Documentation for "Stochastic Reachability for Systems up to a Million Dimensions"

Adam J. Thorpe, Vignesh Sivaramakrishnan, Meeko M. K. Oishi

October 17, 2019

Documentation for the algorithms presented in "Stochastic Reachability for Systems up to a Million Dimensions" by Adam J. Thorpe, Vignesh Sivaramakrishnan, Meeko M. K. Oishi.

Contents

List of Symbols

$\mathcal{H}$	Markov Control Process 2
$\mathcal{X} \subseteq \Re^n$	State Space 2
$\mathcal{U} \subseteq \Re^m$	Control Space 2
Q	Stochastic Kernel 2, 3
$\mathcal{S}$	Sample Set 2

1 Start Here

2 Instructions

2.1 Running The Code

2.2 Generating the Figures

2.3 Modifying the Code

3 Algorithms

The algorithms presented

3.1 Preliminaries

We consider a a Markov control process $\mathcal{H}$, which is defined in [?] as a 3-tuple:

$$\mathcal{H} = (\mathcal{X}, \mathcal{U}, Q) \tag{1}$$

where $\mathcal{X} \subseteq \Re^n$ is the state space, $\mathcal{U} \subseteq \Re^m$ is the control space, and Q is a stochastic kernel $Q : \mathscr{B}(\mathcal{X}) \times \mathcal{X} \times \mathcal{U} \to [0, 1]$, which is a Borel-measurable function that maps a probability measure $Q(\cdot \,|\, x, u)$ to each $x \in \mathcal{X}$ and $u \in \mathcal{U}$ in the Borel space $(\mathcal{X}, \mathscr{B}(\mathcal{X}))$. A Markov control process can describe a wide class of stochastic, time-invariant systems, that can have either linear or nonlinear dynamics, as well as non-Gaussian disturbances. We consider a set $\mathcal{S}$ of M samples of the form $\{(\bar{x}_i, \bar{u}_i, \bar{y}_i)\}_{i=1}^{M}$ taken from the stochastic kernel, such that $\bar{y}_i$ is drawn i.i.d. from the stochastic kernel Q, and $\bar{u}_i$ is drawn from a fixed Markov policy π.

$$\bar{y}_i \sim Q(\,\cdot\,|\,\bar{x}_i, \bar{u}_i) \tag{2}$$

$$\bar{u}_i = \pi(\bar{x}_i) \tag{3}$$

The samples can be generated experimentally or via simulation, meaning they can be taken from real observations of the system evolution, or they can be generated using a known model. For demonstration purposes, all examples use samples collected via simulation. Once the samples are generated, the algorithm assumes no knowledge of the stochastic kernel Q or the disturbance.

3.2 Kernel Distribution Embeddings Backward Recursion Algorithm

3.3 Kernel Distribution Embeddings Backward Recursion (RFF) Algorithm

4 Problems

4.1 Terminal-Hitting Time Problem

4.2 First-Hitting Time Problem

5 Systems

The algorithms accept sample data drawn from a stochastic kernel Q. The data should be formatted such that the realizations of the stochastic kernel are formatted into the columns of a sample vector

$$\bar{x} = [\bar{x}_1, \ldots, \bar{x}_M] \tag{4}$$
$$\bar{u} = [\bar{u}_1, \ldots, \bar{u}_M] \tag{5}$$
$$\bar{y} = [\bar{y}_1, \ldots, \bar{y}_M] \tag{6}$$

where the number of columns is M, and the number of rows is the dimensionality of the samples. For example, if $\bar{x}_i, \bar{y}_i \in \Re^n$, $\bar{x}$ and $\bar{y}$ should be $[n \times M]$.

5.1 System Samples

The system input to the algorithms is a set of samples organized in a class called `SystemSamples`. For example, to generate samples for the discrete-time double integrator system with sampling time $T = 0.25$,

$$\boldsymbol{x}_{k+1} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \boldsymbol{x}_k + \begin{bmatrix} \frac{T^2}{2!} \\ T \end{bmatrix} u_k + \boldsymbol{w}_k \tag{7}$$

```
% Dimensionality of the state space samples.
n = 2;
% Dimensionality of the input space samples.
m = 2;
% Sampling time.
T = 0.25;

% Number of samples.
M = 1000;

% Compute random initial states sampled from a zero-mean Gaussian.
X = randn(n, M);
% Compute the input samples. For this example, the input is chosen to be 0.
U = zeros(m, M);
% Compute the disturbance.
W = randn(n, M);

% Construct the state and input matrices.
A = [1 T; 0 1];
B = [(T^2)/2!; T];

% compute the output samples.
Y = A*X + B*U + W;
```

```
% Create a SystemSamples object.
samples = SystemSamples('X', X, 'U', U, 'Y', Y);
```

References