Monte Carlo simulations closest to its mean belong. Dotted intervals indicate the range of 10000 Monte Carlo simulations. We observe that the size of safety intervals given by our tail probability analysis is reasonably small. It becomes cruder in later time steps. This is expected, as the approximation error of moments, which is a bottleneck in refining the error bounds, becomes larger as time progresses (cf. approximate 2nd moment in Fig. 3).

There are two major advantages of our method compared to Monte Carlo simulation. One is that our technique computes moment approximations much faster (even for large N_{T} and a small number of samples) because we do not rely on generating random numbers. This advantage is highlighted in Table 1, which contains the online computation times for Monte Carlo simulations and our approach, averaged over 100 runs. The offline computation of our approach takes 0.014 seconds for $N_{\mathrm{T}} = 256$. Another advantage is that our safety interval gives a theoretical guarantee on probabilistic safety that cannot be achieved by Monte Carlo simulations.

7.2 Application to Automated Driving

Our second example is an application to automated driving. For safety guarantees, autonomous vehicles need to predict

Table 1. Comparison of online computation times.

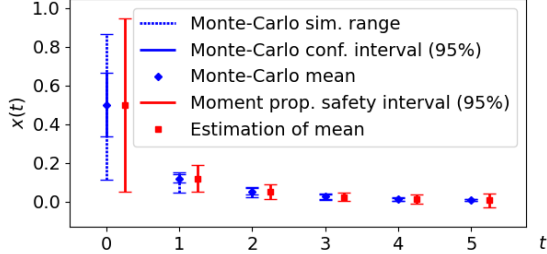| Method	Monte Carlo		Moment propagation			
Parameters	num. samples		N_{T}			
	10	10^4	4	16	64	256
Time (μs)	$4.4e10^3$	$2.8e10^5$	49	51	57	67

Fig. 5. Tail probability analysis with Monte-Carlo simulation and moment propagation with truncation limit $N_{\text{T}} = 16$.

their future positions. One way to achieve this is set-based reachability, as advocated by Althoff and Dolan (2014). To use their method, they must consider systems with bounded disturbances and use linearization around an equilibrium, that is, they approximate a polynomial system by a linear one, using Lagrange remainders. Our method is based on Carleman linearization, which allows taking the effect of higher dimensions of the system into account more precisely than Lagrange remainders. Moreover, our approach is probabilistic, while theirs is set-based, so the two approaches give different types of guarantees.

We consider a scenario in which, at each time step, the vehicle measures its current position with some known sensor error distributions, computes moments of its current state, and predicts its future positions up to t steps ahead in time by applying the truncated system to these moments.

7.2.1 Vehicle Dynamics

More precisely, we consider the *kinematic bicycle model* of a vehicle from Kong *et al.* (2015), which we rewrite as the following equivalent polynomial system

$$\dot{p}_{\text{x}}(t) = v(t)c(t), \quad \dot{p}_{\text{y}}(t) = v(t)s(t),$$
$$\dot{\psi}(t) = \frac{v(t)}{\ell}\sin\beta, \quad \dot{v}(t) = a(t),$$
$$\dot{c}(t) = -\frac{s(t)v(t)\sin\beta}{\ell}, \quad \dot{s}(t) = \frac{c(t)v(t)\sin\beta}{\ell},$$

for $t \geq 0$, where $p_{\text{x}}(t) \in \mathbb{R}$ and $p_{\text{y}}(t) \in \mathbb{R}$ represent the X–Y coordinates of the mass-center of the vehicle, $v(t) \in \mathbb{R}$ denotes its speed, $\psi(t)$ its inertial heading, and $a(t) \in \mathbb{R}$ its acceleration. The constants $\beta \in \mathbb{R}$ and $\ell > 0$ respectively denote the angle of velocity and the distance from the vehicle's rear axle to its mass-center. The scalars $c(t)$ and $s(t)$ are auxiliary variables that are introduced to obtain the polynomial model above from the original model of Kong
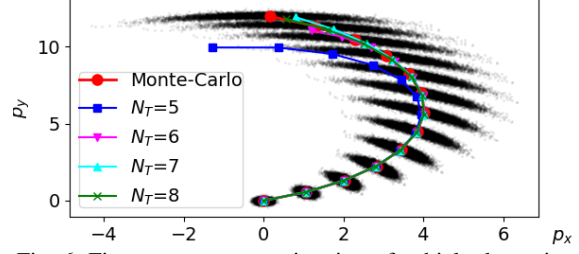

Fig. 6. First moment approximation of vehicle dynamics.

et al. (2015) (which involves trigonometric terms), using the same techniques as Carothers *et al.* (2005).

The second-order Taylor expansion of the model above gives the following discrete-time approximation:

$$p_{\text{x}}(t + \Delta) = p_{\text{x}}(t) + \Delta c(t)v(t)$$
$$+ \frac{\Delta^2}{2}\left(a(t)c(t) - \frac{s(t)v^2(t)\sin\beta}{\ell}\right),$$
$$p_{\text{y}}(t + \Delta) = p_{\text{y}}(t) + \Delta s(t)v(t)$$
$$+ \frac{\Delta^2}{2}\left(a(t)s(t) + \frac{c(t)v^2(t)\sin\beta}{\ell}\right),$$
$$\psi(t + \Delta) = \psi(t) + \Delta\frac{v(t)}{\ell}\sin\beta + \frac{\Delta^2}{2}\frac{a(t)}{\ell}\sin\beta,$$
$$v(t + \Delta) = v(t) + \Delta a(t),$$
$$c(t + \Delta) = c(t) - \Delta\frac{s(t)v(t)\sin\beta}{\ell}$$
$$- \frac{\Delta^2}{2}\left(\frac{c(t)v^2(t)\sin^2\beta}{\ell^2} + \frac{a(t)s(t)\sin\beta}{\ell}\right),$$
$$s(t + \Delta) = s(t) + \Delta\frac{c(t)v(t)\sin\beta}{\ell}$$
$$+ \frac{\Delta^2}{2}\left(-\frac{s(t)v^2(t)\sin^2\beta}{\ell^2} + \frac{a(t)c(t)\sin\beta}{\ell}\right),$$

where $\Delta > 0$. To describe the evolution of the vehicle states at times $0, \Delta, 2\Delta, \ldots$, we write this system in the form of (1). In particular, consider the discrete-time instant $t \in \mathbb{N}$ corresponding to the continuous time $t\Delta$. By letting

$$x(t) \triangleq [p_{\text{x}}(t), p_{\text{y}}(t), \psi(t), v(t), c(t), s(t)]^{\mathsf{T}},$$

we obtain (1) with $\nu = 3$ and the coefficients $F_0(t), \ldots, F_3(t)$ depend on Δ, β, ℓ, and $a(t)$. We consider the setting where the acceleration values $a(0), a(1), \ldots$ are independent uniformly-distributed random variables over $[0.9, 1]$, $\Delta = 0.1$, $\beta = \pi/8$, $\ell = 2.5$, and, for the initial state, $p_{\text{x}}(0), p_{\text{y}}(0), v(0), \psi(0)$ are independent Gaussian random variables with mean 0 and standard deviation 0.1, and $c(0) = \cos(\psi(0) + \beta)$ and $s(0) = \sin(\psi(0) + \beta)$.

7.2.2 Experimental Results

Figure 6 shows the expected trajectory of the vehicle as approximated by our method for different truncation limits, as well as the empirical distribution computed by 10000 runs
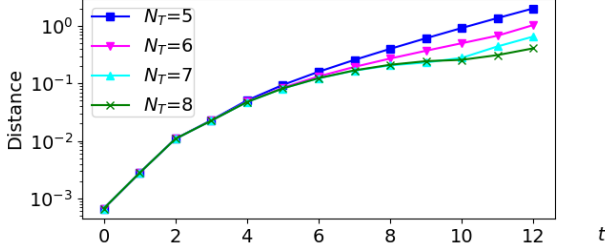
13

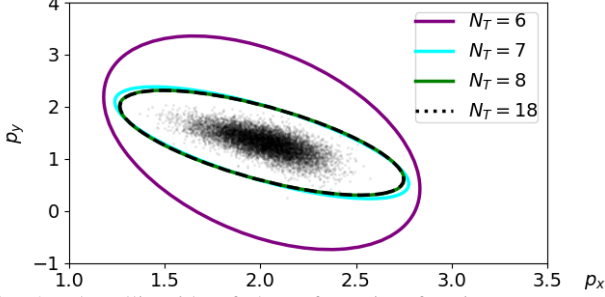Fig. 7. Distance to the mean of the empirical distribution.

Fig. 8. The ellipsoids of the safe region for time step 2 and probability bound 0.9 computed using (32) and Proposition 4.3 with $\widehat{k} = 900$ and different truncation limits N_T. Using $N_T = 18$, we obtain the region computed using the exact first and second moments (see Proposition 4.1).
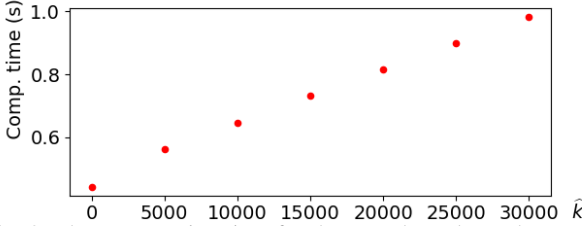

Fig. 9. The computation time for the error bounds on the approximations for $\mathbb{E}[p_x]$, $\mathbb{E}[p_y]$, $\mathbb{E}[p_x^2]$, $\mathbb{E}[p_y^2]$, and $\mathbb{E}[p_x p_y]$, at $t = 2$, using Proposition 4.3 with $N_T = 8$ and different sets K, where each of the set K contains $\widehat{k}$ indices k where $|\tilde{y}_k|$ are the largest.
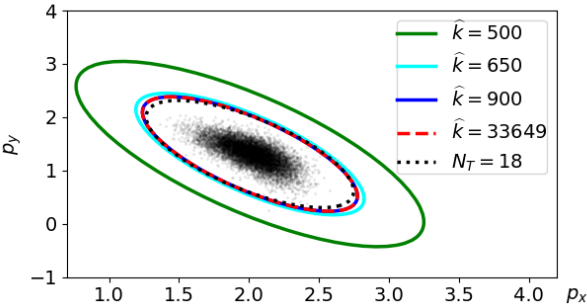

Fig. 10. The ellipsoids of the safe region for time step 2 and probability bound 0.9 computed using (32) and Proposition 4.3 with $N_T = 7$ and different sets K. Each ellipsoid is computed using the set K consisting of $\widehat{k}$ indices k where $|\tilde{y}_k|$ are the largest. Using $N_T = 18$, we obtain the region computed using the exact moments (see Proposition 4.1).

of Monte Carlo simulation and the mean of that distribution. Figure 7 shows the distance between $\widetilde{x}^{(1)}(t)$ and the mean of the empirical distribution for the same truncation limits. Figures 6 and 7 show that larger truncation limits give trun-
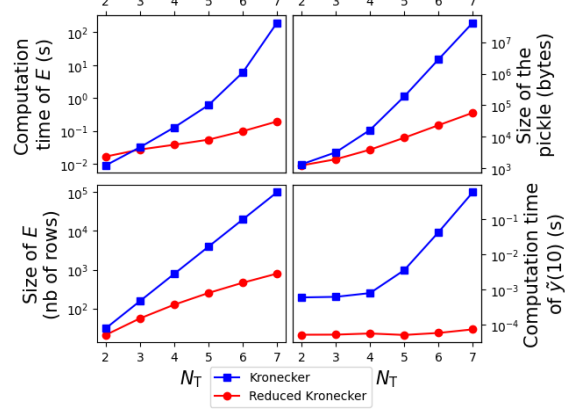

Fig. 11. Comparison between Kronecker powers and reduced ones

cated systems that follow the empirical distribution closer. It also shows that, for a fixed truncation limit, the distance to the empirical system grows larger with time.

Figure 8 shows the ellipsoids of the safe region at time step 2 with probability bound 0.9 computed using (32) and Proposition 4.3 with different truncation limits N_T. We can see that a larger N_T gives a more precise bound.

Figure 9 shows the computation time (averaged over 1000 runs) for the error bounds on moment approximation at time step 2 computed using Proposition 4.3 with different sizes of the sets K. This result shows that, as expected, the computation time increases linearly with the size of K.

Figure 10 also shows the ellipsoids of the safe region at time step 2 with probability bound 0.9, but with different index sets K. We can see that the larger K is, the closer the ellipsoids get to that obtained with the exact computation using $N_T = 18$ and Proposition 4.1 (the black dashed bound). Note that the set K containing all 33,649 indices means that it contains all moments needed for an exact computation of the upper bound of truncation error at $t = 2$ using (20) and the reduced Kronecker powers in Section 6. In other words, the red dotted bound is the tightest bound we can obtain using our method for the truncation limit $N_T = 7$. We note that the bound computed with $\widehat{k} = 900$ is very close to this tightest bound and their computation is much faster.

7.3 Comparison of Kronecker Powers and Reduced Kronecker Powers

In Section 7.2, we used the reduced Kronecker powers described in Section 6 to generate the matrices $E(N_T, N_T)$. Here, we illustrate the gain, both in time and space, obtained by using reduced Kronecker powers instead of non-reduced ones. The comparison results are compiled in Figure 11.

We compared the following performance indicators, for $N_T \in \{2, \ldots, 7\}$:

- The time to compute $E(N_T, N_T)$.

- The size of the compressed data file (pickled npz format in Numpy/Python) containing $E(N_{\mathrm{T}}, N_{\mathrm{T}})$.
- The number of rows of $E(N_{\mathrm{T}}, N_{\mathrm{T}})$.
- The time required to compute $\tilde{y}(10)$.

We remark that using reduced Kronecker powers significantly improves the performance in all aspects, making both the online and the offline computations much faster, and the memory load much lighter. Furthermore, we also note that the computation of $E(8,8)$ using non-reduced Kronecker powers reached an out-of-memory error after several hours of computation, while we could compute $E(25,25)$ in less than an hour with reduced ones.

8 Conclusions

In this paper, we have proposed a method to approximate the moments of a discrete-time stochastic polynomial system. This method is built upon a Carleman linearization approach with truncation, where approximate moments are obtained by propagating initial moments through a finite-dimensional linear deterministic difference equation. We have presented guaranteed bounds on the approximation errors. We have then used the approximate moments and the approximation error bounds together with a convex optimization technique to provide probabilistic safety analysis. We have demonstrated our method on a stochastic logistic map and a vehicle model with stochastic acceleration inputs.

Our moment approximation method is applicable to systems with both additive and multiplicative noise. Furthermore, it provides probabilistic guarantees even when the dynamics and the noise probability distributions are complicated. Our method involves computations in two phases: an initial offline computation phase where the approximate moment dynamics are obtained, and an online computation phase that involves propagation of the initial moments through the obtained dynamics. The online phase is very fast and we have shown in our numerical examples that our method can provide moment approximations in much shorter times compared to the Monte Carlo approach based on repeated simulations. For the offline computations, we have investigated a technique to improve the efficiency by using the symmetry of Kronecker powers and reducing the sizes of the matrices involved in the computations.

While in this paper we have addressed only polynomial systems, our method can also be applied to certain nonpolynomial systems. In some cases, nonpolynomial dynamics can be transformed to polynomial ones by introducing auxiliary variables, as illustrated in one of our numerical examples. In other cases, polynomial approximations can be useful.

We have used the approximated moments of the system's state for computing tail-probability bounds of the state being outside of ellipsoidal safety regions. As pointed out by Schmüdgen (2017) and John *et al.* (2007), approximate moments of random variables can be useful for approximating probability distributions. One of our future research directions is to investigate approximation of the probability distribution of the system state using its approximated moments.

References

Al-Tuwaim, M. S., O. D. Crisalle and S. A. Svoronos (1998). Discretization of nonlinear models using a modified Carleman linearization technique. In: *Proc. Amer. Contr. Conf.*. pp. 3084–3088.

Althoff, M. and J. M. Dolan (2014). Online verification of automated road vehicles using reachability analysis. *IEEE Trans. Robotics* **30**(4), 903–918.

Amini, A., Q. Sun and N. Motee (2019). Carleman State Feedback Control Design of a Class of Nonlinear Control Systems. *IFAC-PapersOnLine* **52**(20), 229–234.

Amini, A., Q. Sun and N. Motee (2020). Quadratization of Hamilton-Jacobi-Bellman equation for near-optimal control of nonlinear systems. In: *Proc. IEEE Conf. Dec. Contr.*. pp. 731–736.

Amini, A., Q. Sun and N. Motee (2021). Error bounds for Carleman linearization of general nonlinear systems. In: *Proc. Conf. Control and its Applications*. SIAM. pp. 1–8.

Athreya, K. B. and J. Dai (2000). Random logistic maps. I. *J. Theor. Probab.* **13**(2), 595–608.

Banzhaf, H., M. Dolgov, J. E. Stellet and J. M. Zöllner (2018). From footprints to beliefprints: Motion planning under uncertainty for maneuvering automated vehicles in dense scenarios. In: *Proc. Int. Conf. Intel. Transport. Systems*. pp. 1680–1687.

Bellman, R. (1970). *Introduction to Matrix Analysis*. Chap. 12, pp. 236–237. McGraw-Hill.

Bellman, R. and J. M. Richardson (1963). On some questions arising in the approximate solution of nonlinear differential equations. *Quarterly Applied Math.* **20**(4), 333–339.

Boyd, S. and L. Vandenberghe (2004). *Convex optimization*. Cambridge University Press.

Bry, A. and N. Roy (2011). Rapidly-exploring random belief trees for motion planning under uncertainty. In: *Proc. IEEE ICRA*. pp. 723–730.

Cacace, F., V. Cusimano, A. Germani and P. Palumbo (2014). A Carleman discretization approach to filter nonlinear stochastic systems with sampled measurements. *IFAC Proceedings Volumes* **47**(3), 9534–9539.

Cacace, F., V. Cusimano, A. Germani, P. Palumbo and M. Papi (2019). Optimal continuous-discrete linear filter and moment equations for nonlinear diffusions. *IEEE Trans. Autom. Control* **65**(10), 3961–3976.

Carothers, D. C., G. E. Parker, J. S. Sochacki and P. G. Warne (2005). Some properties of solutions to polynomial systems of differential equations. *Electr. J. Differ. Eq.* **2005**(40), 1–17.

Carravetta, F., A. Germani and M. Raimondi (1996). Polynomial filtering for linear discrete time non-Gaussian systems. *SIAM J. Control Optim.* **34**(5), 1666–1690.

Cibulka, V., T. Haniš, M. Korda and M. Hromčík (2020). Model Predictive Control of a Vehicle using Koopman Operator. *IFAC-PapersOnLine* **53**(2), 4228–4233.

Forets, M. and A. Pouly (2017). Explicit error bounds for Carleman linearization. *arXiv preprint arXiv:1711.02552*.

Goswami, D. and D. A. Paley (2017). Global bilinearization and controllability of control-affine nonlinear systems: A Koopman spectral approach. In: *Proc. IEEE Conf. Dec. Contr.*. pp. 6107–6112.

Gray, H. L. and S. Wang (1991). A general method for approximating tail probabilities. *J. Am. Stat. Assoc.* **86**(413), 159–166.

Hashemian, N. and A. Armaou (2019). Feedback control design using model predictive control formulation and Carleman approximation method. *AIChE J.* **65**, 1–11.

Jasour, A., A. Wang and B. C. Williams (2021). Moment-based exact uncertainty propagation through nonlinear stochastic autonomous systems. *arXiv preprint arXiv:2101.12490*.

John, V., I. Angelov, A.A. Öncül and D. Thévenin (2007). Techniques for the reconstruction of a distribution from a finite number of its moments. *Chem. Eng. Sci.* **62**(11), 2890–2904.

Kong, J., M. Pfeiffer, G. Schildbach and F. Borrelli (2015). Kinematic and dynamic vehicle models for autonomous driving control design. In: *Proc. IEEE Intell. Veh. Symp.*. pp. 1094–1099.

Laub, A. J. (2005). *Matrix Analysis for Scientists and Engineers*. SIAM.

Mesbahi, A., J. Bu and M. Mesbahi (2019). On modal properties of the Koopman operator for nonlinear systems with symmetry. In: *Proc. Amer. Contr. Conf.*. pp. 1918–1923.

Pruekprasert, S., T. Takisaka, C. Eberhart, A. Cetinkaya and J. Dubut (2020). Moment propagation of discrete-time stochastic polynomial systems using truncated Carleman linearization. *IFAC-PapersOnLine* **53**(2), 14462–14469.

Rauh, A., J. Minisini and H. Aschemann (2009). Carleman Linearization for Control and for State and Disturbance Estimation of Nonlinear Dynamical Processes. *IFAC Proceedings Volumes* **42**(13), 455–460.

Schmüdgen, K. (2017). *The Moment Problem*. Springer.

Shen, X., X. Zhang and P. Raksincharoensak (2020). Probabilistic Bounds on Vehicle Trajectory Prediction Using Scenario Approach. *IFAC-PapersOnLine* **53**(2), 2385–2390.

Steeb, W.-H. and F. Wilhelm (1980). Non-linear autonomous systems of differential equations and Carleman linearization procedure. *J. Math. Anal. Appl.* **77**(2), 601–611.

Wong, W. S. (1983). Carleman linearization and moment equations of nonlinear stochastic equations. *Stochastics* **9**(1-2), 77–101.

A Proof of Theorem 3.3

First, we have the simple observation in Lemma A.1.

Lemma A.1. *The sets $H_{j,k}$ satisfy the recursive formula*

$$H_{j+1,k} = \bigcup_{n=0}^{\nu} \{n\} \times H_{j,k-n},$$

with the convention that $H_{j,k} = \emptyset$ for $k < 0$.

Next, we show in Lemma A.2 that Theorem 3.3 holds for $j = 2$. We omit t for simplicity.

Lemma A.2. $\left(F_0 x^{[0]} + F_1 x^{[1]} + \ldots + F_\nu x^{[\nu]}\right)^{[2]} = \sum_{k=0}^{2\nu} \left(\sum_{(i_1,i_2) \in H_{2,k}} F_{i_1} \otimes F_{i_2} \right) x^{[k]}.$

Proof. Recall that Kronecker product has the *mixed-product property* (i.e., $(A \otimes B)(C \otimes D) = AC \otimes BD$), but not *commutative property* ($(A \otimes B) = (B \otimes A)$ may not hold) (see Section 13.2 of Laub (2005))). Then, this lemma holds by Equation (A.1). □

Now, we are ready to prove Theorem 3.3.

Proof of Theorem 3.3. The theorem obviously holds for $j = 0$ and $j = 1$, and also holds for $j = 2$ by Lemma A.2. We will show the case where $j > 2$ by induction. We assume that the theorem hold for $j = n$, and consider the inductive case where $j = n + 1$ in Equation (A.2), where we write $\bigotimes_{m=1}^{n} F_{i_m}$ for $F_{i_1} \otimes \ldots \otimes F_{i_m}$ for brevity. Note that the last line holds by adding terms that correspond to the same moment $x^{[k]}$ (such terms span diagonals in the large sum in Equation (A.2)) and by Lemma A.1. Therefore, the theorem holds for $j = n + 1$, which concludes the proof. □

$$\left(F_0 x^{[0]} + F_1 x^{[1]} + \ldots + F_\nu x^{[\nu]}\right)^{[2]}$$

$$= F_0 x^{[0]} \otimes \left(F_0 x^{[0]} + \ldots + F_\nu x^{[\nu]}\right) + F_1 x^{[1]} \otimes \left(F_0 x^{[0]} + \ldots + F_\nu x^{[\nu]}\right) + \ldots + F_\nu x^{[\nu]} \otimes \left(F_0 x^{[0]} + \ldots + F_\nu x^{[\nu]}\right)$$

$$= (F_0 \otimes F_0) x^{[0+0]} + (F_0 \otimes F_1) x^{[0+1]} + (F_0 \otimes F_2) x^{[0+2]} + \ldots + (F_0 \otimes F_\nu) x^{[0+\nu]}$$
$$+ (F_1 \otimes F_0) x^{[1+0]} + (F_1 \otimes F_1) x^{[1+1]} + \ldots + (F_1 \otimes F_{\nu-1}) x^{[1+(\nu-1)]} + (F_1 \otimes F_\nu) x^{[1+\nu]}$$
$$+ (F_2 \otimes F_0) x^{[2+0]} + \ldots$$
$$+ \ldots + (F_\nu \otimes F_0) x^{[\nu+0]} + \quad \ldots + (F_\nu \otimes F_\nu) x^{[\nu+\nu]}$$

$$= \sum_{(i_1,i_2)\in H_{2,0}} \left(F_{i_1} \otimes F_{i_2}\right) x^{[0]} + \sum_{(i_1,i_2)\in H_{2,1}} \left(F_{i_1} \otimes F_{i_2}\right) x^{[1]} + \ldots + \sum_{(i_1,i_2)\in H_{2,2\nu}} \left(F_{i_1} \otimes F_{i_2}\right) x^{[2\nu]} = \sum_{k=0}^{2\nu} \left(\sum_{(i_1,i_2)\in H_{2,k}} F_{i_1} \otimes F_{i_2}\right) x^{[k]}.$$

$$(A.1)$$

$$(F_0 x^{[0]} + F_1 x^{[1]} + \ldots + F_\nu x^{[\nu]})^{[n+1]}$$
$$= (F_0 x^{[0]} + F_1 x^{[1]} + \ldots + F_\nu x^{[\nu]}) \otimes (F_0 x^{[0]} + F_1 x^{[1]} + \ldots + F_\nu x^{[\nu]})^{[n]}$$

$$= (F_0 x^{[0]} + F_1 x^{[1]} + \ldots + F_\nu x^{[\nu]}) \otimes \sum_{k=0}^{n\nu} \left(\sum_{(i_1,\ldots,i_n)\in H_{n,k}} F_{i_1} \otimes \ldots \otimes F_{i_n}\right) x^{[k]}$$

$$= F_0 x^{[0]} \otimes \sum_{k=0}^{n\nu} \left(\sum_{(i_1,\ldots,i_n)\in H_{n,k}} F_{i_1} \otimes \ldots \otimes F_{i_n}\right) x^{[k]} + \ldots + F_\nu x^{[\nu]} \otimes \sum_{k=0}^{n\nu} \left(\sum_{(i_1,\ldots,i_n)\in H_{n,k}} F_{i_1} \otimes \ldots \otimes F_{i_n}\right) x^{[k]}$$

$$= \sum_{(i_1,\ldots,i_n)\in H_{n,0}} \left(F_0 \otimes \bigotimes_{m=1}^{n} F_{i_m}\right) x^{[0+0]} + \sum_{(i_1,\ldots,i_n)\in H_{n,1}} \left(F_0 \otimes \bigotimes_{m=1}^{n} F_{i_m}\right) x^{[0+1]} + \ldots + \sum_{(i_1,\ldots,i_n)\in H_{n,n\nu}} \left(F_0 \otimes \bigotimes_{m=1}^{n} F_{i_m}\right) x^{[0+n\nu]}$$
$$+ \sum_{(i_1,\ldots,i_n)\in H_{n,0}} \left(F_1 \otimes \bigotimes_{m=1}^{n} F_{i_m}\right) x^{[1+0]} + \sum_{(i_1,\ldots,i_n)\in H_{n,1}} \left(F_1 \otimes \bigotimes_{m=1}^{n} F_{i_m}\right) x^{[1+1]} + \ldots + \sum_{(i_1,\ldots,i_n)\in H_{n,n\nu}} \left(F_1 \otimes \bigotimes_{m=1}^{n} F_{i_m}\right) x^{[1+n\nu]}$$
$$+ \ldots$$
$$+ \sum_{(i_1,\ldots,i_n)\in H_{n,0}} \left(F_\nu \otimes \bigotimes_{m=1}^{n} F_{i_m}\right) x^{[\nu+0]} + \sum_{(i_1,\ldots,i_n)\in H_{n,1}} \left(F_\nu \otimes \bigotimes_{m=1}^{n} F_{i_m}\right) x^{[\nu+1]} + \ldots + \sum_{(i_1,\ldots,i_n)\in H_{n,n\nu}} \left(F_\nu \otimes \bigotimes_{m=1}^{n} F_{i_m}\right) x^{[\nu+n\nu]}$$
$$= \sum_{k=0}^{(n+1)\nu} \left(\sum_{(i_1,\ldots,i_{n+1})\in H_{n+1,k}} F_{i_1} \otimes \ldots \otimes F_{i_{n+1}}\right) x^{[k]}.$$

$$(A.2)$$