AMYTISS: Parallelized Automated Controller Synthesis for Large-Scale Stochastic Systems

Abolfazl Lavaei*, Mahmoud Khaled*, Sadegh Soudjani, Majid Zamani
*: Both authors have the same contribution.

Discrete-Time Stochastic Control Systems

$\Sigma = (X, U, W, \varsigma, f)$:

- $X \subseteq \mathbb{R}^n$: State set;
- $U \subseteq \mathbb{R}^m$: Input set;
- $W \subseteq \mathbb{R}^p$: Disturbance set;
- ς: A sequence of i.i.d. random variables from a sample space Ω to a measurable set V_ς

$$\varsigma := \{\varsigma(k) : \Omega \to V_\varsigma,\ k \in \mathbb{N}\};$$

- $f : X \times U \times W \to X$: Transition map.

Evolution of the state of Σ:

$$\Sigma : x(k+1) = f(x(k), \nu(k), w(k)) + \Upsilon(k), \qquad k \in \mathbb{N},$$

- $\Upsilon(k) := \varsigma(k)$ with $V_\varsigma = \mathbb{R}^n$ for the case of additive noise;
- $\Upsilon(k) := \varsigma(k)x(k)$ with V_ς equal to the set of diagonal matrices of dimension n for the case of multiplicative noise.

Main Contributions

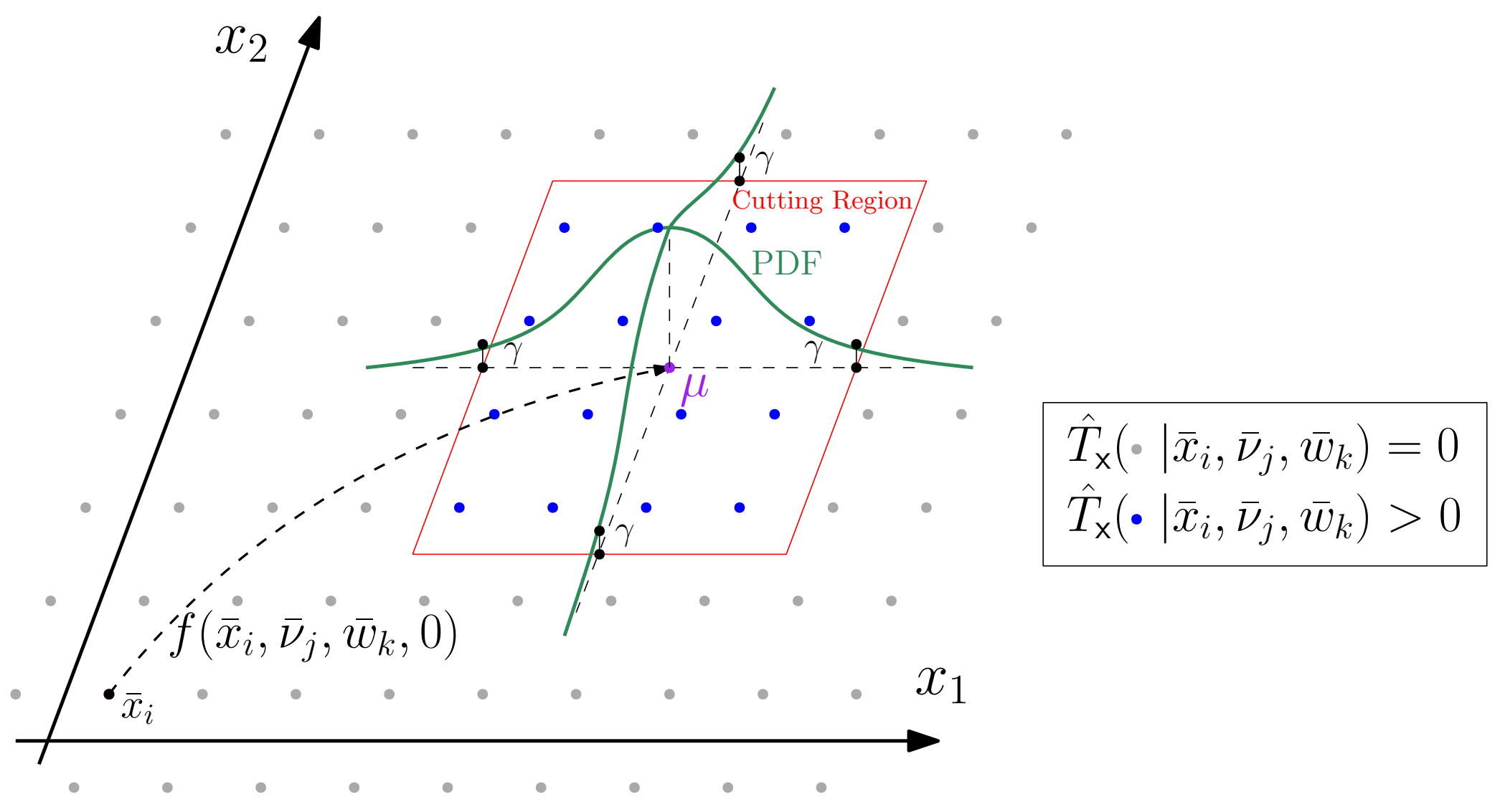
- Implemented on top of `pFaces`: native support for high-performance computing platforms and all operating systems.
- Propose novel data-parallel algorithms for constructing finite MDPs from discrete-time stochastic systems;
- Propose parallel algorithms for synthesizing discrete controllers using the constructed MDPs to satisfy safety, reachability, or reach-avoid specifications;
- Accept bounded disturbances and natively support both additive and multiplicative noises with different practical distributions;
- Apply the proposed implementations to real-world applications including room temperature and road traffic networks, autonomous vehicles, etc.

Comparison between AMYTISS, FAUST^2 and StocHy

Aspect	**FAUST^2**	**StocHy**	**AMYTISS**
Platform	CPU	CPU	CPU, GPU, HW-Accelerators
Algorithms	Serial	Serial	Parallel
Model	SCS: linear, bilinear	SHS: linear, bilinear	SCS: nonlinear
Specification	Safety, reachability	Safety, reachability	Safety, reachability, reach-avoid
Stochasticity	Additive noise	Additive noise	Additive & multiplicative noises
Distribution	Normal, user-defined	Normal, user-defined	Normal, uniform, exponential, beta, user-defined
Disturbance	Not supported	Not supported	Supported

Parallel Construction of Finite MDPs

Massively parallel threads do the computation for each $(\bar{x}, \bar{\nu})$.



Less Memory for Constructing Probability Transition Matrix $\hat{T}_{\mathsf{x}}$

- Reduce the memory usage by setting a cutting probability threshold $\gamma \in [0, 1]$;
- Control the sparsity of the columns of $\hat{T}_{\mathsf{x}}$.

Parallel Synthesis of Controllers

On-the-Fly Construction of $\hat{T}_{\mathsf{x}}$

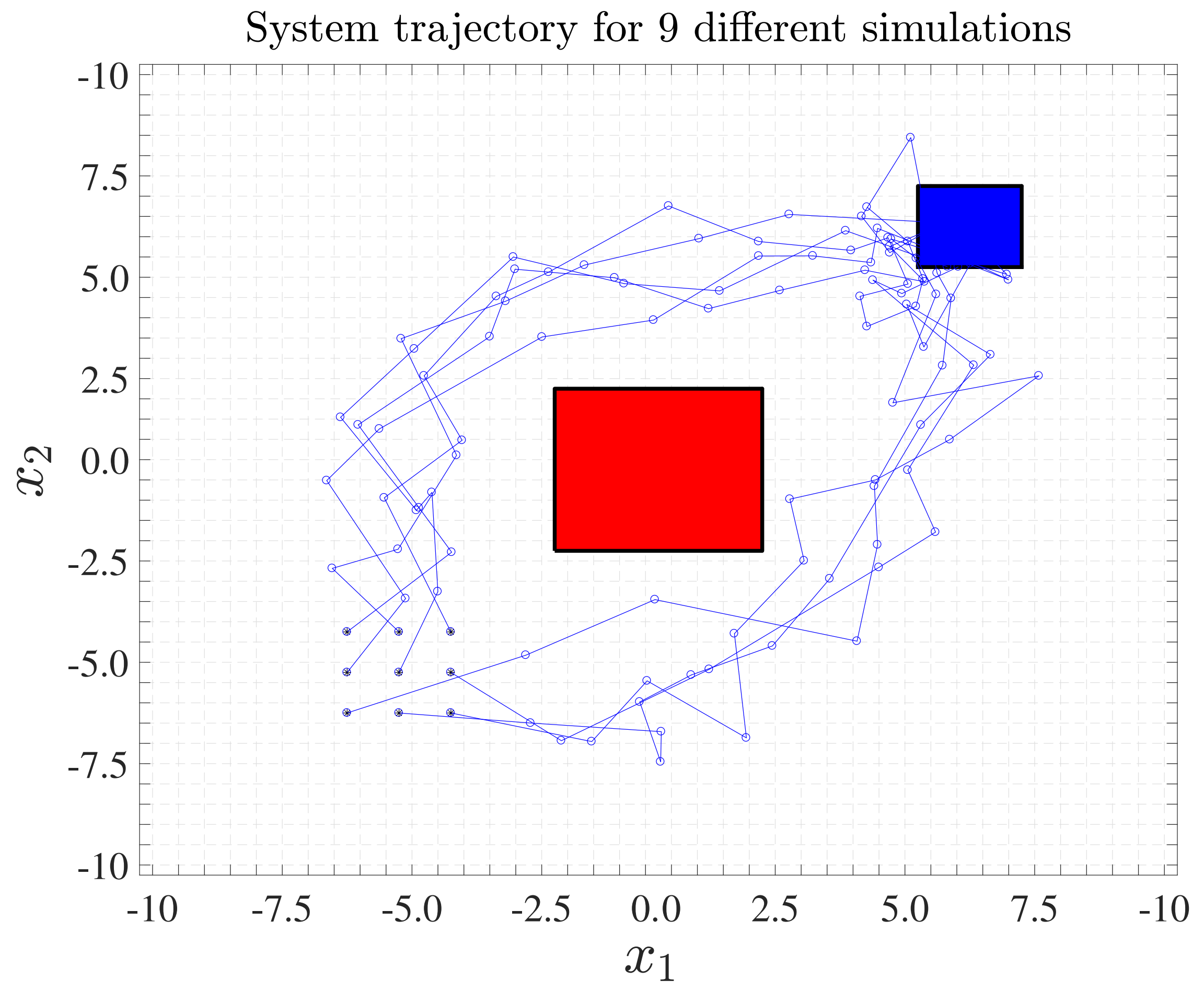
- Skip computing and storing the probability transition matrix $\hat{T}_{\mathsf{x}}$ and only compute the required entries as they are needed for the synthesis part;
- Reduce the required memory for $\hat{T}_{\mathsf{x}}$ at the cost of repeated computation of their entries;
- Give the user an additional control over the trade-off between the computation time and memory usage.

AMYTISS by Example: Nonlinear 2D Robot

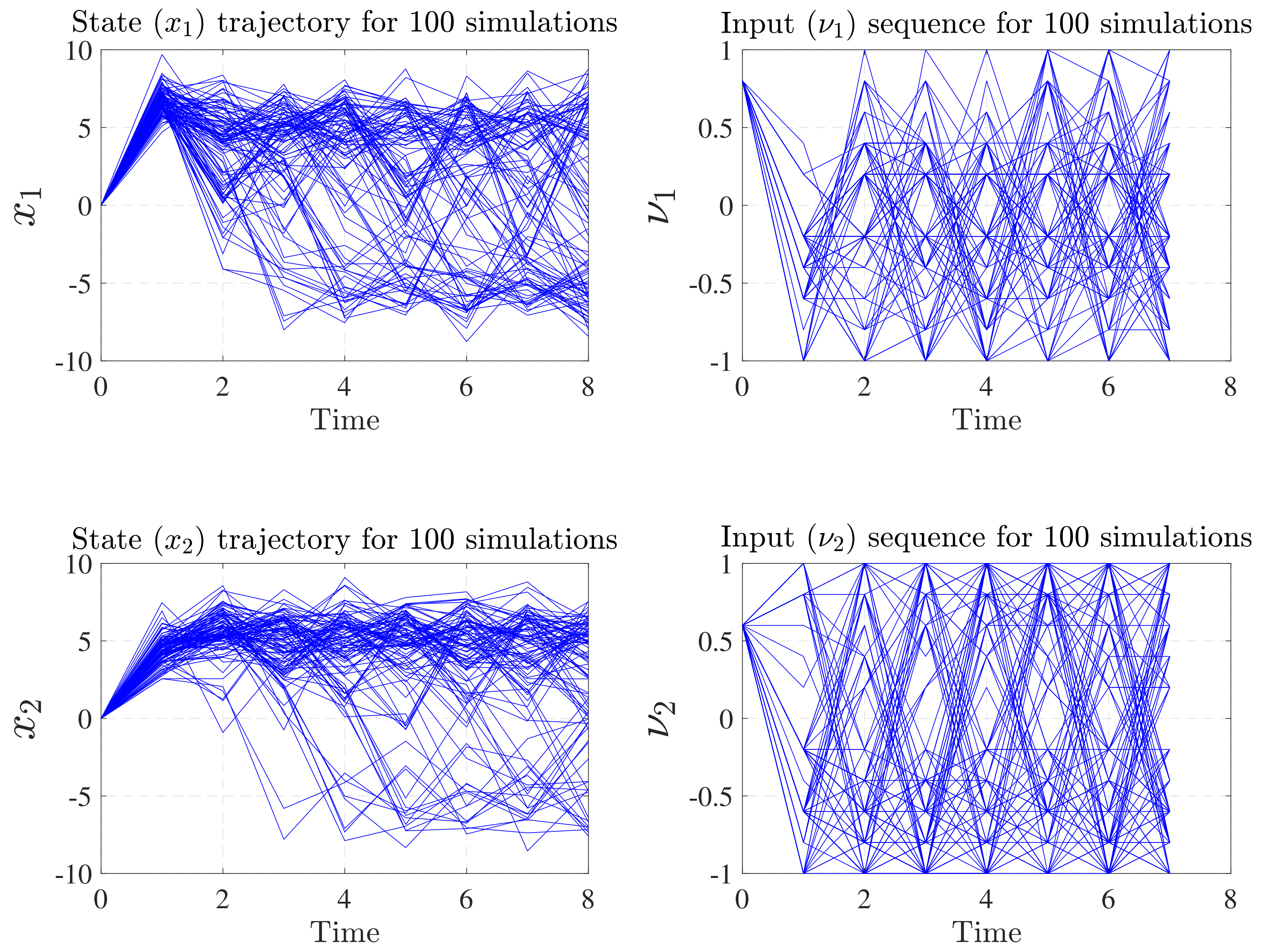
Consider a 2D robot described by the following difference equation:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} x_1(k) + \tau\nu_1(k)\cos(\nu_2(k)) + w(k) + \varsigma_1(k) \\ x_2(k) + \tau\nu_2(k)\sin(\nu_2(k)) + w(k) + \varsigma_2(k) \end{bmatrix}$$

Reach-Avoid Controller:
Target-set $= [5.5, 6.5] \times [5.5, 6.5]$, Avoid-set $= [-2.5, 2.5] \times [-2.5, 2.5]$



Safety Controller:
Safe-set $= [-10, 10] \times [-10, 10]$



Benchmarking

Table: Used HW configurations for benchmarking AMYTISS.

Id	**Description**	**PEs**	**Frequency**
CPU$_1$	Local machine: Intel Xeon E5-1620	8	3.6 GHz
CPU$_2$	MacBook Pro 15: Intel i9-8950HK	12	2.9 GHz
CPU$_3$	Amazon AWS instance `c5.18xlarge`: Intel Xeon P 8000	72	3.6 GHz
GPU$_1$	Macbook Pro 15 laptop laptop: Intel UHD Graphics 630	23	0.35 GHz
GPU$_2$	Macbook Pro 15 laptop: AMD Radeon Pro Vega 20	1280	1.2 GHz
GPU$_3$	Amazon AWS p3.2xlarge instance: NVIDIA Tesla V100	5120	0.8 GHz

Table: Benchmarking results.

			AMYTISS (time)						**FAUST^2**	**StocHy**	**Speedup w.r.t**	
Problem	**Spec.**	$\hat{X} \times \hat{U}$	CPU$_1$	CPU$_2$	CPU$_3$	GPU$_1$	GPU$_2$	GPU$_3$	**(time)**	**(time)**	FAUST	StocHy
4-d StocHy CSB	Safety	16	≤ 1.0	≤ 1.0	≤ 1.0	≤ 1.0	≤ 1.0	0.0002	0.01	0.17	**50 x**	**850 Kx**
5-d StocHy CSB	Safety	32	≤ 1.0	≤ 1.0	≤ 1.0	≤ 1.0	≤ 1.0	0.0003	0.01	0.54	**33 x**	**1.8 Kx**
6-d StocHy CSB	Safety	64	≤ 1.0	≤ 1.0	≤ 1.0	≤ 1.0	≤ 1.0	0.0006	1.2	2.17	**2.0 Kx**	**3.6 Kx**
7-d StocHy CSB	Safety	128	≤ 1.0	≤ 1.0	≤ 1.0	≤ 1.0	≤ 1.0	0.0012	13	9.57	**10.8 Kx**	**7.9 Kx**
8-d StocHy CSB	Safety	256	≤ 1.0	≤ 1.0	≤ 1.0	≤ 1.0	≤ 1.0	0.0026	104	40.5	**40 Kx**	**15.6 Kx**
9-d StocHy CSB	Safety	512	≤ 1.0	≤ 1.0	≤ 1.0	≤ 1.0	≤ 1.0	0.0057	1126	171.6	**197 Kx**	**30.1 Kx**
10-d StocHy CSB	Safety	1024	≤ 1.0	≤ 1.0	≤ 1.0	≤ 1.0	≤ 1.0	0.0122	N/M	385.5	N/A	**32 Kx**
11-d StocHy CSB	Safety	2048	1.0912	≤ 1.0	≤ 1.0	≤ 1.0	≤ 1.0	0.0284	N/M	171.2	N/A	**60 Kx**
12-d StocHy CSB	Safety	4096	4.3029	4.1969	≤ 1.0	≤ 1.0	≤ 1.0	0.0624	N/M	11216	N/A	**179 Kx**
13-d StocHy CSB	Safety	8192	18.681	19.374	1.8515	1.6802	≤ 1.0	0.1277	N/M	$\geq$ 24h	N/A	**$\geq$676 Kx**
14-d StocHy CSB	Safety	16384	81.647	94.750	7.9987	7.3489	6.1632	0.2739	N/M	$\geq$ 24h	N/A	**$\geq$320 Kx**
2-d Robot	R.Avoid	741321	48.593	18.554	4.5127	2.5311	3.4353	0.3083	N/S	N/S	N/A	N/A
3-d Room Temp.	Safety	7776	0.1072	0.0915	0.0120	≤ 1.0	≤ 1.0	0.0018	3027	N/M	**1680 Kx**	N/A
5-d Room Temp.	Safety	279936	200.00	107.93	19.376	10.084	N/M	1.8663	6822	N/M	**3790 Kx**	N/A
3-d Vehicle	R.Avoid	1528065	1614.7	2.5h	1.1h	871.89	898.38	271.41	N/S	N/S	N/A	N/A
7-d BMW **320**i	R.Avoid	3937500	10169.4	N/M	$\geq$ 24h	21.5h	N/M	N/M	N/S	N/S	N/A	N/A

N/S: Not supported. **N/M**: No enough memory. **N/A**: Not applicable for comparison.