

SReachTools: A MATLAB Stochastic Reachability Toolbox*

Abraham P. Vinod
The University of New Mexico
Albuquerque, NM, USA
aby.vinod@gmail.com

Joseph D. Gleason
The University of New Mexico
Albuquerque, NM, USA
gleasonj@unm.edu

Meeko M. K. Oishi
The University of New Mexico
Albuquerque, NM, USA
oishi@unm.edu

ABSTRACT

We present SReachTools, an open-source MATLAB toolbox for performing stochastic reachability of linear, potentially time-varying, discrete-time systems that are perturbed by a stochastic disturbance. 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 collection of time-varying target sets for a given time horizon, while respecting the system dynamics and bounded control authority. SReachTools implements several new algorithms based on convex optimization, computational geometry, and 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 and can also perform controller synthesis via open-loop, affine, and state-feedback controllers. The code base is available online at <https://github.com/unm-hsc1/SReachTools>, and it is designed to be extensible and user friendly.

CCS CONCEPTS

• **General and reference** → **Verification**; • **Mathematics of computing** → **Stochastic control and optimization**; *Solvers*;

KEYWORDS

Reachability, Stochastic optimal control, Linear systems

ACM Reference Format:

Abraham P. Vinod, Joseph D. Gleason, and Meeko M. K. Oishi. 2019. SReachTools: A MATLAB Stochastic Reachability Toolbox. In *22nd ACM International Conference on Hybrid Systems: Computation and Control (HSCC '19)*, April 16–18, 2019, Montreal, QC, Canada. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3302504.3311809>

*This material is based upon work supported by the National Science Foundation under Grant Numbers IIS-1528047 and CMMI-1254990, and by the Air Force Research Laboratory under AFRL Grant No. FA9453-17-C-0087. 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. Figure 1 is licensed by the authors under the Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/>.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
HSCC '19, April 16–18, 2019, Montreal, QC, Canada

© 2019 Copyright held by the owner/author(s). Publication rights licensed to the Association for Computing Machinery.
ACM ISBN 978-1-4503-6282-5/19/04...\$15.00
<https://doi.org/10.1145/3302504.3311809>

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 [17], stochastic motion planning problems [18], automated anesthesia delivery [1], and many others. SReachTools solves the problem of stochastic reachability of a target tube for linear systems [9, 26]. In [26], it is shown that this problem subsumes the terminal-time hitting stochastic reach-avoid problem [22] and the stochastic viability problems [2].

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 [23] to characterize the stochasticity of the state at a future time of interest.

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 [2, 22], 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 [17, 27], particle filters [17], and Fourier transform-based techniques [24]. In [25, 26], 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 [8, 9] provide a set-theoretic approach based on computational geometry to under- and over-approximations of the stochastic reach set, as well as compute a state-feedback admissible controller. 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 [12], FAUST² [21], PRISM [16], and STORM [4] provide high-level language or graphical tools for model-checking of continuous and discrete-time Markov chains. FAUST², handles continuous state and action spaces with a specified error via discretization of the state and action spaces. Barrier

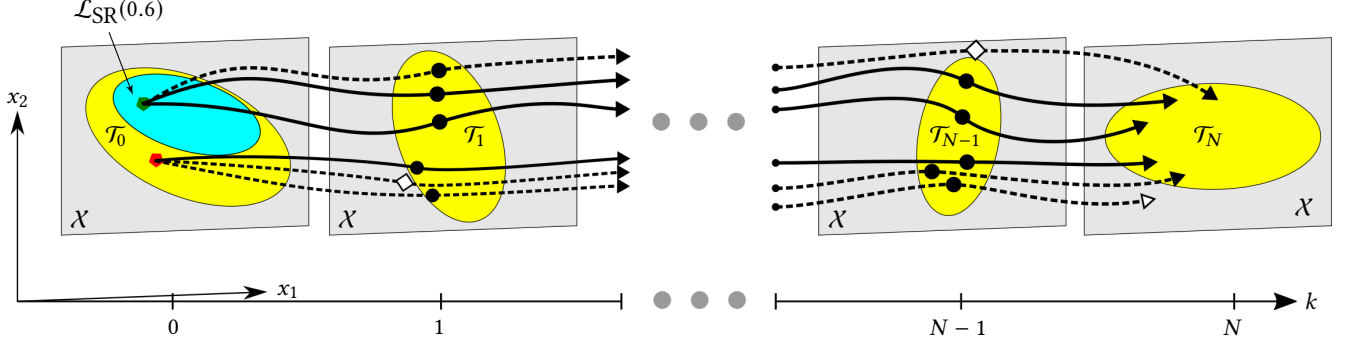


Figure 1: The target tube \mathcal{T}_N , the stochastic evolution of (1) or (2) under a maximal reach policy π^* , and the stochastic reach set $\mathcal{L}_{\text{SR}}(\alpha)$ (equation (6)) for $\alpha = 0.6$. The stochastic reachability of a target tube problem subsumes the terminal hitting-time reach-avoid problem [22] ($\forall k \in \mathbb{N}_{[0, N-1]}, \mathcal{T}_i = S, \mathcal{T}_N = \mathcal{R}$) and the viability problem [2] ($\forall k \in \mathbb{N}_{[0, N]}, \mathcal{T}_k = S$) for safe and terminal sets S and \mathcal{R} . This image is licensed by the authors under the Creative Commons Attribution-ShareAlike 4.0 International License.

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 [1, 14]. The key advantage of SReachTools is that it solves stochastic reachability problems without using gridding, providing significant scalability.

The toolbox is open-source. This allows 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 is conducted; and 3) 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 $\mathbb{N}_{[a, b]} = \{c \in \mathbb{N} : a \leq c \leq b\}$, 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 . For $\mathcal{X} \subseteq \mathbb{R}^n$, $n \in \mathbb{N}, n > 0$, the indicator function corresponding to a set S is $1_S : \mathcal{X} \rightarrow \{0, 1\}$, where $1_S(x) = 1$ if $x \in S$ and is zero otherwise; the Cartesian product of S with itself $k \in \mathbb{N}$ times is S^k .

2.2 System formulation

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

$$x_{k+1} = Ax_k + Bu_k + Fw_k \quad (1)$$

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),

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

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 $\mu : \mathcal{X} \rightarrow \mathcal{U}$ 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 $\rho = (M, d) \in (\mathbb{R}^{m \times n \times N}, \mathbb{R}^{mN})$ as an affine transformation of the concatenated disturbance vector $\bar{w}_{[0, N-1]} \in \mathcal{W}^N$ into a concatenated input space $\bar{u}_{[0, N-1]}$,

$$\bar{u}_{[0, N-1]} = M\bar{w}_{[0, N-1]} + d, \quad (3)$$

in which the affine disturbance feedback gain assures causality. It is well-known that (3) is equivalent to a history-dependent linear state-feedback law [10]. 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 $\bar{x}_{[1, N]}$ has a probability measure $\mathbb{P}_{\bar{x}_{[1, N]}}^\pi$ (or $\mathbb{P}_{\bar{x}_{[1, N]}}^\rho$, respectively), induced from \mathbb{P}_w and dynamics (1) or (2).

As in [9, 26], we define a target tube $\mathcal{T}_N = [\mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_N]$ 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 \mathcal{T} be of length N , we define the function $V_{\text{SR}} : \mathcal{X} \times \mathcal{M} \rightarrow [0, 1]$ as

$$V_{\text{SR}}(x_0, \pi) = 1_{\mathcal{T}_0}(x_0) \mathbb{P}_{\bar{x}_{[1, N]}}^{\pi, x_0} \{x_i \in \mathcal{T}_i, \forall i \in \mathbb{N}_{[1, N]}\}, \quad (4)$$

which can also be written as an expectation of a product of indicator functions [2, 9, 22, 26].

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 (4), i.e.

$$V_{\text{SR}}^*(x) = \max_{\pi \in \Pi} V_{\text{SR}}(x, \pi). \quad (5)$$

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 [2, 22].

In many applications, we are also interested in $\mathcal{L}_{\text{SR}}(\alpha)$, 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,

$$\begin{aligned} \mathcal{L}_{\text{SR}}(\alpha) &= \{x \in \mathbb{R}^n \mid \exists \pi \in \Pi, V_{\text{SR}}(x, \pi) \geq \alpha\} \\ &= \{x \in \mathbb{R}^n \mid V_{\text{SR}}^*(x) \geq \alpha\}. \end{aligned} \quad (6)$$

To ensure safety guarantees, we desire either exact representations or assured under-approximations of $V_{\text{SR}}^*(x)$ and $\mathcal{L}_{\text{SR}}(\alpha)$. An over-approximation of $\mathcal{L}_{\text{SR}}(\alpha)$, when available, provides insights into the quality of under-approximation.

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 $\bar{x}_{[0,k]}$ for some $k > 0$ [23]. 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. FAUST², PRISM, and STORM, as well as SReachTools.

3 FEATURES OF SReachTools

The primary goal of SReachTools is to enable efficient 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) [13] 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.

Function	method_str	Utility
SReachPoint	chance-open (G)	Under-approx. $V_{\text{SR},\text{open}}^*(x)$
	genzps-open (G)	Approx. $V_{\text{SR},\text{open}}^*(x)$ within ϵ_{genz}
	particle-open	Approx. $V_{\text{SR},\text{open}}^*(x)$
	voronoi-open	Approx. $V_{\text{SR},\text{open}}^*(x)$ within $\epsilon_{\text{voronoi}}$
	chance-affine (G)	Under-approx. $V_{\text{SR},\text{affine}}^*(x)$
SReachSet	chance-open (G)	Under-approx. $\mathcal{K}_{\text{SR}}(\alpha)$
	genzps-open (G)	Approx. $\mathcal{K}_{\text{SR}}(\alpha)$ within ϵ_{genz}
	lag-under	Under-approx. $\mathcal{L}_{\text{SR}}(\alpha)$
	lag-open	Over-approx. $\mathcal{L}_{\text{SR}}(\alpha)$
SReachFwd	state-stoch	Stochasticity of x_k
	concat-stoch	Stochasticity of $\bar{x}_{[0,k]}$
	state-prob	$\mathbb{P}_{x_k}^{x_0} \{x_k \in \mathcal{T}\}$
	concat-prob	$\mathbb{P}_{\bar{x}_{[1,k]}}^{x_0} \{\bar{x}_{[1,k]} \in \times_{i=1}^k \mathcal{T}_i\}$
SReachDyn	–	Dyn. prog.-based approx. of $V_{\text{SR}}^*(x)$ and $\mathcal{L}_{\text{SR}}(\alpha)$

Table 1: Current features of SReachTools. By (9), (11), and (13), we can characterize $V_{\text{SR}}^*(x)$ and $\mathcal{L}_{\text{SR}}(\alpha)$ using $V_{\text{SR},\text{open}}^*(x)$, $V_{\text{SR},\text{affine}}^*(x)$, and $\mathcal{K}_{\text{SR}}(\alpha)$. Methods that require Gaussian disturbance are marked with (G).

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 [13] to define a collection of time-stamped polytopic safe sets. One can also easily specify a reach-avoid [22] or viability [2] specification using `Tube`. Currently, `RandomVector` supports Gaussian disturbances, and user-defined disturbances defined by a random sample generator function.

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 \mathcal{T}_N . For all the functions, `prob_str` refers to the type of reachability problem. For `SReachPoint`, `SReachSet`, and `SReachDyn`, `prob_str` must always be ‘term’ as of SReachTools v1.2 (the current version). This refers to 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 [22].

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 [2, 22].

The recursion is given by

$$V_N^*(x) = \mathbf{1}_{\mathcal{T}_N}(x) \quad (7a)$$

$$V_k^*(x) = \sup_{u \in \mathcal{U}} \left(\mathbf{1}_{\mathcal{T}_k}(x) \int_{\mathcal{X}} V_{k+1}^*(y) Q_k(dy|x, u) \right), \quad (7b)$$

where Q_k is the stochastic transition kernel and $V_{\text{SR}}^*(x) = V_0^*(x)$. We can use SReachTools to implement a uniform grid-based implementation of (7) using SReachDynProg.

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

Here, x_inc and u_inc are grid step sizes for the state space \mathcal{X} and input space \mathcal{U} respectively. SReachTools's implementation of the dynamic programming currently handles only LtiSystem and Gaussian disturbances, and is utilized for benchmarking other techniques. In future versions, we plan to link SReachTools with existing tools for approximate dynamic programming like FAUST².

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

SReachTools can be used to approximate $V_{\text{SR}}^*(x)$. 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 $V_{\text{SR,open}}^* : \mathcal{X} \rightarrow [0, 1]$, with

$$V_{\text{SR,open}}^*(x_0) = \max_{d \in \mathcal{U}^N} \mathbf{1}_{\mathcal{T}_0}(x_0) \mathbb{P}_{\bar{x}_{[1,N]}}^{d, x_0} \{ \forall i \in \mathbb{N}_{[1,N]}, x_i \in \mathcal{T}_i \}. \quad (8)$$

Note that (8) is a log-concave optimization problem whenever the target tube is polytopic and the disturbance is log-concave [26, Thm. 6]. Moreover, (8) under-approximates (5) [26, Thm. 7],

$$V_{\text{SR}}^*(x) \geq V_{\text{SR,open}}^*(x), \quad \forall x \in \mathcal{X}. \quad (9)$$

With an affine controller (M, d) , we enforce a probabilistic constraint $\mathbb{P}_{\bar{u}_{[0,N-1]}}^{M,d} \{ \bar{u}_{[0,N-1]} \in \mathcal{U}^N \} = \mathbb{P}_{\bar{w}_{[0,N-1]}} \{ M\bar{w}_{[0,N-1]} + d \in \mathcal{U}^N \} \geq 1 - \Delta_U$ 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 (8), we define $V_{\text{SR,affine}}^*(x)$ as the optimal value function of (10),

$$\begin{aligned} \max_{M,d} \quad & \mathbb{P}_{\bar{x}_{[1,N]}}^{M,d,x_0} \{ \forall i \in \mathbb{N}_{[1,N]}, x_i \in \mathcal{T}_i \} \\ \text{s.t.} \quad & \begin{cases} M, d \text{ as in (3)}, & x_0 = x \in \mathcal{T}_0 \\ \mathbb{P}_{\bar{w}_{[0,N-1]}} \{ M\bar{w}_{[0,N-1]} + d \in \mathcal{U}^N \} \geq 1 - \Delta_U \end{cases} \end{aligned} \quad (10)$$

Given Δ_U , we have [27, Thm. 1]

$$V_{\text{SR}}^*(x) \geq 1 - \frac{1 - V_{\text{SR,affine}}^*(x)}{1 - \Delta_U}, \quad \forall x \in \mathcal{X}. \quad (11)$$

SReachTools can synthesize an open-loop controller by solving (8) using the following four approaches.

- (M1) *Convex chance constraints*: An under-approximative reformulation, which is a linear program, via Boole's inequality and Gaussian vector properties [17, 27].
- (M2) *Fourier transform (Genz's algorithm & patternsearch)*: An approximative reformulation via the characteristic function (Fourier transform of $\psi_{\bar{x}_{[0,N]}}$), integrated over the target

tube [24]. The high-dimensional numerical quadrature up to a user-specified accuracy of ϵ_{genz} is done using Genz's algorithm [6, 7], and the optimization is done using MATLAB's patternsearch [15].

- (M3) *Particle filter*: Mixed integer-linear reformulation via scenarios drawn from $\mathbb{P}_{\bar{x}_{[1,N]}}^{d, x_0}$ [17].
- (M4) *Voronoi-approaches*: A voronoi partition-based under-approximation of the particle filter approach [19]

Methods (M1) and (M2) require the additive disturbance to be Gaussian, while (M3) and (M4) do not impose this restriction. Also for Gaussian disturbances, SReachTools can also synthesize an affine controller (3) by solving (10) using chance constraint reformulation and difference of convex programming [27]. SReachTools uses CVX [11] to implement all of the convex and mixed-integer 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', 'genzps-open', 'particle-open', 'voronoi-open', or 'chance-affine'), `init_state` is the initial state $x_0 \in \mathcal{X}$ at which an approximation of $V_{\text{SR}}^*(x_0)$ must be evaluated, and `options` enables specification of accuracy and method-specific arguments.

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 $\mathcal{L}_{\text{SR}}(\alpha)$ for $\alpha \in [0, 1]$ [26, Thm. 4]. This is also true for the superlevel sets of $V_{\text{SR,open}}^*(x)$ [26, Thm. 6],

$$\mathcal{K}_{\text{SR}}(\alpha) = \{x \in \mathbb{R}^n \mid V_{\text{SR,open}}^*(x) \geq \alpha\}. \quad (12)$$

Since $V_{\text{SR}}^*(x) \geq V_{\text{SR,open}}^*(x)$ for every $x \in \mathcal{X}$,

$$\mathcal{K}_{\text{SR}}(\alpha) \subseteq \mathcal{L}_{\text{SR}}(\alpha), \quad \forall \alpha \in [0, 1]. \quad (13)$$

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 $\mathcal{K}_{\text{SR}}(\alpha)$ [25, 26]. This approach requires the additive disturbance to be Gaussian, and it provides an optimal open-loop controller for all of the vertices of the under-approximation.

3.4.2 Lagrangian approach. This technique computes grid-free approximations to the stochastic reach set $\mathcal{L}_{\text{SR}}(\alpha)$. Specifically, it computes disturbance minimal and disturbance maximal reach sets that serve as under- and over-approximations of $\mathcal{L}_{\text{SR}}(\alpha)$ [8, 9]. These computations are based on bounded disturbance sets, computed using the stochasticity of w_k . We model the sets via polytopes or ellipsoids. SReachEllipsoid, an auxillary class of SReachTools, allows for easy manipulation of ellipsoids.

SReachTools implements backward recursions that employs Minkowski sum, difference, intersection, and affine transformations, which affects scalability of the approach due to the computationally hard *vertex-facet enumeration problem*. SReachTools implements a vertex-complexity preserving recursion for under-approximation using support vectors. Further, using robust linear programming and support functions, SReachTools also implements a recursion-free method to compute the over-approximation. Finally, we can perform state-feedback controller synthesis using the under-approximation sets, by solving a collection of linear programs. SReachTools performs vertex-facet enumeration via LRS [3, 20] or CDD [5].

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

SReachFwd characterizes the stochasticity of the state at a future time k x_k and the concatenated state vector $\bar{x}_{[1,k]}$.

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

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 \mathcal{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. In order to compare methods, we restrict ourselves to Gaussian disturbances in the examples.

4.1 Chain of integrators

This example demonstrates accuracy and scalability. A chain of integrators [26, 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 $m \leq 5$ 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.

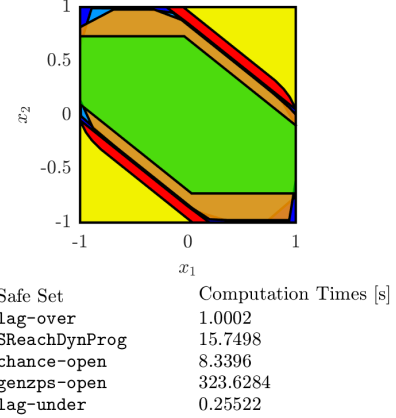


Figure 2: Comparison of $\mathcal{L}_{SR}(0.8)$ for a 2-dimensional double integrator computed via dynamic programming, and multiple SReachSet approximations. Tradeoffs between accuracy and computation time are problem dependent.

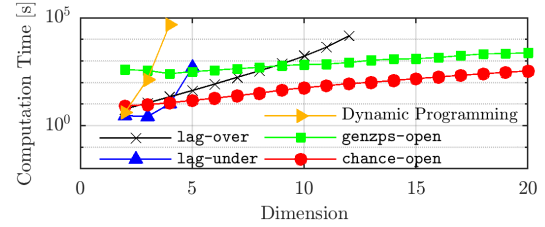


Figure 3: Scalability of SReachSet methods demonstrated on a chain of integrators, as state dimension increases.

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 [26, Eq. (26)], which simplifies the original nonlinear dynamics to an LTV system. We use a time-varying target tube with target sets $\mathcal{T}_k = \text{Box}\left(c_k, 0.5 \exp\left(\frac{-k}{100}\right)\right)$, 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 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 to a satellite rendezvous and docking problem. Probabilistic guarantees of safety are crucial in spacecraft systems because of the high cost of failure. We use Clohessy-Wiltshire-Hill near-orbit spacecraft

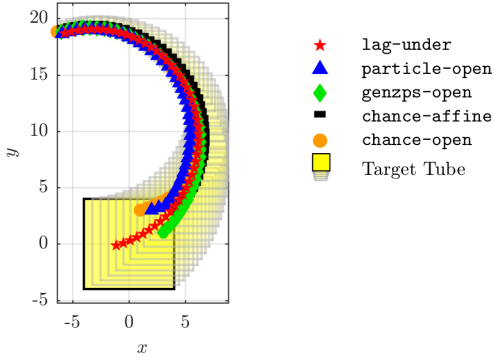


Figure 4: Mean trajectories of the Dubins' vehicle corresponding to the optimal controllers synthesized via $SReachPoint$. All trajectories report $V_{SR}^*(\cdot) \geq 0.9$.

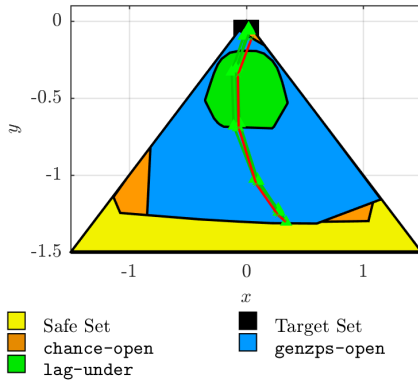


Figure 5: Satellite rendezvous problem with approximations of $\mathcal{L}_{SR}(0.8)$ computed using $SReachSet$. We use Monte-Carlo simulations to validate the controller (chance-open) for the initial state $x = 0.21$, $y = -1.30$, and zero velocity; good and bad trajectories are marked with 'Δ' and 'x' respectively.

dynamics [17, Eqs. (1)–(2)] to describe an LTI system with a 4-dimensional state vector. Figure 5 shows the approximations of $\mathcal{L}_{SR}(0.8)$ computed with various methods in $SReachSet$. We validate the optimal open-loop controller obtained with chance-open using Monte-Carlo simulations.

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] A. Abate, H. Blom, N. Cauchi, S. Haesaert, A. Hartmanns, K. Lesser, M. Oishi, V. Sivaramakrishnan, S. Soudjani, C. Vasile, and A. Vinod. 2018. ARCH-COMP18 Category Report: Stochastic Modelling. In *ARCH18. 5th International Workshop on Applied Verification of Continuous and Hybrid Systems*. Oxford, UK, 71–103.
- [2] A. Abate, M. Prandini, J. Lygeros, and S. Sastry. 2008. Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems. *Automatica* 44 (October 2008), 2724–2734.
- [3] D. Avis. 1997. LRS. <http://cgm.cs.mcgill.ca/~avis/C/lrs.html>. (1997).
- [4] C. Dehner, S. Junges, J. Katoen, and M. Volk. 2017. A storm is coming: A modern probabilistic model checker. In *International Conference on Computer Aided Verification*. Heidelberg, Germany, 592–600.
- [5] K. Fukuda. 2002. CDD. http://www.inf.ethz.ch/personal/fukudak/cdd_home/index.html. (2002).
- [6] A. Genz. 1992. Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics* 1, 2 (1992), 141–149.
- [7] A. Genz. 2014. Quadrature of a multivariate normal distribution over a region specified by linear inequalities: QSCMVNV. (2014). <http://www.math.wsu.edu/faculty/genz/software/matlab/qscmvnv.m>
- [8] J. Gleason, A. Vinod, and M. Oishi. 2017. Underapproximation of Reach-Avoid Sets for Discrete-Time Stochastic Systems via Lagrangian Methods. In *Proc. of the Conference on Decision and Control*. Melbourne, Australia, 4283–4290.
- [9] J. Gleason, A. Vinod, and M. Oishi. 2018. Lagrangian Approximations for Stochastic Reachability of a Target Tube. online. (2018). <https://arxiv.org/abs/1810.07118>
- [10] P. Goulart, E. Kerrigan, and J. Maciejowski. 2006. Optimization over state feedback policies for robust control with constraints. *Automatica* 42 (2006), 523–533.
- [11] M. Grant and S. Boyd. 2017. CVX: MATLAB Software for Disciplined Convex Programming. (2017). <http://cvxr.com/cvx>
- [12] A. Hartmanns and H. Hermanns. 2014. The Modest Toolset: An integrated environment for quantitative modelling and verification. In *Proc. of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Grenoble, France, 593–598.
- [13] M. Herceg, M. Kvasnica, C.N. Jones, and M. Morari. 2013. Multi-Parametric Toolbox 3.0. In *Proc. of the European Control Conference*. Zürich, Switzerland, 502–510. <http://people.ee.ethz.ch/~%7Empt/3/>.
- [14] P. Jagtap, S. Soudjani, and M. Zamani. 2018. Temporal Logic Verification of Stochastic Systems Using Barrier Certificates. In *Automated Technology for Verification and Analysis*. Los Angeles, CA, USA, 177–193.
- [15] T. Kolda, R. Lewis, and V. Torczon. 2003. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM review* 45, 3 (2003), 385–482.
- [16] M. Kwiatkowska, G. Norman, and D. Parker. 2011. PRISM 4.0: Verification of Probabilistic Real-time Systems. In *Proc. of the International Conference on Computer Aided Verification*, Vol. 6806. Snowbird, UT, USA, 585–591.
- [17] K. Lesser, M. Oishi, and R. Erwin. 2013. Stochastic Reachability for Control of Spacecraft Relative Motion. In *Proc. of the Conference on Decision and Control*. Florence, Italy, 4705–4712.
- [18] N. Malone, H. Chiang, K. Lesser, M. Oishi, and L. Tapia. 2017. Hybrid Dynamic Moving Obstacle Avoidance Using a Stochastic Reachable Set-Based Potential Field. *IEEE Transactions on Robotics* 33, 5 (2017), 1124–1138.
- [19] H. Sartipzadeh, A. Vinod, Behçet Açikmese, and M. Oishi. 2019. Voronoi Partition-based Scenario Reduction for Fast Sampling-based Stochastic Reachability Computation of LTI Systems. In *Proc. of the American Control Conference*. (accepted).
- [20] R. Schaich. 2018. GeoCalcLib: A Matlab Interface for the LRS Library. <http://worc4021.github.io/>. (2018).
- [21] S. Soudjani, C. Gevaerts, and A. Abate. 2015. FAUST²: Formal Abstractions of Uncountable S-Tate Stochastic processes. In *Proc. of the International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, Vol. 15. London, UK, 272–286.
- [22] S. Summers and J. Lygeros. 2010. Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem. *Automatica* 46 (September 2010), 1951–1961.
- [23] A. Vinod, B. HomChaudhuri, and M. Oishi. 2017. Forward Stochastic Reachability Analysis for Uncontrolled Linear Systems using Fourier Transforms. In *Proc. of the Hybrid Systems: Computation and Control*. Pittsburgh, PA, USA, 35–44.
- [24] A. Vinod and M. Oishi. 2017. Scalable Underapproximation for the Stochastic Reach-Avoid Problem for High-Dimensional LTI Systems Using Fourier Transforms. *IEEE Control Systems Letters (L-CSS)* 1, 2 (Oct 2017), 316–321.
- [25] A. Vinod and M. Oishi. 2018. Scalable underapproximative verification of stochastic LTI Systems using convexity and compactness. In *Proc. of the Hybrid Systems: Computation and Control*. Porto, Portugal, 1 – 10.
- [26] A. Vinod and M. Oishi. 2018. Stochastic reachability of a target tube: Theory and computation. *IEEE Trans. Automat. Control* (2018). <https://arxiv.org/abs/1810.05217v1> (submitted).
- [27] A. Vinod and M. Oishi. 2019. Affine controller synthesis for stochastic reachability via difference of convex programming. In *Proc. of the Conference on Decision and Control*. <https://hscl.unm.edu/affinecontrollersynthesis/> (submitted).