SReachTools: A MATLAB Stochastic Reachability Toolbox*

Abraham P. Vinod, Joseph D. Gleason, and Meeko M. K. Oishi†

October 17, 2019

Abstract

We present `SReachTools`, an open-source MATLAB toolbox for performing stochastic reachability of linear discrete-time systems that are perturbed by additive Gaussian noise. The toolbox addresses the problem of stochastic reachability of a target tube, which also encompasses the terminal-time hitting reach-avoid and viability problems. The stochastic reachability of a target tube problem maximizes the likelihood that the state of a stochastic system will remain within a time-varying target tube for a give time horizon, while respecting the system dynamics and bounded control authority. `SReachTools` implements several new algorithms, based on convex optimization, computational geometry, and on Fourier transforms, to efficiently compute over- and under-approximations of the stochastic reach set. `SReachTools` can be used to perform probabilistic verification of closed-loop systems, by providing probabilistic guarantees of safety and performance. In addition, `SReachTools` can also perform controller synthesis to assure probabilistic safety or performance, via open-loop or affine controllers. The code base is designed to be extensible and user friendly.

Index terms— Reachability, Stochastic optimal control, linear systems

1 Introduction

Stochastic reachability analysis is an important tool for providing probabilistic verification of both performance and safety. It has been applied to a wide range of systems in which safety is paramount, such as spacecraft applications [1], stochastic motion planning problems [2], automated anesthesia delivery [3], and many others. `SReachTools` solves the problem of stochastic reachability of a target tube [4,5]. In [4], it is shown that this problem subsumes the terminal-time hitting stochastic reach-avoid problem [6] and the stochastic viability problems [7].

The problem of stochastic reachability of a target tube requires synthesis of an optimal control policy that maximizes the reach probability, the likelihood that the state will remain within a time-varying target tube (a time-indexed collection of sets) for the duration of a given time horizon, while respecting the system dynamics and bounded control authority. The stochastic reach set is the set of initial states from which an admissible controller exists, such that the reach probability is above a given threshold. `SReachTools` can compute over- and under-approximations of the stochastic reach set, as well as synthesize controllers that maximize the reach probability. Additionally, `SReachTools` can perform forward stochastic reachability analysis for uncontrolled linear systems [8] to characterize the stochasticity of the state at a future time of interest.

*This material is based upon work supported by the National Science Foundation under Grant Number IIS-1528047, CMMI-1254990 (Oishi, CAREER), CNS-1329878, and the Lighting Enabled Systems and Applications Engineering Research Center (EEC-0812056). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

†Abraham Vinod, Joseph Gleason, and Meeko Oishi are with Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM; e-mail: aby.vinod@gmail.com, gleasonj@unm.edu, oishi@unm.edu (corresponding author)

This work has been submitted to the 22nd ACM International Conference on Hybrid Systems: Computation and Control.

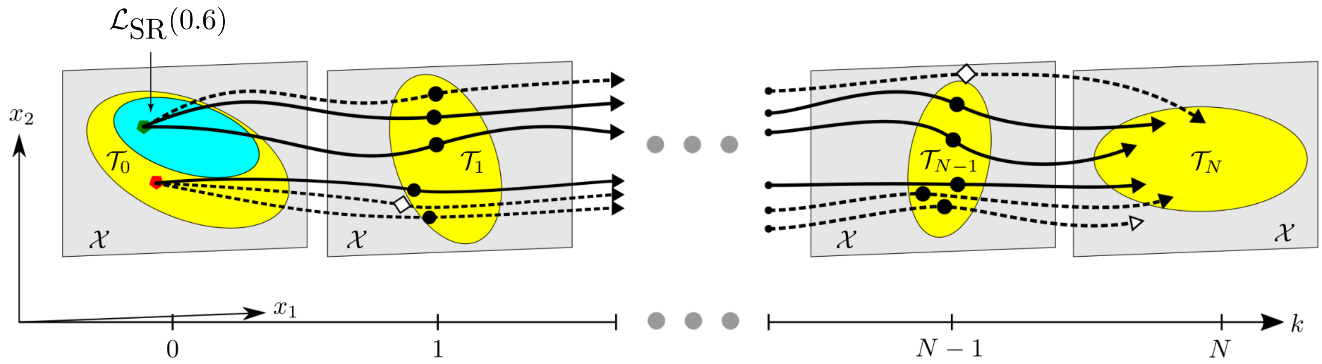

Figure 1: The target tube $\mathscr{T}_N$, the stochastic evolution of (1) or (2) under a maximal reach policy π^*, and the stochastic reach set $\mathcal{L}_{\mathrm{SR}}(\alpha)$ (equation (7)) for $\alpha = 0.6$. The stochastic reachability of a target tube problem subsumes the terminal hitting-time reach-avoid problem [6] ($\forall k \in \mathbb{N}_{[0,N-1]}, \mathcal{T}_i = \mathcal{S}, \mathcal{T}_N = \mathcal{R}$) and the viability problem [7] ($\forall k \in \mathbb{N}_{[0,N]}, \mathcal{T}_k = \mathcal{S}$) for Borel safe $\mathcal{S}$ and Borel terminal set $\mathcal{R}$.

An exact solution to the reachability of a target tube problem for nonlinear, discrete-time, hybrid systems is computationally difficult. Dynamic programming recursions, similar to [6, 7], can be used to obtain optimal feedback controllers for a discretized, finite state and input spaces. However, these recursions suffer from the well-known *curse of dimensionality*, which limits their use to systems of at most three or four dimensions. On the other hand, there are a host of approximation techniques for point-based verification and controller synthesis for linear systems that do not suffer from the curse of dimensionality, including chance constraints [1, 9], particle filters [1], and Fourier transform-based techniques [10]. In [4, 11], sufficient conditions were proposed for polytopic representations of stochastic reach sets, enabling computation of polytopic under-approximations of the stochastic reach sets. In addition, Lagrangian methods [5, 12] provide a set-theoretic approach based on computational geometry to under- and over-approximations of the stochastic reach set. `SReachTools` provides a user-friendly and extensible implementation of all these algorithms.

There are several related toolboxes that tackle the problem of stochastic verification and model-checking for systems with finite state and action spaces. The Modest Toolset [13], `FAUST`^{2} [14], `PRISM` [15], and `STORM` [16] provide high-level language or graphical tools for model-checking of continuous and discrete-time Markov chains. `FAUST`^{2}, handles continuous state and action spaces with a specified error via discretization of the state and action spaces. Barrier certificates have also been used with Linear Temporal Logic specifications for computing bounds on the reach probability for systems with continuous spaces and polynomial dynamics [3, 17]. The key advantage of `SReachTools` is that it solves stochastic reachability problems without using gridding, providing significant scalability. Further, `SReachTools` can perform open-loop and affine controller synthesis.

The toolbox is open-source, to allow for continued community driven improvement in a number of ways: 1) the number of pre-defined systems can expand as more users apply methods to various LTI/LTV systems; 2) new or updated versions of the solving algorithms can be added as more research in conducted; and 3) detection of code inefficiencies and non-trivial errors can be detected, reported, and fixed by community involvement.

The rest of the paper is organized as follows: Section 2 introduces notation and details the relevant systems and reachability problems. Section 3 describes features of the `SReachTools` toolbox including base components and an overview of the different solution methods. We demonstrate `SReachTools` on selected examples in Section 4 and conclude the work in Section 5.

2 Stochastic reachability

2.1 Notation

We denote the set of natural numbers, including zero, as $\mathbb{N}$, and discrete-time intervals as $\mathbb{N}_{[a,b]} = \mathbb{N} \cap \{a, a+1, \dots, b-1, b\}$, for $a, b \in \mathbb{N}$, $a \leq b$. The concatenation of a discrete-time series of vectors is denoted with a bar above the variable and subscripted indices, i.e. $\bar{x}_{[k,N]} = [x_k^\top, x_{k+1}^\top, \dots, x_N^\top]^\top$, $x_t \in \mathbb{R}^n$ for $t \in \mathbb{N}_{[k,N]}$. The n-dimensional identity matrix is denoted as I_n.

The Minkowski summation of two sets $\mathcal{S}_1, \mathcal{S}_2 \subseteq \mathbb{R}^n$ is $\mathcal{S}_1 \oplus \mathcal{S}_2 = \{s_1 + s_2 : s_1 \in \mathcal{S}_1, s_2 \in \mathcal{S}_2\}$; the Minkowski (or Pontryagin) difference of two sets $\mathcal{S}_1, \mathcal{S}_2$ is $\mathcal{S}_2 \ominus \mathcal{S}_1 = \{s : s + s_1 \in \mathcal{S}_2 \ \forall s_1 \in \mathcal{S}_1\}$. For $\mathcal{X} \subseteq \mathbb{R}^n, n \in \mathbb{N}, n > 0$, the indicator function corresponding to a set $\mathcal{S}$ is $\mathbf{1}_{\mathcal{S}} : \mathcal{X} \to \{0, 1\}$, where $\mathbf{1}_{\mathcal{S}}(x) = 1$ if $x \in \mathcal{S}$ and is zero otherwise; the Cartesian product of $\mathcal{S}$ with itself $k \in \mathbb{N}$ times is $\mathcal{S}^k$.

2.2 System formulation

A linear time-invariant system (LTI) is given by

$$x_{k+1} = Ax_k + Bu_k + Fw_k \tag{1}$$

where $x_k \in \mathcal{X} \subseteq \mathbb{R}^n$ is the state vector, $A \in \mathbb{R}^{n \times n}$, $u_k \in \mathcal{U} \subseteq \mathbb{R}^m$ is the input vector, $B \in \mathbb{R}^{n \times m}$, and $w_k \in \mathcal{W} \subseteq \mathbb{R}^q$ is the stochastic disturbance, $F \in \mathbb{R}^{n \times q}$. A linear time-varying system (LTV),

$$x_{k+1} = A_k x_k + B_k u_k + F_k w_k, \tag{2}$$

has all the relevant quantities equivalently defined as in (1) with time-varying system matrices. We assume w_k is absolutely continuous with a probability density function ψ_w and probability measure $\mathbb{P}_w$, and the disturbance process $\{w_k\}_{k=0}^{N-1}$ is an independent and identically distributed (i.i.d.) random process.

We denote an admissible universally-measurable state-feedback law as $\mu : \mathcal{X} \to \mathcal{U}$ and the set of Markov control policies $\pi = [\mu_0(\cdot), \mu_1(\cdot), \dots, \mu_{N-1}(\cdot)]$ as $\mathcal{M}$. We also define an affine control policy $\rho = (M, d) \in (\mathbb{R}^{mN \times qN}, \mathbb{R}^{mN})$ as an affine transformation of the concatenated disturbance vector $\overline{w}_{[0,N-1]} \in \mathcal{W}^N$ into a concatenated input space $\overline{u}_{[0,N-1]}$,

$$\overline{u}_{[0,N-1]} = M\overline{w}_{[0,N-1]} + d, \tag{3}$$

in which the affine disturbance feedback gain assures causality with the structure,

$$M = \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ M_{1,0} & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ M_{N-1,0} & \cdots & M_{N-1,N-2} & 0 \end{bmatrix} \tag{4}$$

and matrices $M_{i,j} \in \mathbb{R}^{m \times q}$. It is well-known that (3) is equivalent to a history-dependent linear state-feedback law [18]. However, unlike Markov policies, we can not impose hard control bounds on (3) for unbounded disturbances. Restricting M to be a zero matrix renders ρ to be an open-loop control policy, which is amenable to hard control bounds. Given an initial state $x_0 \in \mathcal{X}$ and a policy (π or ρ), the random vector $\overline{x}_{[1,N]}$ has a probability measure $\mathbb{P}^{\pi}_{\overline{x}_{[1,N]}}$ (or $\mathbb{P}^{\rho}_{\overline{x}_{[1,N]}}$, respectively), induced from $\mathbb{P}_w$ and dynamics (1) or (2).

As in [4,5], we define a target tube $\mathscr{T} = [\mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_N]$ as an indexed collection of subsets of the state space, $\mathcal{T}_i \subseteq \mathcal{X}$, for all $i \in \mathbb{N}_{[0,N]}$. We assign attributes of the tube that are typically given to sets, e.g. closed, bounded, compact, convex, etc., if and only if every set in the target tube has those properties.

2.3 Problem descriptions

We formally define the stochastic reachability problems `SReachTools` can solve. Given a target tube $\mathscr{T}$ be of length N, we define the function $V_{\mathrm{SR}} : \mathcal{X} \times \mathcal{M} \to [0,1]$ as

$$V_{\mathrm{SR}}(x_0, \pi) = \mathbf{1}_{\mathcal{T}_0}(x_0)\, \mathbb{P}^{\pi, x_0}_{\overline{x}_{[1,N]}}\{x_i \in \mathcal{T}_i,\, \forall i \in \mathbb{N}_{[1,N]}\}, \tag{5}$$

which can also be written as an expectation of a product of indicator functions [4–7].

2.3.1 Backward stochastic reachability of a target tube

The backward reachability problem (Figure 1) focuses on finding a Markov policy, π, that optimizes the value function (5), i.e.

$$V^*_{\mathrm{SR}}(z) = \underset{\pi \in \Pi}{\mathrm{maximize}}\, V_{\mathrm{SR}}(z, \pi). \tag{6}$$

The qualifier "backward" arises from the fact that we wish to ascertain properties of initial system states based on safety specifications defined for the future states [6, 7].

In many applications, we are also interested in $\mathcal{L}_{\mathrm{SR}}(\alpha)$, the set of initial states for which there exists a controller such that the probability of achieving the reach objective is greater than some threshold α. Specifically,

$$\begin{aligned}\mathcal{L}_{\mathrm{SR}}(\alpha) &= \{z \in \mathbb{R}^n \mid \exists \pi \in \Pi, V_{\mathrm{SR}}(z, \pi) \geq \alpha\} \\ &= \{z \in \mathbb{R}^n \mid V^*_{\mathrm{SR}}(z) \geq \alpha\}.\end{aligned} \tag{7}$$

To ensure safety guarantees, we desire either exact representations or assured under-approximations of $V^*_{\mathrm{SR}}(z)$ and $\mathcal{L}_{\mathrm{SR}}(\alpha)$.

2.3.2 Forward stochastic reachability of a target tube

Given an initial state, the forward stochastic reachability problem focuses on characterizing the stochasticity of the state x_k or the associated trajectory $\overline{x}_{[0,k]}$ for some $k > 0$ [8]. This analysis is done on either uncontrolled systems, $u_k = 0$ for all k, or systems with a fixed input policy, π. For these systems, the Markov decision process simplifies to a discrete-time continuous-state Markov chain, making the problem amenable to many toolboxes, e.g. `FAUST`[2], `PRISM`, and `STORM`, as well as `SReachTools`.

3 Features of `SReachTools`

The primary goal of `SReachTools` is to enable quick solutions of stochastic reachability problems without requiring exhaustive knowledge of the underlying algorithms or their implementation. Currently, available algorithms include: Dynamic Programming, Lagrangian techniques, Fourier Transform-based methods, Chance-Constrained Optimization, and Particle Filter-based methods. Since the toolbox is open source, more algorithms can be included as the research progresses. Table 1 shows the primary solving functions in `SReachTools`, the different solution methods available for each, and detail the utility of each.

3.1 Auxillary functions

`SReachTools` comes with a built-in initialization script `srtinit` and a broad set of unit-testing functions to ensure proper performance of the toolbox. The initialization script is designed to eliminate potential function-name overloading by only adding the `SReachTools` source functions to the MATLAB path when they are in use.

For easy handling of linear, discrete-time systems, we have developed `LtiSystem` and `LtvSystem` classes to define systems (1) and (2). We use the `Multi-Parameteric Toolbox (MPT3)` [19] to define polytopic input spaces and polytopic disturbance spaces, if required. SReachTools has several demonstration systems, such as a chain of integrators, Dubins car, and Clohessy-Wiltshire-Hill spacecraft near-orbit relative dynamics.

SReachTools also has the Tube class to define polytopic target tubes, and RandomVector class to define random vector disturbances or initial states. Tube is implemented using MPT3 [19] to define a collection of time-stamped polytopic safe sets. One can also easily specify a reach-avoid [6] or viability [7] specification using Tube. Currently, RandomVector supports only Gaussian disturbances, while development is underway to include non-Gaussian disturbances.

Using objects of Tube, RandomVector, and LtiSystem or LtvSystem classes, one can pose the problems of stochastic reachability discussed in Section 2.3. We presume for the remainder that sys refers to an instance of the LtiSystem or LtvSystem classes to describe (1) or (2) respectively, and tube refers to an instance of the Tube class to describe $\mathscr{T}$.

3.2 Stochastic reachability of a target tube: Dynamic Programming

The dynamic programming recursion for the stochastic reachability of target tube is a straightforward extension of the recursion known for stochastic viability and reach-avoid problems [6,7]. The recursion is given by

$$V_N^*(z) = \mathbf{1}_{\mathcal{T}_N}(z) \tag{8a}$$

$$V_k^*(z) = \sup_{u\in\mathcal{U}} \left(\mathbf{1}_{\mathcal{T}_k}(z) \int_{\mathcal{X}} V_{k+1}^*(y) Q_k(dy|z,a) \right), \tag{8b}$$

where Q_k is the stochastic transition kernel and $V_{\mathrm{SR}}^*(z) = V_0^*(x)$. For the linear time-varying system (2), $Q_k(dy|z,a) = \psi_{F_k w}(y - A_k z - B_k a)dy$ is a Gaussian probability density function. We can use SReachTools to implement a grid-based implementation of (8) using SReachDynProg.

```
% Dynamic programming solution with SReachTools
SReachDynProg(prob_str, sys, x_inc, u_inc, tube)
```

Here, prob_str refers to the type of reachability problem, x_inc and u_inc are grid step sizes for the state space $\mathcal{X}$ and input space $\mathcal{U}$ respectively. Currently in the SReachTools toolbox, prob_str must always be 'term', as version 1.0.0 is only designed for the problem of stochastic reachability of a target tube, i.e. the terminal-time hitting problem. Future releases will address other reachability problems, e.g. the first-time hitting problem [6].

3.3 Stochastic reachability of a target tube: Point-based computation

SReachTools can be used to approximate $V_{\mathrm{SR}}^*(z)$. To compute the maximal reach probability using an open-loop controller $d \in \mathcal{U}^N$ instead of a Markov policy π, we define the corresponding optimal value function as $V_{\mathrm{SR,open}}^* : \mathcal{X} \to [0,1]$, with

$$V_{\mathrm{SR,open}}^*(x_0) = \max_{d\in\mathcal{U}^N} \mathbf{1}_{\mathcal{T}_0}(x_0)\, \mathbb{P}_{\overline{x}_{[1,N]}}^{d,x_0}\{\forall i \in \mathbb{N}_{[1,N]},\ x_i \in \mathcal{T}_i\}. \tag{9}$$

Note that (9) is a log-concave optimization problem whenever the target tube is polytopic and the disturbance is log-concave [4, Thm. 6], Moreover, (9) under-approximates (6) [4, Thm. 7],

$$V_{\mathrm{SR}}^*(z) \geq V_{\mathrm{SR,open}}^*(z), \quad \forall z \in \mathcal{X}. \tag{10}$$

With an affine controller (M, d), we enforce a probabilistic constraint $\mathbb{P}_{\overline{u}_{[0,N-1]}}^{M,d}\{\overline{u}_{[0,N-1]} \in \mathcal{U}^N\} = \mathbb{P}_{\overline{w}_{[0,N-1]}}\{M\overline{w}_{[0,N-1]} + d \in \mathcal{U}^N\} \geq 1 - \Delta_U$ for some user-specified threshold $\Delta_U \in [0,1)$. Here, Δ_U is the maximum likelihood with which the affine controller (M, d) can violate the hard control bound. Similarly to (9), we define $V^\ast_{\mathrm{SR,affine}}(z)$ as the optimal value function of (11),

$$\begin{array}{rl} \underset{M,d}{\mathrm{max.}} & \mathbb{P}^{M,d,x_0}_{\overline{x}_{[1,N]}}\{\forall i \in \mathbb{N}_{[1,N]},\ x_i \in \mathcal{T}_i\} \\ \mathrm{s.t.} & \begin{cases} M, d \text{ as in (3)}, \quad x_0 = z \in \mathcal{T}_0 \\ \mathbb{P}_{\overline{w}_{[0,N-1]}}\{M\overline{w}_{[0,N-1]} + d \in \mathcal{U}^N\} \geq 1 - \Delta_U \end{cases} \end{array} \tag{11}$$

Given Δ_U, we have [9, Thm. 1]

$$V^\ast_{\mathrm{SR}}(z) \geq 1 - \frac{1 - V^\ast_{\mathrm{SR,affine}}(z)}{1 - \Delta_U}, \quad \forall z \in \mathcal{X}. \tag{12}$$

`SReachTools` can synthesize an open-loop controller by solving (9) using the following three approaches.

1. *Convex chance constraints*: An under-approximative reformulation, which is a linear program, via Boole's inequality and Gaussian vector properties [1, 9].

2. *Particle filter*: Mixed integer-linear reformulation via scenarios drawn from $\mathbb{P}^{d,x_0}_{\overline{x}_{[1,N]}}$ [1].

3. *Fourier transform (Genz's algorithm and* `patternsearch`*)*: An approximative reformulation via the characteristic function (Fourier transform of $\psi_{\overline{x}_{[0,N]}}$), integrated over the target tube. Genz's algorithm provides high-dimensional numerical quadrature [20, 21] with user-specified accuracy of ϵ_{genz}, and MATLAB's `patternsearch` [22] provides gradient-free optimization to solve $\mathbb{P}^{d,x_0}_{\overline{x}_{[1,N]}}\{\forall i \in \mathbb{N}_{[1,N]},\ x_i \in \mathcal{T}_i\}$ [10].

`SReachTools` can also synthesize an affine controller (3) by solving (11) using chance constraint reformulation and difference of convex programming [9]. `SReachTools` uses `CVX` [23] to implement all of the convex programs described in this section.

```
% Generate SReachTools options
options = SReachPointOptions(prob_str, method_str)
% Point-based stochastic target tube reachability
SReachPoint(prob_str, method_str, sys, init_state, tube, options)
```

Here, `method_str` refers to the optimization approach ('`chance-open`', '`particle-open`', '`genzps-open`', or '`chance-affine`'), `init_state` is the initial state $x_0 \in \mathcal{X}$ at which an approximation of $V^\ast_{\mathrm{SR}}(x_0)$ must be evaluated, and `options` enables specification of accuracy and method-specific arguments (i.e., Δ_U).

3.4 Stochastic reachability of a target tube: Set-based computation

3.4.1 Point-based computations and polytopic representations

Linear systems with polytopic target tube and input spaces, and disturbances with log-concave probability density functions, have a convex and compact stochastic reach set $\mathcal{L}_{\mathrm{SR}}(\alpha)$ for $\alpha \in (0, 1]$ [4, Thm. 4]. This is also true for the superlevel sets of $V^\ast_{\mathrm{SR,open}}(z)$ [4, Thm. 6],

$$\mathcal{K}_{\mathrm{SR}}(\alpha) = \{z \in \mathbb{R}^n \mid V^\ast_{\mathrm{SR,open}}(z) \geq \alpha\}. \tag{13}$$

Since $V^\ast_{\mathrm{SR}}(z) \geq V^\ast_{\mathrm{SR,open}}(z)$ for every $z \in \mathcal{X}$,

$$\mathcal{K}_{\mathrm{SR}}(\alpha) \subseteq \mathcal{L}_{\mathrm{SR}}(\alpha), \quad \forall \alpha \in [0, 1]. \tag{14}$$

Recall that convex and compact sets permit tight polytopic representations. `SReachTools` utilizes a line search algorithm and a point-based evaluation of stochastic reachability (see Section 3.3) to compute a tight polytopic under-approximation of $\mathcal{K}_{\mathrm{SR}}(\alpha)$ [4, 11].

Function	method_str	Utility
SReachPoint	chance-open	Under-approx. $V^\ast_{\mathrm{SR,open}}(z)$
	genzps-open	Approx. $V^\ast_{\mathrm{SR,open}}(z)$ within ϵ_{genz}
	particle-open	Approx. $V^\ast_{\mathrm{SR,open}}(z)$
	chance-affine	Under-approx. $V^\ast_{\mathrm{SR,affine}}(z)$
SReachSet	chance-open	Under-approx. $\mathcal{K}_{\mathrm{SR}}(\alpha)$
	genzps-open	Approx. $\mathcal{K}_{\mathrm{SR}}(\alpha)$ within ϵ_{genz}
	lag-open	Under-approx. $\mathcal{L}_{\mathrm{SR}}(\alpha)$
	lag-under	Over-approx. $\mathcal{L}_{\mathrm{SR}}(\alpha)$
SReachFwd	state-stoch	Stochasticity of x_k
	concat-stoch	Stochasticity of $\overline{x}_{[0,k]}$
	state-prob	$\mathbb{P}^{x_0}_{x_k}\{x_k \in \mathcal{T}\}$
	concat-prob	$\mathbb{P}^{x_0}_{\overline{x}_{[1,k]}}\{\forall i \in \mathbb{N}_{[1,k]},\ x_i \in \mathcal{T}_i\}$
SReachDyn	–	Dyn. prog.-based approx. of $V^\ast_{\mathrm{SR}}(z)$ and $\mathcal{L}_{\mathrm{SR}}(\alpha)$

Table 1: Current features of SReachTools. By (10), (12), and (14), we can characterize $V^\ast_{\mathrm{SR}}(z)$ and $\mathcal{L}_{\mathrm{SR}}(\alpha)$ using $V^\ast_{\mathrm{SR,open}}(z)$, $V^\ast_{\mathrm{SR,affine}}(z)$, and $\mathcal{K}_{\mathrm{SR}}(\alpha)$.

3.4.2 Lagrangian approach

This technique computes approximations to the stochastic reach set $\mathcal{L}_{\mathrm{SR}}(\alpha)$. Specifically, we compute the sets $\mathcal{C}_0(\mathscr{T},\mathcal{O})$, $\mathcal{D}_0(\mathscr{T},\mathcal{E}) \subseteq \mathcal{X}$ such that $\mathcal{D}_0(\mathscr{T},\mathcal{E}) \subseteq \mathcal{L}_{\mathrm{SR}}(\alpha) \subseteq \mathcal{C}_0(\mathscr{T},\mathcal{O})$. It uses a backward recursion that employs Minkowski sum, difference, intersection, and affine transformations [5, 12].

Lagrangian methods in SReachTools are implemented using polytopes with MPT3 [19]. Since we require operations that need both the vertex and the facet representation, this approach is subject to the computationally hard *vertex-facet enumeration problem*. Future releases of the toolbox will allow for zonotope-based computation of over-approximations which require only the facet representation, permitting scalable computations [5, 24].

```
% Generate SReachTools options
options = SReachSetOptions(prob_str, method_str)
% Compute stochastic reach sets with SReachTools
SReachSet(prob_str, method_str, sys, thresh, tube, options)
```

Here, `method_str` can be `'lag-under'` or `'lag-over'` for the Lagrangian under- and over-approximation, respectively; `'genz-open'` for an under-approximation using Genz algorithm; and `'chance-open'` for an under-approximation using chance-constrained optimization. The input `'thresh'` is the probabilistic bound, α, and `options` is used to specify additional solver options like direction vectors to use for the point-based stochastic reach set computations.

3.5 Forward stochastic reachability

SReachTools characterizes the mean and covariance of x_k and $\overline{x}_{[0,k]}$ using the fact that affine transformations of Gaussian random vectors are Gaussian [25, Sec. 9.2] (i.e., the prediction step of a Kalman filter). We employ Genz's algorithm [20, 21] for numerical evaluation of the high-dimensional quadrature of multivariate Gaussians over polytopes, up to a user-specified accuracy of ϵ_{genz}.

```
% Forward stochastic reachability via SReachTools
SReachFwd(prob_str, sys, init_state, k, [set/tube])
```

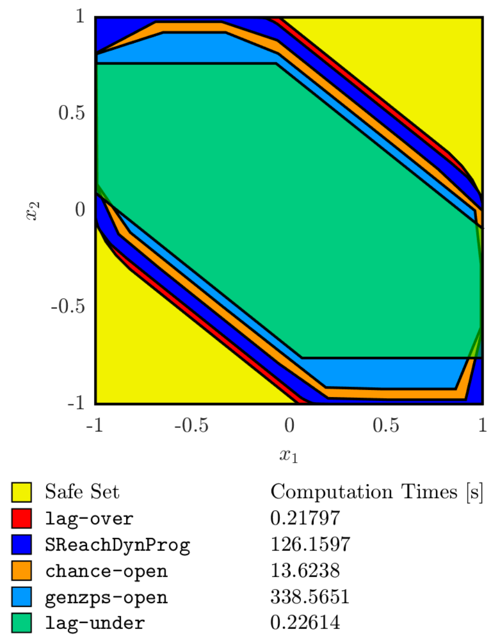

Figure 2: Comparison of $\boldsymbol{\mathcal{L}}_{\mathrm{SR}}(\mathbf{0.8})$ for a 2-dimensional double integrator computed via dynamic programming, and multiple `SReachSet` approximations. Tradeoffs between accuracy and computation time are typically problem dependent.

Table 1 describes various values `prob_str` can take. The input `init_state` takes in the initial state, which can be a deterministic vector or a random vector (an object of `RandomVector`). `SReachFwd` needs a `set` (target set $\mathcal{T}$) or `tube` (target tube $\mathscr{T}$), only if `prob_str` is '`state-prob`' or '`concat-prob`', respectively.

4 Numerical experiments

Numerical experiments were performed on a Intel Xeon CPU with 3.4GHz clock rate and 32 GB RAM running MATLAB R2017a.

4.1 Chain of integrators

This example demonstrates accuracy and scalability. A chain of integrators [4, Eq. (23)] is implemented to enable 1) comparison of the accuracy of `SReachSet` approximations of the backwards reach set with dynamic programming (with $m = 2$, Figure 2), as well as 2) scalability of high-dimensional calculations (Figure 3). Lagrangian methods are fast in low dimensions, but practically limited to $m \leq 7$ because of the vertex-facet enumeration problem. Open-loop methods, `genzps-open` and `chance-open`, scale well with dimension, but have a higher initial cost.

4.2 Dubin's vehicle with known turn rate

This example demonstrates `SReachTools`' ability to synthesize optimal controllers using `SReachPoint`. We consider the Dubin's vehicle with a known turn rate sequence [4, Eq. (26)], which simplifies the original nonlinear dynamics to an LTV system. We use a time-varying target tube with target sets $\mathcal{T}_k = \mathrm{Box}\left(c_k, 0.5\exp\left(\frac{-k}{100}\right)\right)$, boxes centered at a pre-defined trajectory c_k that gradually shrink in size as time progresses, to address trajectory regulation problems. Although the system is only 2-dimensional, dynamic programming is not feasible because of the extremely fine grid that would be required to capture the time-varying target set.

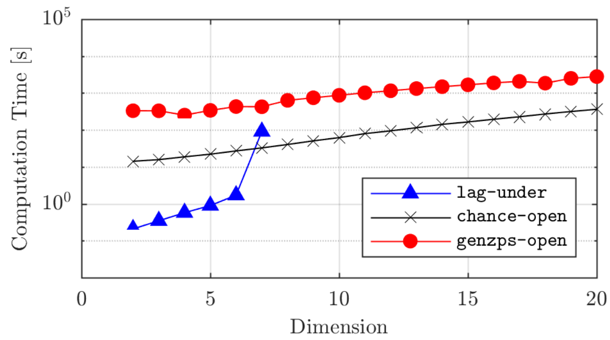

Figure 3: Scalability of `SReachSet` methods demonstrated on a chain of integrators, as state dimension increases.

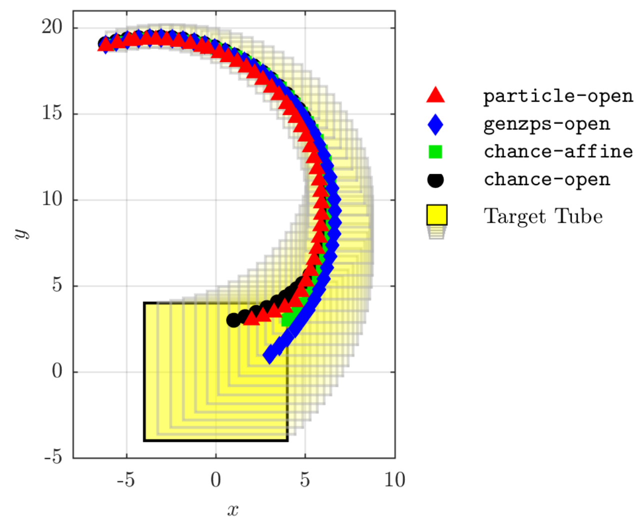

Figure 4: Mean trajectories of the Dubins' vehicle corresponding to the optimal controllers synthesized via `SReachPoint`. All trajectories report $\boldsymbol{V}^{*}_{\mathrm{SR}}(\cdot) \geq \mathbf{0.9}$.

Figure 4 shows the mean trajectory for each controller synthesis method available with `SReachPoint`. The computation times were 27.28 for `particle-open`, 527.85 for `genzps-open`, 279.11 for `chance-affine`, and 1.80 for `chance-open`.

4.3 Satellite rendezvous problem

This example demonstrates applicability of `SReachTools` to realistic viability problems. We apply methods in `SReachSet` and `SReachPoint` to a satellite rendezvous and docking problem. Probabilistic guarantees of safety are crucial in spacecraft systems because of the high cost of of failure. We use Clohessy-Wiltshire-Hill near-orbit spacecraft dynamics [1, Eqs. (1)–(2)] to describe an LTI system with a 4-dimensional state vector. Approximations of $\mathcal{L}_{\mathrm{SR}}(0.8)$ are computed with multiple `SReachSet` algorithms, and an optimal open-loop controller is synthesized using `SReachPoint` with `chance-open`. This controller was validated via Monte Carlo simulation (Figure 5).

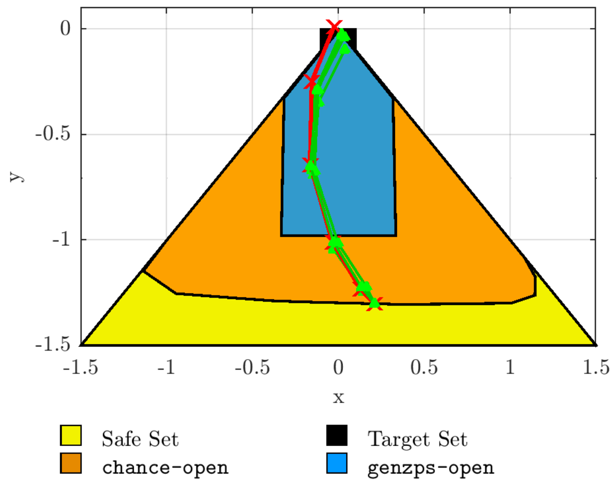

Figure 5: Satellite rendezvous problem with approximations of $\mathcal{L}_{\mathrm{SR}}(\mathbf{0.8})$ computed using `SReachSet`. We show the Monte-Carlo trajectories based on the controller from `SReachPoint` with `chance-open` for the initial state $\boldsymbol{x} = \mathbf{0.21}$, $\boldsymbol{y} = \mathbf{-1.30}$; good and bad trajectories are marked with '△' and '×' respectively.

5 Conclusion and Future Work

We describe `SReachTools`, an open-source MATLAB toolbox designed to solve the problem of stochastic reachability of a target tube. The toolbox is designed for ease of use and extensibility. Currently, the toolbox supports several methods, including Fourier transform-based methods, particle filtering, chance constrained optimization, Lagrangian techniques, and dynamic programming. These solutions can be computed for both linear time-varying and time-invariant discrete-time systems. The toolbox is extensible as well as open-source, enabling input from community contributions.

References

[1] K. Lesser, M. Oishi, and R. Erwin, "Stochastic reachability for control of spacecraft relative motion," in *IEEE Conference on Decision and Control*, December 2013.

[2] N. Malone, H. Chiang, K. Lesser, M. Oishi, and L. Tapia, "Hybrid dynamic moving obstacle avoidance using a stochastic reachable set-based potential field," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1124–1138, 2017.

[3] A. Abate, H. Blom, N. Cauchi, S. Haesaert, A. Hartmanns, K. Lesser, M. Oishi, V. Sivaramakrishnan, S. Soudjani, C. Vasile, and A. Vinod, "Arch-comp18 category report: Stochastic modelling," in *ARCH18. 5th International Workshop on Applied Verification of Continuous and Hybrid Systems*, vol. 54, 2018, pp. 71–103.

[4] A. Vinod and M. Oishi, "Stochastic reachability of a target tube: Theory and computation," *IEEE Transactions on Automatic Control*, 2018, (submitted). [Online]. Available: https://arxiv.org/abs/1810.05217v1

[5] J. D. Gleason, A. P. Vinod, and M. M. K. Oishi, "Lagrangian approximations for stochastic reachability of a target tube," online, 2018. [Online]. Available: https://arxiv.org/abs/1810.07118

[6] S. Summers and J. Lygeros, "Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem," *Automatica*, vol. 46, pp. 1951–1961, September 2010.

[7] A. Abate, M. Prandini, J. Lygeros, and S. Sastry, "Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems," *Automatica*, vol. 44, pp. 2724–2734, October 2008.

[8] A. Vinod, B. HomChaudhuri, and M. Oishi, "Forward stochastic reachability analysis for uncontrolled linear systems using Fourier transforms," in *Proceeding of the Hybrid Systems: Computation and Control*, 2017, pp. 35–44.

[9] A. Vinod and M. Oishi, "Affine controller synthesis for stochastic reachability via difference of convex programming," in *Proceeding of the Hybrid Systems: Computation and Control*, 2019, (submitted). [Online]. Available: https://hscl.unm.edu/affinecontrollersynthesis/

[10] ——, "Scalable underapproximation for the stochastic reach-avoid problem for high-dimensional LTI systems using Fourier transforms," *IEEE Control Systems Letters (L-CSS)*, vol. 1, no. 2, pp. 316–321, Oct 2017.

[11] ——, "Scalable underapproximative verification of stochastic LTI systems using convexity and compactness," in *Proceeding of the Hybrid Systems: Computation and Control*, Porto, Portugal, 2018.

[12] J. Gleason, A. Vinod, and M. Oishi, "Underapproximation of reach-avoid sets for discrete-time stochastic systems via Lagrangian methods," in *Proceedings of the IEEE Conference on Decision and Control*, Melbourne, Australia, December 2017.

[13] A. Hartmanns and H. Hermanns, "The modest toolset: An integrated environment for quantitative modelling and verification," in *Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 2014, pp. 593–598.

[14] S. Soudjani, C. Gevaerts, and A. Abate, "FAUST^2 : Formal abstractions of uncountable-STate STochastic processes," in *Proceedings of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, vol. 15, 2015, pp. 272–286.

[15] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM 4.0: Verification of probabilistic real-time systems," in *Proceedings of the International Conference on Computer Aided Verification*, ser. LNCS, G. Gopalakrishnan and S. Qadeer, Eds., vol. 6806. Springer, 2011, pp. 585–591.

[16] C. Dehner, S. Junges, J. Katoen, and M. Volk, "A storm is coming: A modern probabilistic model checker," in *International Conference on Computer Aided Verification*, 2017, pp. 592–600.

[17] P. Jagtap, S. Soudjani, and M. Zamani, "Temporal logic verification of stochastic systems using barrier certificates," in *International Symposium on Automated Technology for Verification and Analysis*, 2018.

[18] P. Goulart, E. Kerrigan, and J. Maciejowski, "Optimization over state feedback policies for robust control with constraints," *Automatica*, vol. 42, pp. 523–533, 2006.

[19] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proceedings of the European Control Conference*, Zürich, Switzerland, July 17–19 2013, pp. 502–510, http://people.ee.ethz.ch/%7Empt/3/.

[20] A. Genz, "Quadrature of a multivariate normal distribution over a region specified by linear inequalities: QSCMVNV," 2014. [Online]. Available: http://www.math.wsu.edu/faculty/genz/software/matlab/qscmvnv.m

[21] ——, "Numerical computation of multivariate normal probabilities," *J. of Comp. and Graph. Stat.*, vol. 1, no. 2, pp. 141–149, 1992.

[22] T. Kolda, R. Lewis, and V. Torczon, "Optimization by direct search: New perspectives on some classical and modern methods," *SIAM review*, vol. 45, no. 3, pp. 385–482, 2003.

[23] M. Grant and S. Boyd, "CVX: MATLAB software for disciplined convex programming," 2017. [Online]. Available: http://cvxr.com/cvx

[24] M. Althoff and B. H. Krogh, "Zonotope bundles for the efficient compu tation of reachable sets," in *Proceedings of the IEEE Confernce on Decision and Control*, Orlando, FL, USA, 2011.

[25] J. Gubner, *Probability and random processes for electrical and computer engineers.* Cambridge Univ. Press, 2006.