Uncertainty Propagation Using Probabilistic Affine Forms and Concentration of Measure Inequalities

Olivier Bouissou[2], Eric Goubault[1], Sylvie Putot[1], Aleksandar Chakarov[3], and Sriram Sankaranarayanan[3(✉)]

[1] LIX, Ecole Polytechnique, CNRS, Université Paris-Saclay, Paris-Saclay, France
[2] CEA, LIST, Gif-sur-Yvette, France
[3] University of Colorado, Boulder, USA
srirams@colorado.edu

Abstract. We consider the problem of reasoning about the probability of assertion violations in straight-line, nonlinear computations involving uncertain quantities modeled as random variables. Such computations are quite common in many areas such as cyber-physical systems and numerical computation. Our approach extends probabilistic affine forms, an interval-based calculus for precisely tracking how the distribution of a given program variable depends on uncertain inputs modeled as noise symbols. We extend probabilistic affine forms using the precise tracking of dependencies between noise symbols combined with the expectations and higher order moments of the noise symbols. Next, we show how to prove bounds on the probabilities that program variables take on specific values by using concentration of measure inequalities. Thus, we enable a new approach to this problem that explicitly avoids subdividing the domain of inputs, as is commonly done in the related work. We illustrate the approach in this paper on a variety of challenging benchmark examples, and thus study its applicability to uncertainty propagation.

1 Introduction

We consider the problem of propagating uncertainty through computation that generates random numbers with known distributions on-the-fly, and computes a variety of arithmetic operations on these numbers. Such computations are common in a wide variety of applications including systems biology, robotics, control theory and randomized algorithms. Reasoning about uncertainties involves answering *queries* about the probabilities of assertions over the program variables, expectations of expressions, and more generally, characterizing the possible probability distributions of program expressions, at the output. Often, the random number generators draw values from simple distributions such as uniform random, gaussian or exponential. However, as a result of nonlinear operations, the resulting distributions can be quite complex.

In this work, we restrict our attention to *straight line computations* involving random variables. In other words, the programs do not branch on the values

M. Chechik and J.-F. Raskin (Eds.): TACAS 2016, LNCS 9636, pp. 225–243, 2016.
DOI: 10.1007/978-3-662-49674-9_13

of the random variables involved. Nevertheless, such computations are surprisingly common in many applications arising from controls, robotics and scientific computation that can generate thousands of random variables. Currently, these applications are beyond many of the existing approaches for reasoning about probabilistic programs. Our approach combines the framework of probabilistic affine forms introduced by Bouissou et al. [7] to represent program variables in terms of interval linear expressions involving uncertain *noise symbols*, and *concentration of measure* inequalities in probability theory [13] to answer queries. This approach has two main advantages: (a) probabilistic affine forms can be used to rapidly approximate several nonlinear arithmetic operations including trigonometric operations, and (b) the application of concentration of measure inequalities yields valid probability bounds without the need to perform expensive subdivisions of the set of support. In fact, in situations involving more than a few tens of noise symbols, such a subdivision is prohibitively expensive.

The contributions of this paper include (a) we extend probabilistic affine forms with precise tracking of the bounds on the expectations and higher-order moments of these forms, (b) we propose the use of concentration of measure inequalities to reason about the probabilities of queries over affine forms and (c) we demonstrate our approach on many challenging examples involving nonlinear arithmetic operations. Wherever possible, we also compare our approach with the previous use of probabilistic affine forms without concentration of measure inequalities [7]. The experimental evaluation in this paper allows us to draw two main conclusions. *(A)* Probabilistic affine forms are seen to be quite efficient even for nonlinear trigonometric and rational functions over random variables. However, this is at the cost of information lost due to linear approximation of nonlinear computations. *(B)* Concentration of measure inequalities can prove bounds on the probabilities of rare events for large affine forms, quite efficiently. Often, such bounds seem beyond the reach of related techniques. On the flip side, the bounds may sometimes be too conservative due to the abstraction.

Related Work

Many approaches have focused on the problem of reasoning about uncertainties as they propagate through computation. These include approaches from interval arithmetic, polynomial chaos approximations, symbolic verification, and statistical approaches.

Interval Arithmetic and Imprecise Probabilities: Imprecise probability representations describe sets of probability distributions. These are well-suited for describing situations where some values, or events are known non-deterministically (e.g. values in an interval), whereas others are known probabilistically. Tools from this domain include P-boxes [17] and Dempster-Shafer structures [33]. These have been used to propagate both probabilistic and non-deterministic information in numerical simulation for instance, see also [8,18,21,30,37,38]. Arithmetic rules for P-boxes have been studied [39] and implemented in toolboxes such as DSI, INTLAB, and RiskCalc [3,16,31]. Our work builds on probabilistic affine forms

proposed by Bouissou et al., wherein a variety of operators over these forms including meet, join and widening operators are presented [2,7].

However, these approaches rely on an explicit, finite representation of probability bounds that requires us to decompose the joint domain of distributions of these random variables. Such a decomposition rapidly becomes intractable beyond a few tens of random variables. We partly tackle this issue in our approach using concentration of measure inequalities, whose application does not require a decomposition.

Polynomial chaos approximations express the output distributions as polynomials over the input random variables [40]. However, these approximations also suffer from the curse of dimensionality. Moreover, polynomial chaos approximations focus on estimating moments, but not necessarily on providing probability bounds.

Formal Verification Approaches: Prism and related model checking tools have revolutionized the problem of reasoning about finite state probabilistic programs [25]. This has spurred interest in infinite state programs involving more complex random variables with distributions such as gaussian and exponential.

Related approaches include probabilistic symbolic executions that extend traditional symbolic execution over probabilistic programs and probabilistic abstract interpretation. Probabilistic symbolic execution has been explored for analyzing complex programs computing over random variables [4,19,32]. These approaches rely on expensive volume approximation techniques either off the shelf [12], or using domain decomposition [32]. Barring a few exceptions [4], they are restricted to programs with linear assignments and conditionals. However, recent work by Chistikov et al. has demonstrated a randomized approximation to volume estimation that holds the promise of scaling to larger systems involving thousands of random variables [10]. However, that approach is currently restricted to linear arithmetic SMT formulas. The ProbReach tool by Shmarov et al. also provides precise probability bounds for nonlinear continuous-time systems, building on top of the dReach tool [35]. While capable of precise reasoning for complex nonlinear systems, it relies on domain decomposition. In particular, it is currently restricted to systems with uncertainties in initial parameters as opposed to stochastic systems that are driven by noisy inputs. Similar ideas using Taylor models have been investigated by Enszer et al. [15]. Finally, the work of Abate et al. derives discrete Markov chain abstractions to compute probability of reaching unsafe states in general stochastic Markov processes [1]. The discretization also involves a subdivision of the state space of these processes with a finer subdivision providing better results. In contrast, our approach does not subdivide the state or random variables. However, our approach depends intimately on obtaining good bounds for expectations and higher-order moments for noise symbols.

Abstract domains for probabilistic programs have been investigated by Monniaux [29] and Cousot and Monereau [11]. Whereas our approach focuses on finite computations, abstract interpretation typically excels in dealing with unbounded

```
1  angles = [10, 60, 110, 160, 140, ...
2            100, 60, 20, 10, 0]
3  x := TruncGaussian(0,0.05,-0.5,0.5)
4  y := TruncGaussian(0, 0.1,-0.5,0.5)
5  for reps in range(0,100):
6  #iterate through angles
7    for theta in angles:
8      # Distance travelled variation
9      d = Uniform(0.98,1.02)
10     # Steering angle variation
11     t = deg2rad(theta) * (1 + ...
12         TruncGaussian(0,0.01,-0.05,0.05))
13     # Move distance d with angle t
14     x = x + d * cos(t)
15     y = y + d * sin(t)
16 #Probability that we went too far?
17 assert(x >= 272)
```

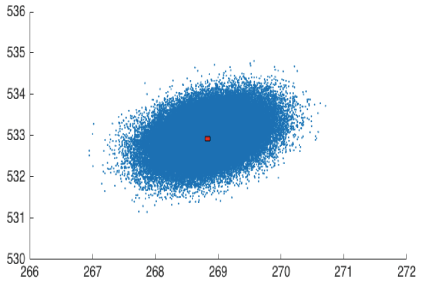


Fig. 1. Left: A probabilistic program capturing the final position of 2D robotic end effector. **Right:** Scatter plot showing the final (x, y) values.

length computations wherein approximations such as join (see also [2]) and widening provide the ability to generalize. Previous work by Chakarov et al. also uses concentration of measure inequalities in this context to handle loops in probabilistic programs [9].

Statistical Approaches: Finally, statistical approaches use hypothesis testing to answer queries on uncertainties [24,41]. The main advantage lies in the ability to handle quite complex systems through simulations. However, the disadvantages often involve rare events, wherein the number of simulations required to gain a given degree of statistical confidence is simply prohibitive. In such situations, techniques like importance sampling have been applied to minimize the number of simulations [23]. However, statistical approaches provide guarantees that are fundamentally different from ours. Also, with very few exceptions [26], they do not attempt to represent the output distribution but simply answer queries by examining the evidence from simulations. As such, very little work has been undertaken to relate the two types of guarantees. A related approach by Bernholt et al. [5], introduces an explicit uncertainty data type to reason about uncertainty using Bayesian hypothesis testing. Therein, the main idea is to use Bayes networks to represent the influence of random variables over program variables and allow hypothesis testing techniques to enable programmers to deal with this uncertainty in making decisions.

2 Motivating Example

Figure 1 shows an example probabilistic program that models the (x, y) position of a simple 2D robotic end effector that starts close to the origin and whose series of motions is specified by the list `angles`. The initial position is uncertain with a truncated normal distribution centered at the origin and with given variance as shown in lines 3, 4. At each iteration, the effector moves from its current position (x, y) to $x + d_j \cos(\theta_j), y + d_j \sin(\theta_i)$, wherein d_j is distributed as a uniform

random number in the interval $[0.95, 1.05]$ (line 9, modeling the distance 1.0 with a 5 % uniform error). Likewise, θ_i is given by multiplying `angles` (i) with a truncated Gaussian random variable centered around 1 with variance 0.01 in the interval $[0.95, 1.05]$ (line 12). The position update is shown in lines 14 and 15. We are interested in the probability that an assertion violation is triggered in line 17.

A scatter plot (Fig. 1) of the values of (x, y) at the end of the computation are shown. As noted, 10^5 simulations do not produce any violations of the property $x \geq 272$. In fact, the largest value of x seen in our simulations is around 271. Therefore, we may rightfully conclude that it is "quite rare" to reach $x \geq 272$. On the other hand, using nondeterministic semantics for the random choices concludes a potentially reachable range of $x \in [210.5, 324.3]$. We therefore, seek to know bounds on the probability that the assertion is satisfied.

Affine Forms at Output: Our approach uses symbolic execution to track the value of x at the output as a function of random variables called *noise symbols*. The affine form for x is (partially) shown below:

$$x : \begin{pmatrix} [268.78, 268.82] + [1,1] * y_0 + [0.984, 0.985] * y_2 + [0.030, 0.031] * y_3 + [-1,-1] * y_4 + [0.030, 0.031] * y_5 \\ +[-1,-1] * y_6 + [0.49, 0.51] * y_9 + [0.90, 0.91] * y_{10} + [-1,-1] * y_{11} + [0.90, 0.91] * y_{12} + \\ \cdots \\ [0.03, 0.031] * y_{6892} + [-1,-1] * y_{6893} + [1,1] * y_{6896} + [-1,-1] * y_{6898} + [-1,-1] * y_{6899} \end{pmatrix}.$$

Here, each y_i is a noise symbol with associated information concerning it's range, dependencies with other noise symbol, expectations and higher order moments (e.g., the second moment). For instance, y_0 corresponds to the truncated Gaussian random variable in line 3. Using this affine form, we conclude at the end of computation that the value of x has an expectation in the range $[265.9, 268.9]$ and variance in the range $[0.17, 0.23]$. This matches with the empirical evidence gathered from 10^5 simulations. The time required for the affine form was ~ 15 s and comparable to 10^5 simulations in Matlab (~ 20 s).

Reasoning with Affine Forms: Finally, we utilize a concentration of measure inequality to obtain the guarantee $\mathbb{P}(x \geq 272) \leq 6.2 \times 10^{-7}$ [13]. We note that such bounds on rare events are often valuable, and hard to establish.

3 Probabilistic Affine Forms

In this section, we introduce probabilistic affine forms involving random variables known as *noise symbols*, and discuss the approximation of straight line computations using these affine forms.

3.1 Random Variables, Expectations, Moments and Independence

Let $\mathbb{R}$ represent the real numbers and $\overline{\mathbb{R}} = \mathbb{R} \cup \{\infty, -\infty\}$. Univariate random variables over reals are defined by a *cumulative density function* (CDF) $F : \overline{\mathbb{R}} \mapsto [0, 1]$, wherein $F(-\infty) = 0$, $F(\infty) = 1$ and F is a non-decreasing, right continuous function with left limits. The value of $F(t)$ represents the probability

$\mathbb{P}(X \leq t)$ for any $t \in \overline{\mathbb{R}}$. The CDF naturally extends to multivariate random variables as well [14].

The *expectation* of a function $g(X)$ for random variable X, denoted by $\mathbb{E}(g(X))$ is defined as the integral: $\mathbb{E}(g(X)) : \int_{\mathcal{D}} g(\boldsymbol{x}) dF(\boldsymbol{x})$. Here $\mathcal{D}$, the domain of integration, ranges over the *set of support* for the random variable X. The expectation exists if the integral is well-defined and yields a finite value. An important property of expectations is their *linearity*. Whenever the expectations exist, and are finite, we have $\mathbb{E}(\sum_{i=1}^{k} a_i g_i(\boldsymbol{x})) = \sum_{i=1}^{k} a_i \mathbb{E}(g_i(\boldsymbol{x}))$, for constants $a_1, \ldots, a_k$ and functions $g_1, \ldots, g_k$. Likewise, the k^{th} moment for $k \geq 1$ for a random variable X is defined as $\mathbb{E}(X^k)$. Its variance is defined as $\textsc{Var}(X) : \mathbb{E}((X - \mathbb{E}(X))^2)$.

A pair of random variables (X_1, X_2) are *independent* if and only if their CDF $F(x_1, x_2)$ can be decomposed as $F(x_1, x_2) : F_1(x_1)F_2(x_2)$. Otherwise, the random variables are called *correlated*. More generally, $(X_1, \ldots, X_n)$ are pairwise independent iff $F(x_1, \ldots, x_n) : F_1(x_1) \cdots F_n(x_n)$. If X_1, X_2 are independent then it follows that $\mathbb{E}(g(X_1)h(X_2)) = \mathbb{E}(g(X_1))\mathbb{E}(h(X_2))$.

We assume that random variables that we encounter in this paper are well-behaved in the following sense: (a) Each random variable has a bounded set of support. However, we present a simple trick to handle distributions such as gaussians that have unbounded sets of support. (b) Expectations and higher moments of the random variables are finite and computable. We recall useful properties of expectations:

Lemma 1. *Let X be a (univariate) random variable whose set of support is the interval $I \subseteq \overline{\mathbb{R}}$. It follows that $\mathbb{E}(X) \in I$.*

Let X_1, X_2 be two random variables. The following inequality holds:

$$-\sqrt{\mathbb{E}(X_1^2)\mathbb{E}(X_2^2)} \leq \mathbb{E}(X_1 X_2) \leq \sqrt{\mathbb{E}(X_1^2)\mathbb{E}(X_2^2)}\,.$$

The inequality above follows from the Cauchy-Schwarz inequality.

3.2 Environments and Affine Forms

Before introducing affine forms, we first define noise symbols and the data associated with these symbols. Let $\boldsymbol{y} : (y_1, \ldots, y_n)$ represent a set of random variables called *noise symbols*. Each noise symbol y_j is associated with an interval of support I_j, and a vector of moment intervals $I(y_j) = (I_j^{(1)}, \ldots, I_j^{(k)})$, wherein $\mathbb{E}(y_j^l) \in I_j^{(l)}$.

Note that in addition to storing estimates of $\mathbb{E}(y_i^l)$, we may optionally store moments of the form $\mathbb{E}(y_i y_j)$ for pairs $y_i, y_j \in \boldsymbol{y}$ for $i \neq j$. This can also extend to higher order moments of the form $\mathbb{E}(y_1^{l_1} \cdots y_n^{l_n})$ for monomials. In this presentation, we restrict ourselves to (marginal) expectations of single random variables of the form $\mathbb{E}(y_j^l)$, using Lemma 1 to conservatively estimate missing moment information.

Finally, our approach produced new noise symbols y_j that are functions of other noise symbols $y_j : f(y_{j_1}, \ldots, y_{j_m})$. While we abstract away the function f,

we remember these functional dependencies as a directed (functional) *dependence* graph G with vertices $V : \{y_1, \ldots, y_n\}$ and edges $E \subseteq V \times V$ wherein the edge (y_i, y_j) signifies that the random variable $y_i : f(\cdots, y_j, \cdots)$ for some function f. Clearly, if $(y_i, y_j) \in E$ and $(y_j, y_k) \in E$ we will also require $(y_i, y_k) \in E$. The edge relation E is thus a transitive relation over $\boldsymbol{y}$. For simplicity, we also add all self-loops $(y_i, y_i) \in E$.

Definition 1 (Environment). *An* environment $\mathcal{E} : \langle \boldsymbol{y}, \mathcal{I}, \mathcal{M}, G \rangle$ *is a collection of noise symbols* $\boldsymbol{y} : (y_1, \ldots, y_n)$*, the sets of support for each noise symbol* $\mathcal{I} : (I_1, \ldots, I_n)$*, the moment intervals for each noise symbol* $\mathcal{M} : (I(m_1), \ldots, I(m_n))$ *and the directed functional dependence graph* G*.*

Based on the functional dependence graph, we define the notion of independence between random variables.

Definition 2 (Probabilistic Dependence). *Noise symbols* y_i *and* y_j *are* probabilistically dependent *random variables if there exists* y_k *such that* (y_i, y_k) *and* (y_j, y_k) *belong to the graph* G*. Otherwise, they represent independent random variables.*

The probabilistic dependence graph $\widehat{G}$ *is an undirected graph where an undirected edge* (y_i, y_j) *exists in* $\widehat{G}$ *iff there exists* y_k *such that* (y_i, y_k)*,* $(y_j, y_k) \in E$ *of* G[1]*.*

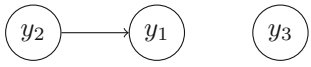
An affine form is an interval-valued linear expression over noise symbols [7].

Definition 3 (Affine Form). *An* affine form $f(\boldsymbol{y})$ *is a linear expression* $f(\boldsymbol{y}) : a_0 + \sum_{j=1}^{n} a_j y_j$*, with real*[2] *coefficients* a_j*.*

Example 1 (Environments and Affine Forms). Let us consider an environment $\mathcal{E}$ with the noise symbols y_1, y_2, y_3. Here, y_j is a random variable over the set of support $I_j : [-j, j]$, for $j = 1, 2, 3$, respectively. The moment vectors containing information up to the 4^{th} moments are provided below:

	$\mathbb{E}(y_j)$	$\mathbb{E}(y_j^2)$	$\mathbb{E}(y_j^3)$	$\mathbb{E}(y_j^4)$		
$I(m_1)$:	$([0, 0]$,	$[\frac{2}{3}, \frac{2}{3}]$,	$[0, 0]$,	$[\frac{2}{5}, \frac{2}{5}])$	$\leftarrow$	Moments for y_1
$I(m_2)$:	$([0, 0.1]$,	$[1, 1.1]$,	$[-0.1, 0.1]$,	$[0.1, 0.2])$	$\leftarrow$	Moments for y_2
$I(m_3)$:	$([-1, 0.2]$,	$[0.1, 1.2]$,	$[-0.5, 0.5]$,	$[1.1, 2.3])$	$\leftarrow$	Moments for y_3

The graph with dependencies is shown below (without the self-loops):



As a result, the variables y_1, y_3 are independent. But y_1 and y_2 are dependent. The expression $f_1 : [-1, 2] + [3, 3.1]y_1 + [1.9, 2.3]y_2 + [-0.3, -0.1]y_3$ is an affine form over $y_1, \ldots, y_3$ in the environment $\mathcal{E}$.

[1] The functional dependence graph is akin to the points-to graph in programs, whereas the probabilistic dependence graph is analogous to the alias graph.

[2] In the implementation, these coefficients will be safely over-approximated either by intervals of floating-point numbers, or by floating-point coefficients but with additional noise terms over-approximating the error.

Semantics: We briefly sketch the semantics of environments and affine forms.

An environment $\mathcal{E}$ with noise symbols $\boldsymbol{y} : (y_1, \ldots, y_n)$ corresponds to a set of possible random vectors $Y : (Y_1, \ldots, Y_n)$ that conform to the following constraints: (a) $(Y_1, \ldots, Y_n)$ must range over the set of support $I_1 \times \cdots \times I_n$. They cannot take on values outside this set. (b) The moment vectors lie in the appropriate ranges defined by $\mathcal{E} : (\mathbb{E}(Y_j), \ldots, \mathbb{E}(Y_j^k)) \in I(m_j)$. (c) If noise symbols y_i, y_j are independent according to the dependence graph G (Definition 2), the corresponding random variables Y_i, Y_j are mutually independent. Otherwise, they are "arbitrarily" correlated while respecting the range and moment constraints above. Semantically, an affine form $f(\boldsymbol{y}) : a_0 + \sum_{i=1}^{n} a_i y_i$ represents a set of linear expressions $[\![f(\boldsymbol{y})]\!]$ over $\boldsymbol{y}$:

$$[\![f(\boldsymbol{y})]\!] := \left\{ r_0 + \sum_{i=1}^{n} r_i Y_i \mid r_i \in a_i, (Y_1, \ldots, Y_n) \in [\![\mathcal{E}]\!] \right\}.$$

We now present the basic operations over affine forms including sums, differences, products and continuous (and k-times differentiable) functions over affine forms.

Sums, Differences and Products: Let f_1, f_2 be affine forms in an environment $\mathcal{E}$ given by $f_1 : \boldsymbol{a}^t \boldsymbol{y} + a_0$ and $f_2 : \boldsymbol{b}^t \boldsymbol{y} + b_0$. We define the sum $f_1 \oplus f_2$ to be the affine form $(\boldsymbol{a} + \boldsymbol{b})^t \boldsymbol{y} + (a_0 + b_0)$.

Likewise, let λ be a real number. The affine form λf_1 is given by $(\lambda \boldsymbol{a})^t \boldsymbol{y} + \lambda a_0$.

We now define the product of two forms $f_1 \otimes f_2$.

$$f_1 \otimes f_2 : \; a_0 b_0 + a_0 f_2 + b_0 f_1 + \mathsf{approx}\Big(\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j y_i y_j\Big).$$

The product operation separates the affine and linear parts of this summation from the nonlinear part that must be approximated to preserve the affine form. To this end, we define a function approx that replaces the nonlinear terms by a collection of fresh random variables. In particular, we add a fresh random variable y_{ij} to approximate the product term $y_i y_j$.

Dependencies: We add the dependency edges (y_{ij}, y_i) and (y_{ij}, y_j) to the graph G to denote the functional dependence of the fresh noise symbol on y_i and y_j.

Set of Support: The set of support for y_{ij} is the interval product of the set of supports for y_i, y_j, respectively. In particular if $i = j$, we compute the set of support for y_i^2. Let I_{ij} be the interval representing the set of support for y_{ij}.

Moments: The moments of y_{ij} are derived from those of y_i and y_j, as follows. *Case-1* $(i = j)$. If $i = j$, we have that the $\mathbb{E}(y_{ij}^p) = \mathbb{E}(y_i^{2p})$. Therefore, the even moments of y_i are taken to provide the moments for y_{ij}. However, since we assume that only the first k moments of y_i are available, we have that the first $\frac{k}{2}$ moments of y_{ij} are available, in general. To fill in the remaining moments, we approximate using intervals as follows: $\mathbb{E}(y_{ij}^r) \in I_{ij}^r$. While this approximation is often crude, this is a tradeoff induced by our inability to store infinitely many moments for the noise symbols.

Case-2 ($i \neq j$). If $i \neq j$, we have that $\mathbb{E}(y_{ij}^p) = \mathbb{E}(y_i^p y_j^p)$. If y_i, y_j form an independent pair, this reduces back to $\mathbb{E}(y_i^p)\mathbb{E}(y_j^p)$. Thus, in this instance, we can fill in all k moments directly as entry-wise products of the moments of y_i and y_j. Otherwise, they are dependent, so we use the Cauchy-Schwarz inequality (see Lemma 1): $-\sqrt{\mathbb{E}(y_i^{2p})\mathbb{E}(y_j^{2p})} \leq \mathbb{E}(y_{ij}^p) \leq \sqrt{\mathbb{E}(y_i^{2p})\mathbb{E}(y_j^{2p})}$, and the interval approximation $\mathbb{E}(y_{ij}^p) \in I_{ij}^p$.

Continuous Functions: Let $g(\boldsymbol{y})$ be a continuous and $(m+1)$-times differentiable function of $\boldsymbol{y}$. The Taylor expansion of g around a point $\boldsymbol{y}_0$ allows us to approximate g as a polynomial.

$$g(\boldsymbol{y}) = g(\boldsymbol{y}_0) + Dg(\boldsymbol{y}_0)(\boldsymbol{y} - \boldsymbol{y}_0) + \sum_{2 \leq |\alpha|_1 \leq m} \frac{D^\alpha g(\boldsymbol{y}_0)(\boldsymbol{y} - \boldsymbol{y}_0)^\alpha}{\alpha!} + R_g^{m+1},$$

wherein Dg denotes the vector of partial derivatives $(\frac{\partial g}{\partial y_j})_{j=1,\ldots,n}$, $\alpha : (d_1, \ldots, d_n)$ ranges over all vector of indices where $d_i \in \mathbb{N}$ is a natural number, $|\alpha|_1 : \sum_{i=1}^n d_i$, $\alpha! = d_1!d_2! \cdots d_n!$, $D^\alpha g$ denotes the partial derivative $\frac{\partial^{d_1} g \cdots \partial^{d_n} g}{\partial y_1^{d_1} \cdots \partial y_n^{d_n}}$ and $(\boldsymbol{y} - \boldsymbol{y}_0)^\alpha : \prod_{j=1}^n (y_j - y_{0,j})^{d_j}$. Finally, R_g^{m+1} is an interval valued *Lagrange remainder*. Since we have discussed sums and products of affine forms, the Taylor approximation may be evaluated entirely using affine forms.

The remainder is handled using a fresh noise symbol $y_g^{(m+1)}$. Its set of support is R_g^{m+1} and moments are estimated based on this interval. The newly added noise symbol is functionally dependent on all variables $\boldsymbol{y}$ that appear in $g(\boldsymbol{y})$. These dependencies are added to the graph G.

The Taylor expansion allows us to approximate continuous functions including rational functions and trigonometric functions of these random variables.

Example 2. We illustrate this by computing the sine of an affine form. Let y_1 be a noise symbol over the interval $[-0.2, 0.2]$ with the moments $(0, [0.004, 0.006], 0, [6 \times 10^{-5}, 8 \times 10^{-5}], 0)$. We consider the form $\sin(y_1)$. Using a Taylor series expansion around $y_1 = 0$, we obtain

$$\sin(y_1) = y_1 - \frac{1}{3!}y_1^3 + [-1.3 \times 10^{-5}, 1.4 \times 10^{-5}].$$

We introduce a fresh variable y_2 to replace y_1^3 and a fresh variable y_3 for the remainder interval $I_3 : [-1.3 \times 10^{-5}, 1.4 \times 10^{-5}]$.

Dependence: We add the edges (y_2, y_1) and (y_3, y_1) to G.
Set of Support: $I_2 : [-0.008, 0.008]$ and $I_3 : [-1.3 \times 10^{-5}, 1.4 \times 10^{-5}]$.
Moments: $\mathbb{E}(y_2) = \mathbb{E}(y_1^3) = 0$. Further moments are computed using interval arithmetic. The moment vector $I(m_2)$ is $(0, [0, 64 \times 10^{-6}], [-512 \times 10^{-9}, 512 \times 10^{-9}], \ldots)$. For y_3, the moment vector $I(m_3)$: $(I_3, \text{square}(I_3), \text{cube}(I_3), \ldots)$.

The resulting affine form for $\sin(y_1)$ is $[1, 1]y_1 - [0.16, 0.17]y_2 + [1, 1]y_3$.

3.3 Approximating Computations Using Affine Forms

Having developed a calculus of affine forms, we may directly apply it to propagate uncertainties across straight-line computations. Let $X = \{x_1, \ldots, x_p\}$ be a set of *program variables* collectively written as $\boldsymbol{x}$ with an initial value $\boldsymbol{x}_0$. Our semantics consist of a tuple $(\mathcal{E}, \eta)$ wherein $\mathcal{E}$ is an environment and $\eta : X \to \mathsf{AffineForms}(\mathcal{E})$ maps each variable $x_i \in X$ to an affine form over $\mathcal{E}$.

The initial environment $\mathcal{E}_0$ has no noise symbols and an empty dependence graph. The initial mapping η_0 associates each x_i with the constant $x_{i,0}$. The basic operations are of two types: (a) assignment to a fresh random variable, and (b) assignment to a function over existing variables.

Random Number Generation: This operation is of the form $x_i := \mathsf{rand}(I, \mathbf{m})$, wherein I denotes the set of support interval for the new random variable, and $\mathbf{m}$ denotes a vector of moments for the generated random variable. The operational rule is $(\mathcal{E}, \eta) \xrightarrow{x_i := \mathsf{rand}(I, \mathbf{m})} (\mathcal{E}', \eta')$, wherein the environment $\mathcal{E}'$ extends $\mathcal{E}$ by a fresh random variable y whose set of support is given by I and moments by $\mathbf{m}$. The dependence graph is extended by adding a new node corresponding to y but without any new edges since freshly generated random numbers are assumed independent. However, if the newly generated random variable is dependent on some previous symbols, such a dependency is also easily captured in our framework.

Assignment: The assignment operation is of the form $x_i := g(\boldsymbol{x})$, assigning x_i to a continuous and $(j+1)$-times differentiable function $g(\boldsymbol{x})$. The operational rule has the form $(\mathcal{E}, \eta) \xrightarrow{x_i := g(\boldsymbol{x})} (\mathcal{E}', \eta')$. First, we compute an affine form f_g that approximates the function $g(\eta(x_1), \ldots, \eta(x_n))$. Let Y_g denote a set of fresh symbols generated by this approximation with new dependence edges E_g. The environment $\mathcal{E}'$ extends $\mathcal{E}$ with the addition of the new symbols Y_g and and new dependence edges E_g. The new map is $\eta' : \eta[x_i \mapsto f_g]$.

Let $\mathcal{C}$ be a computation defined by a sequence of random number generation and assignment operations. Starting from the initial environment $(\mathcal{E}_0, \eta_0)$ and applying the rules above, we obtain a final environment $(\mathcal{E}, \eta)$. However, our main goal is to answer *queries* such as $\mathbb{P}(x_j \in I_j)$ that seek the probability that a particular variable x_j belongs to an interval I_j. This directly translates to a query involving the affine form $\eta(x_j)$ which may involve a prohibitively large number of noise symbols that may be correlated according to the dependence graph G.

4 Concentration of Measure Inequalities

We present the use of concentration of measure inequalities to bound probabilities of the form $\mathbb{P}(f \geq c)$ and $\mathbb{P}(f \leq c)$. Let f be an affine form in an environment $\mathcal{E}$.

There are numerous inequalities in probability theory that provide bounds on the probability that a particular function of random variables deviates "far"

from its expected value [13]. Let $X_1, \ldots, X_n$ be a sequence of random variables that may be pairwise independent or depend on each other according to a probabilistic dependence graph $\widehat{G}$. Consider their sum $X : \sum_{j=1}^{n} X_j$ and its expected value $\mathbb{E}(X) : \sum_{j=1}^{n} \mathbb{E}(X_j)$. Under numerous carefully stated conditions, the sum "concentrates" around its average value so that the "tail" probabilities: the right tail probability $\mathbb{P}(X - \mathbb{E}(X) \geq t)$ of the sum being $t > 0$ to the right of the expectation, or the left "tail" probability $\mathbb{P}(X - \mathbb{E}(X) \leq -t)$ are bounded from above and rapidly approach zero as $t \to \infty$. We note that concentration of measure inequalities provide valid bounds on large deviations. In other words, they are more powerful than asymptotic convergence results, although they are typically used to prove convergence. A large category of concentration of measure inequalities conform to the sub-gaussian type below.

Definition 4 (Sub-Gaussian Concentration of Measure). *Let $X_1, \ldots, X_n$ be a set of random variables wherein each X_i has a compact set of support in the interval $[a_i, b_i]$. A sub-gaussian type concentration of measure inequality is specified by two parts: (a) a condition Ψ on the dependence structure between the random variables X_i, and (b) a constant $c > 0$. The inequality itself has the following form for any $t \geq 0$,*

$$\mathbb{P}(X - \mathbb{E}(X) \geq t) \leq exp\left(\frac{-t^2}{c\sum_{j=1}^{n}(b_i - a_i)^2}\right).$$

The expression for the left tail probability is derived identically.

In general, many forms of these inequalities exist under various assumptions. We focus on two important inequalities that will be used here.

Chernoff-Hoeffding: The condition Ψ states that $X_1, \ldots, X_n$ are independent. Alternatively, the probabilistic dependence graph $\widehat{G}$ does not have any edges. In this situation, the inequality applies with a constant $c = \frac{1}{2}$.

Chromatic Number-Based: Janson generalizes the Chernoff-Hoeffding inequality using the chromatic number of the graph $\widehat{G}$ [22]. Let $\chi(\widehat{G})$ be an upper bound on the minimum number of colors required to color $\widehat{G}$ (i.e., it's chromatic number). The condition Ψ states that the random variables depend according to $\widehat{G}$. In this situation, the inequality applies with a constant $c = \frac{\chi(\widehat{G})}{2}$. For the independent case, $\chi(\widehat{G}) = 1$ and thus, Chernoff-Hoeffding bounds are generalized.

The sub-gaussian bounds depend on the range $[a_i, b_i]$ of the individual random variables. Often, the variance σ_i^2 of each random variable is significantly smaller. In such situations, the Bernstein inequality provides useful bounds.

Theorem 1 (Bernstein Inequality). *Let $X_1, \ldots, X_n$ be independent random variables such that (a) there exists a constant $M > 0$ such that $|X_i - \mathbb{E}(X_i)| \leq M$ for each $i \in [1, n]$, and (b) the variance of each X_i is σ_i^2. For any $t \geq 0$:*

$$\mathbb{P}(X - \mathbb{E}(X) \geq t) \;\leq\; exp\left(\frac{-t^2}{2\sum_{i=1}^{n}\sigma_i^2 + \frac{2}{3}Mt}\right)$$

For the left tail probability, we may derive an identical bound.

We now illustrate how these inequalities can be used for the motivating example from Sect. 2. Let $\mathcal{E}$ be an environment and $f(\boldsymbol{y}) : a_0 + \sum_{i=1}^{n} a_i y_i$ be an affine form involving noise symbols $\boldsymbol{y}$.

Chromatic Number-Based Inequality: The application of Janson's dependent random variable inequality requires the following pieces of information: (a) An upper bound on the chromatic number of the graph $\chi(\widehat{G})$. While the precise chromatic number is often hard to compute, it is often easy to estimate upper bounds. For instance, $\chi(\widehat{G}) \leq 1 + \Delta$ wherein Δ is the maximum degree of any node in $\widehat{G}$. (b) We compute the expectation $I_E : \mathbb{E}(f(\boldsymbol{y}))$ by summing up the expectations of the individual terms. (c) Next, for each term $a_i y_i$, we compute its set of support $[c_i, d_i] := a_i I_i$ wherein I_i is the range of the noise symbol y_i in $\mathcal{E}$. Specifically, we calculate $C : \sum_{i=1}^{n} (d_i - c_i)^2$.

Since the expectation I_E is an interval, we apply the concentration of measure inequality using the upper bound of I_E for right tail inequalities and the lower bound for the left tail inequalities.

Example 3. Continuing the affine form in the 2D robotic effector model in Fig. 1, we compute the relevant constants to enable our application of the dependent random variable inequality.

The chromatic number $\chi(\widehat{G}) \leq 4$. The sum $C : \sum_{i=1}^{n} (d_i - c_i)^2$ was calculated as 12.2642. The expectation lies in the range $[265.9, 268.9]$. Combining, we obtain the concentration of measure inequalities: $\mathbb{P}(f \geq 268.9 + t) \leq \exp\left(\frac{-t^2}{24.53}\right)$ Similarly, $\mathbb{P}(f \leq 265.9 - t) \leq \exp\left(\frac{-t^2}{24.53}\right)$.

$f \leq 220$	$f \leq 235$	$f \leq 250$	$f \leq 260$	$f \geq 275$	$f \geq 285$	$f \geq 295$	$f \geq 310$
4.2E−35	1.2E−13	5E−5	0.48	0.21	2.2E−7	7E−13	9.2E−31

Applying Chernoff-Hoeffding and Bernstein Inequalities: The Bernstein inequality and Chernoff-Hoeffding bounds require independence of the random variables in the summation. However, the noise symbols involved in $f(\boldsymbol{y})$ may be dependent.

Suppose we compute the maximal strongly connected components (MSCC) of the graph $\widehat{G}$. Note that symbols that belong to different MSCCs are mutually independent. As a result, we decompose a given affine form $f(\boldsymbol{y})$ into *independent clusters* as $f(\boldsymbol{y}) : f_1(\boldsymbol{y}_1) + \cdots + f_k(\boldsymbol{y}_k)$. Each cluster corresponds to an affine form $f_j(\boldsymbol{y}_j)$ over noise symbols $\boldsymbol{y}_j$ involved in the j^{th} MSCC of $\widehat{G}$. Note that each f_i itself will be independent of f_k for $k \neq i$. Thus, we may apply the Chernoff-Hoeffding bounds or the Bernstein inequality by treating each $f_j(\boldsymbol{y}_j)$ as a summand. Let $[\ell_j, u_j]$ represent the set of support for each cluster affine form $f_j(\boldsymbol{y}_j)$. To apply the Chernoff-Hoeffding bounds, we compute $C : \sum_{j=1}^{k} (u_j - \ell_j)^2$.

To apply the Bernstein inequality, we collect the information on the variance σ_j^2 of each f_j and compute M as $\max_{j=1}^{n} (|u_j - \mathbb{E}(f_j)|)$. The environment $\mathcal{E}$ tracks the required information to compute $\sigma^2 : \sum_{j=1}^{n} \sigma_j^2$ and M, respectively.

Since the variance is estimated over an interval, when we apply the Bernstein inequality, we always use the upper bound on σ^2.

Example 4. We illustrate our ideas on the example from Fig. 1. For Chernoff-Hoeffding bounds, the original form with nearly 6900 variables yields about 3000 clusters. The value of C is 17.027. Combining, we obtain the concentration of measure inequalities: $\mathbb{P}(f \geq 268.9 + t) \leq \exp\left(\frac{-t^2}{8.5138}\right)$ for the right tail and $\mathbb{P}(f \leq 265.9 - t) \leq \exp\left(\frac{-t^2}{8.5138}\right)$ for the left tail. This yields much improved bounds when compared to the bounds in Example 3.

$f \leq 220$	$f \leq 235$	$f \leq 250$	$f \leq 260$	$f \geq 275$	$f \geq 285$	$f \geq 295$	$f \geq 310$
$2.5E{-}108$	$2E{-}49$	$1.1E{-}13$	0.016	0.21	$4E{-}14$	$1E-35$	$3E{-}87$

Applying the Bernstein inequality, we note that $\sigma^2 \in [0.1699985951, 0.2292648934]$ and $M = \max(|f_i - \mathbb{E}(f_i)|) = 0.1035521711$.

$f \leq 220$	$f \leq 235$	$f \leq 250$	$f \leq 260$	$f \geq 275$	$f \geq 285$	$f \geq 295$
$5E{-}253$	$9E{-}161$	$2.6E{-}71$	$4E{-}18$	$4.2E{-}19$	$1.8E{-}72$	$2E{-}223$

In particular, we obtain the result in Sect. 2: $\mathbb{P}(X \geq 272) \leq 6.2E{-}7$.

Finally, it is sometimes seen that the value of M in Bernstein inequality is large but the value of σ^2 lies inside a small range. In such a situation, Chebyshev inequalities are easy to apply and prove tight bounds.

Theorem 2 (Chebyshev-Cantelli Inequality). *For any random variable X, $\mathbb{P}(X - \mathbb{E}(X) \geq k\sigma) \leq \frac{1}{1+k^2}$. A similar inequality holds for the right tail, as well.*

Handling Unbounded Random Variables: Finally, we mention a simple trick that allows us to bound random variables with distributions such as the normal or the exponential.

Suppose the truncated Gaussian distributions in lines 3, 4 and 12 of the program in Fig. 1 are all replaced by normal random variables. The concentration of measure inequalities no longer apply directly. However, for most distributions the probability of a large deviation from the mean is easily computed. For instance, it is known that for a normally distributed variable X with mean μ and standard deviation σ, $\mathbb{P}(|X - \mu| \geq 5\sigma) \leq 6 \times 10^{-7}$. Therefore, we simply truncate the domain of each such random variable to $[\mu - 5\sigma, \mu + 5\sigma]$ and simply add $6K \times 10^{-7}$ to any probability upper bound, wherein K is the number of times a Gaussian random variable is generated. Similar bounds can be obtained for other common distribution types. Even if the distribution is not known but its mean and variance are provided, a weaker Chebyshev inequality bound can be derived: $\mathbb{P}(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}$.

Example 5. If the random variable in line 12 of Fig. 1 were a normally distributed variable with $\sigma = 0.01$, we note that 1500 such variables are generated during the computation. The result from Example 4 is updated as $\mathbb{P}(X \geq 272) \leq 6.2 \times 10^{-7} + 1500 \times 6 \times 10^{-7} \leq 9.0062 \times 10^{-4}$.

5 Experiments

In this section, we report on an experimental evaluation of our ideas and a comparison the p-box based implementation of Bouissou et al. [7], wherever possible.

Implementation: Our prototype analyzer is built as a data-type in C++ on top of the boost interval arithmetic library with overloaded operators that make it easy to carry out sequences of computations. Our implementation includes support for nonlinear trigonometric operators such as sine and cosine. It tracks the expectation and second moments of noise symbols. Currently, we do not explicitly account for floating point/round off errors. However, as future work, we will integrate our work inside the Fluctuat analysis tool that has a sophisticated model of floating point errors [20]. The dependency G and probabilistic dependency $\widehat{G}$ graphs are maintained exactly as described in Sect. 3. All concentration of measure inequalities presented in Sect. 4 have been implemented.

Table 1 reports on the results from our prototype on a collection of interesting examples taken from related work : Ferson [2], Filter [2], Tank [2], CartPole [36], Tumor [6], RmlsWhl [36], Anesthesia [28] as well as new examples for this domain: DblWell, Euler, Arm2D, Steering. We present for each example, the number of instructions including the random variables

Table 1. Experimental results at a glance: †: indicates a nonlinear example, #ins: total number of instructions, #rv: random variable generator calls, **n**: number of noise symbols, $\mathbf{T_{aff}}$: Time (seconds) to generate affine form, $\mathbf{T_{cmi}}$: Time (seconds) to perform concentration of measure inequality, χ: Chromatic number of the probabilistic dependence graph $\hat{G}$, #scc: number of strongly connected components, Jan.: Jansen 2004, c-h.: Chernoff-Hoeffding, Bern.: Bernstein inequality, Cheb. Chebyshev inequality.

ID	#ins	#rv	**n**	$\mathbf{T_{aff}}$	$\mathbf{T_{cmi}}$	χ	#scc	End of Range Probability			
								jan.	c-h.	bern.	cheb.
Ferson †	20	2	20	<0.1	<0.1	19	2	0.95	0.55	0.78	1
Filter	182	32	32	<0.1	<0.1	1	32	0.2	0.2	0.1	0.1
Tank	78	52	52	<0.1	<0.1	1	52	5E-12	5E-12	5E-21	1E-4
CartPole †	180	40	164	0.2	<0.1	92	71	0.94	0.30	0.09	2.5E-4
Tumor †	400	100	200	2.7	0.1	200	1	0.94	0.65	0.31	0.05
DblWell†	400	100	200	<0.1	<0.1	99	102	0.95	0.63	0.43	0.34
Euler	3K	1K	1K	2.7	0.1	1	1K	1E-217	1E-217	3E-620	1E-8
Arm2D†	4K	2K	6.9K	5.8	9.5	5	3.1K	3E-44	3E-160	1.1E-309	1E-4
RmlsWhl †	6K	2K	3K	7.4	2.7	3	1K	0.32	0.07	0.02	0.03
Steering†	11.3K	45	4.5K	3	22	2.9K	1.5K	0.993	0.599	0.224	0.016
Anesthesia	22.4K	5.6K	5.6K	438.2	12.2	1	5.6K	9E-19	9E-19	3E-26	0.006

involved. Note that for all but one example (FERSON), this number ranges from many tens of random variables to many thousands. We also report on the number of noise symbols involved in our affine forms. Finally, the times to derive the affine form and analyze it using concentration of measure inequalities (CMI) are reported. To evaluate the performance of various CMIs at a single glance, we simply compare the probability bounds that each CMI provides for the affine form taking a value past its upper or lower bound. This probability should ideally be zero, but most CMIs will ideally report a small value close to 0. We note that Bernstein inequality is by far the most successful, thanks to our careful tracking of higher order moments as part of the affine form. The overestimation of chromatic number makes the Jansen inequality much less effective than Chernoff-Hoeffding bounds. However, for the STEERING and TUMOR examples, we find that CMIs do not yield bounds close to zero, whereas we still obtain small bounds through Chebyshev inequality. We now highlight a few examples, briefly. A detailed description of each benchmark is provided in the Appendix.

Comparison with p-Boxes: We directly compared our approach with the previous work of Adjé et al. on three reported examples: FERSON, TANK and FILTER [2]. At this stage, we could not handle any of the other examples using that prototype.

The FERSON example uses a large degree 5 polynomial $p(\theta_1, \theta_2)$ over two random variables θ_1, θ_2. In this example, Adjé et al. obtain a much smaller range of $[1.12, 1.17]$ for p due to the subdivisions of the domain of θ_1, θ_2. In contrast, our tool reports a range of $[1.05, 1.21]$. Our approach produces a relatively narrow bound on the expectation of p and is able to conclude that $\mathbb{P}(p \leq 1.13) \leq 0.5$. However, they report a much more precise bound of 0.05 for the same probability. This suggests that subdividing random variables can indeed provide us more precision. In contrast, our running time is roughly 0.01 s while Bouissou et al. report a running time of nearly 100 s.

The TANK example considers the process of filling a tank using noisy tap and measurement devices. In this example, Adjé et al. bound the probability that the tank does not fill within 20 iterations as 0.63. In fact, our approach bounds the same probability by 0.5. Likewise, they incorrectly report that the tank will always fill within 26 iterations. Our approach correctly proves a bound of at most 10^{-6} on the probability that the tank is not full. A simple calculation also reveals that this probability is tiny but non-zero.

Finally, we compare the filter example wherein the affine form is obtained as a linear combination of independent random variables. Bouissou et al. [7] analyze the same example and report probability bounds for the assertion $y \leq -1$ as $\mathbb{P}(y \leq -1) \leq 0.16$. Our approach on the other hand finds a bound of 0.5 for the same assertion. The difference here is a pitfall of using concentration of measure inequalities which ignore characteristics of the underlying distributions of the noise symbol. Our approach is quite fast taking less than 0.01 s whereas depending on the number of subdivisions, Bouissou et al. report between 1 s to 5 min.

We now consider models that could not be attempted by the P-Box implementation.

Anesthesia Model: The anesthesia model consists of a four chamber pharmacokinetic model of the anesthetic fentanyl that is administered to a surgical patient using an infusion pump [28]. This model is widely used as part of automated anesthesia delivery systems [34]. As part of this process, we model an erroneous infusion that results in varying amounts of anesthesia infused over time as truncated gaussian random noise. The target state variable x_4 measures the concentration of anesthesia in the blood plasma. The goal is to check the probability that the infusion errors result either in too much anesthesia $x_4 \geq 300ng/mL$ potentially causing loss of breathing or too little anesthesia $x_4 \leq 150ng/mL$ causing consciousness during surgery. Our approach bounds the probability $\mathbb{P}(x_4 \geq 300) \leq 7 \times 10^{-13}$ and $\mathbb{P}(x_4 \leq 150) \leq 10^{-23}$. These bounds guarantee that small infusion errors alone have a very small probability of causing safety violations.

Tumor Model: We examine a stochastic model of tumor growth with immunization [6]:

$$x_{n+1} = x_n + \delta(ax_n - (b_0 + \frac{\beta}{1+x^2})x^2 + xw_n)\,,$$

where x_n denotes the fraction of tumor cells at time $t = n\delta$. We use $a = b_0 = \beta = 1$ and w as a truncated normal random variable with mean 0, variance $\sigma^2 = \delta$ and range $[-10\sigma, 10\sigma]$. We ask for the probability that $x_{100} \geq 0.6$, and obtain a Chebyshev inequality bound $\mathbb{P}(x_{100} \geq 0.6) \leq 0.405$. Note that, the structure of the model leads to a situation wherein all noise symbols in our final form end up depending on each other.

Rimless Wheel Model: The rimless wheel model, taken from Tedrake et al. [36], models a wheel with spokes but no rims rolling down a slope. Such models are used as human gait models in robotics. Details of the model are given in the appendix. As part of this model, we wish to verify whether $P(x_{1000} \leq 0) \leq 0.5$. Our approach proves a bound of 0.07 on this probability, verifying the property.

6 Conclusion and Future Work

Thus far, we have presented a tractable method for answering queries on probabilities of assertions over program variables, using a combination of set-based methods (affine forms), moment propagation and concentration of measure inequalities. We showed that this method often yields precise results in a very (time and space) efficient manner, especially when tracking rare events. However, we also documented failures of this approach on some examples.

As part of the future work, we are considering extensions to programs with conditional branches and the use of concentration of measure inequalities on higher order moments. We are exploring possible improvements to our approach using the so-called "moment problem" [27].

Acknowledgments. This work was partially supported by the US NSF under award number 1320069, and the academic research chair "Complex Systems Engineering" of Ecole polytechnique, Thalès, FX, DGA, Dassault Aviation, DCNS Research, ENSTA ParisTech, Télécom ParisTech, Fondation ParisTech and FDO ENSTA. All opinions involved are those of the authors and not necessarily of our sponsors.

References

1. Abate, A., Katoen, J., Lygeros, J., Prandini, M.: Approximate model checking of stochastic hybrid systems. Eur. J. Control **6**, 624–641 (2010)
2. Adje, A., Bouissou, O., Goubault-Larrecq, J., Goubault, E., Putot, S.: Static analysis of programs with imprecise probabilistic inputs. In: Cohen, E., Rybalchenko, A. (eds.) VSTTE 2013. LNCS, vol. 8164, pp. 22–47. Springer, Heidelberg (2014)
3. Auer, E., Luther, W., Rebner, G., Limbourg, P.: A verified matlab toolbox for the dempster-shafer theory. In: Workshop on the Theory of Belief Functions (2010)
4. Borges, M., Filieri, A., d'Amorim, M., Păsăreanu, C.S., Visser, W.: Compositional solution space quantification for probabilistic software analysis (2014)
5. Bornholt, J., Mytkowicz, T., McKinley, K.S.: Uncertain<T>: abstractions for uncertain hardware and software. IEEE Micro. **35**(3), 132–143 (2015)
6. Bose, T., Trimper, S.: Stochastic model for tumor growth with immunization. Phys. Rev. E **79**, 051903 (2009)
7. Bouissou, O., Goubault, E., Goubault-Larrecq, J., Putot, S.: A generalization of p-boxes to affine arithmetic. Computing **94**(2–4), 189–201 (2012)
8. Busaba, J., Suwan, S., Kosheleva, O.: A faster algorithm for computing the sum of p-boxes. J. Uncertain Syst. **4**(4), 244–249 (2010)
9. Chakarov, A., Sankaranarayanan, S.: Probabilistic program analysis with martingales. In: Sharygina, N., Veith, H. (eds.) CAV 2013. LNCS, vol. 8044, pp. 511–526. Springer, Heidelberg (2013)
10. Chistikov, D., Dimitrova, R., Majumdar, R.: Approximate counting in SMT and value estimation for probabilistic programs. In: Baier, C., Tinelli, C. (eds.) TACAS 2015. LNCS, vol. 9035, pp. 320–334. Springer, Heidelberg (2015)
11. Cousot, P., Monerau, M.: Probabilistic abstract interpretation. In: Seidl, H. (ed.) Programming Languages and Systems. LNCS, vol. 7211, pp. 169–193. Springer, Heidelberg (2012)
12. De Loera, J., Dutra, B., Koeppe, M., Moreinis, S., Pinto, G., Wu, J.: Software for Exact Integration of Polynomials over Polyhedra. ArXiv e-prints, July 2011
13. Dubhashi, D., Panconesi, A.: Concentration of Measure for the Analysis of Randomized Algorithms. Cambridge University Press, Cambridge (2009)
14. Durrett, R.: Probability. Theory and Examples. Wadsworth & Brooks/Cole, Belmont (1991)
15. Enszer, J., Lin, Y., Ferson, S., Corliss, G., Stadtherr, M.: Probability bounds analysis for nonlinear dynamic process models. AIChE J. **57**, 404–422 (2011)
16. Ferson, S.: RAMAS Risk Calc 4.0 Software: Risk Assessment with Uncertain Numbers. Lewis Publishers, Boca Raton (2002)
17. Ferson, S., Kreinovich, V., Ginzburg, L., Myers, D., Sentz, K.: Constructing probability boxes and Dempster-Shafer structures. Technical report SAND2002-4015, Sandia National Laboratories (2003)

18. Fuchs, M., Neumaier, A.: Potential based clouds in robust design optimization. J. Stat. Theo. Pract. **3**, 225–238 (2009)
19. Geldenhuys, J., Dwyer, M.B., Visser, W.: Probabilistic symbolic execution. In: ISSTA, pp. 166–176. ACM (2012)
20. Goubault, É., Putot, S.: Static analysis of numerical algorithms. In: Yi, K. (ed.) SAS 2006. LNCS, vol. 4134, pp. 18–34. Springer, Heidelberg (2006)
21. Goubault-Larrecq, J.: Continuous previsions. In: Duparc, J., Henzinger, T.A. (eds.) CSL 2007. LNCS, vol. 4646, pp. 542–557. Springer, Heidelberg (2007)
22. Janson, S.: Large deviations for sums of partly dependent random variables. Random Struct. Algorithms **24**(3), 234–248 (2004)
23. Jegourel, C., Legay, A., Sedwards, S.: Cross-entropy optimisation of importance sampling parameters for statistical model checking. In: Madhusudan, P., Seshia, S.A. (eds.) CAV 2012. LNCS, vol. 7358, pp. 327–342. Springer, Heidelberg (2012)
24. Jha, S.K., Clarke, E.M., Langmead, C.J., Legay, A., Platzer, A., Zuliani, P.: A bayesian approach to model checking biological systems. In: Degano, P., Gorrieri, R. (eds.) CMSB 2009. LNCS, vol. 5688, pp. 218–234. Springer, Heidelberg (2009)
25. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: verification of probabilistic real-time systems. In: Gopalakrishnan, G., Qadeer, S. (eds.) CAV 2011. LNCS, vol. 6806, pp. 585–591. Springer, Heidelberg (2011)
26. Lassaigne, R., Peyronnet, S.: Probabilistic verification and approximation. Ann. Pure Appl. Logic **152**(1–3), 122–131 (2008)
27. Lasserre, J.B.: Moments, Positive Polynomials and Their Applications. Imperial College Press Optimization Series, vol. 1. World Scientific, Singapore (2011)
28. McClain, D.A., Hug, C.C.: Intravenous fentanyl kinetics. Clin. Pharmacol. Ther. **28**(1), 106–114 (1980)
29. Monniaux, D.: Abstract interpretation of probabilistic semantics. In: Palsberg, J. (ed.) SAS 2000. LNCS, vol. 1824, pp. 322–339. Springer, Heidelberg (2000)
30. Neumaier, A.: Clouds, fuzzy sets and probability intervals. Reliable Comput. **10**(4), 249–272 (2004)
31. Rump, S.: INTLAB - INTerval LABoratory. In: Csendes, T. (ed.) Developments in Reliable Computing, pp. 77–104. Kluwer Academic Publishers, Berlin (1999)
32. Sankaranarayanan, S., Chakarov, A., Gulwani, S.: Static analysis for probabilistic programs: inferring whole program properties from finitely many paths. In: PLDI 2013, pp. 447–458. ACM Press (2013)
33. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton (1976)
34. Shafer, S.L., Siegel, L.C., Cooke, J.E., Scott, J.C.: Testing computer-controlled infusion pumps by simulation. Anesthesiology **68**, 261–266 (1988)
35. Shmarov, F., Zuliani, P.: Probreach: verified probabilistic delta-reachability for stochastic hybrid systems. In: HSCC 2015, pp. 134–139 (2015)
36. Steinhardt, J., Tedrake, R.: Finite-time regional verification of stochastic non-linear systems. Int. J. Robot. Res. **31**(7), 901–923 (2012)
37. Sun, J., Huang, Y., Li, J., Wang, J.M.: Chebyshev affine arithmetic based parametric yield prediction under limited descriptions of uncertainty. In: ASP-DAC, pp. 531–536. IEEE Computer Society Press (2008)
38. Terejanu, G., Singla, P., Singh, T., Scott, P.D.: Approximate interval method for epistemic uncertainty propagation using polynomial chaos and evidence theory. In: ACC 2010 (2010)

39. Williamson, R.C., Downs, T.: Probabilistic arithmetic: numerical methods for calculating convolutions and dependency bounds. J. Approximate Reasoning **4**(2), 89–158 (1990)
40. Xiu, D.: Numerical Methods for Stochastic Computation: A Spectral Method Approach. Princeton University Press, Princeton (2010)
41. Younes, H.L.S., Simmons, R.G.: Statistical probabilitistic model checking with a focus on time-bounded properties. Inform. Comput. **204**(9), 1368–1409 (2006)