Abstracting an ODE as a Markov model

David JULIEN
david.julien@univ-nantes.fr

Nantes Université - LS2N (VELO)
B. DELAHAYE, G. CANTIN, G. ARDOUREL

Oct. 26, 2023

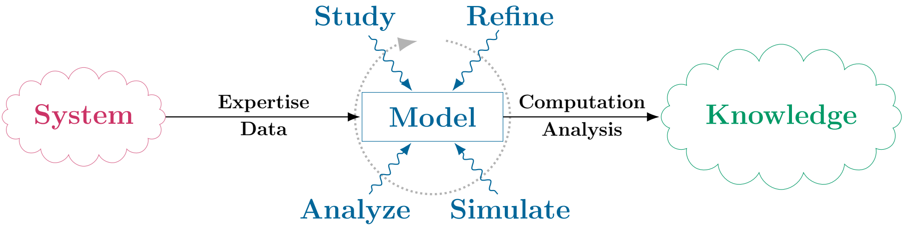
Outline

1. System abstraction
2. 0-player games
3. 1-player games

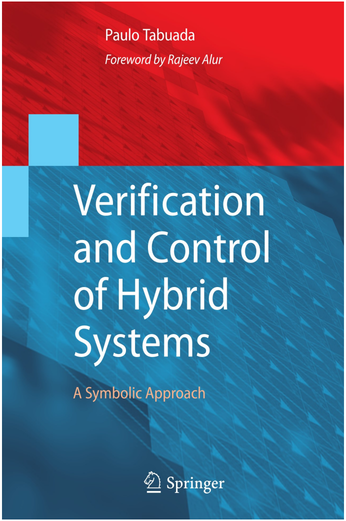
Outline

1. System abstraction
2. 0-player games
3. 1-player games

Reminder


Study
Refine
System
Expertise
Data
Model
Computation
Analysis
Knowledge
Analyze
Simulate

Abstraction of a Dynamical System

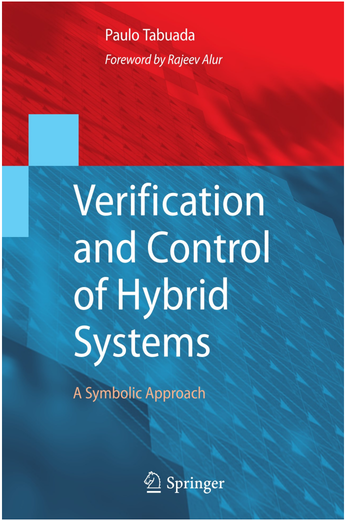


Systems (approximate) simulation

- Prove that two systems are (almost) equivalent;
- Capture the behavior of the concrete system with the abstract one;
- Conclude.

TABUADA, Paulo. Verification and Control of Hybrid Systems. Springer US, 2009.

Abstraction of a Dynamical System



Systems (approximate) simulation

- Prove that two systems are (almost) equivalent;
- Capture the behavior of the concrete system with the abstract one;
- Conclude.

Simulation relation

- states sets X_1 and X_2 respective to systems S_1 and S_2;
- $(x_1, x_2) \in \mathcal{R} \subseteq X_1 \times X_2$;
- for every $x_1' \in Succ(x_1)$, there exists $x_2' \in Succ(x_2)$ such that $x_1' \mathcal{R} x_2'$.

TABUADA, Paulo. Verification and Control of Hybrid Systems. Springer US, 2009.

Example of hybrid system

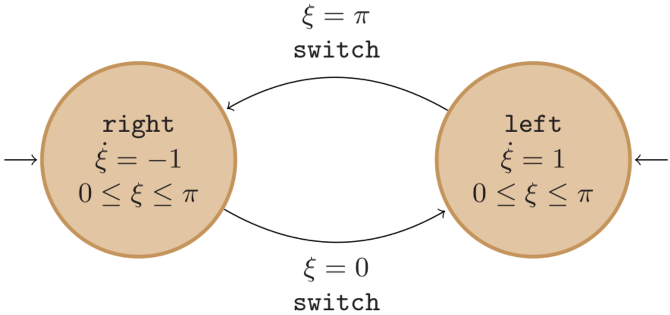
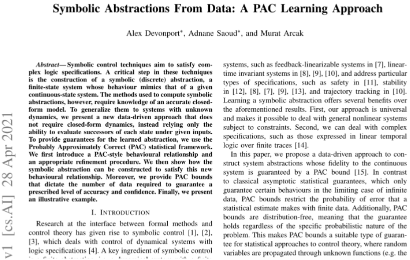


Fig. 7.1. Hybrid dynamical system modeling a windshield wiper.

Building an abstraction of a system



Symbolic Abstractions From Data: A PAC Learning Approach

Alex Devonport*, Adnane Saoud*, and Murat Arcak

***Abstract*— Symbolic control techniques aim to satisfy complex logic specifications. A critical step in these techniques is the construction of a symbolic (discrete) abstraction, a finite-state system whose behaviour mimics that of a given continuous-state system. The methods used to compute symbolic abstractions, however, require knowledge of an accurate closed-form model. To generalize them to systems with unknown dynamics, we present a new data-driven approach that does not require closed-form dynamics, instead relying only the ability to evaluate successors of each state under given inputs. To provide guarantees for the learned abstraction, we use the Probably Approximately Correct (PAC) statistical framework. We first introduce a PAC-style behavioural relationship and an appropriate refinement procedure. We then show how the symbolic abstraction can be constructed to satisfy this new behavioural relationship. Moreover, we provide PAC bounds that dictate the number of data required to guarantee a prescribed level of accuracy and confidence. Finally, we present an illustrative example.**

I. Introduction

Research at the interface between formal methods and control theory has given rise to symbolic control [1], [2], [3], which deals with control of dynamical systems with logic specifications [4]. A key ingredient of symbolic control

systems, such as feedback-linearizable systems in [7], linear-time invariant systems in [8], [9], [10], and address particular types of specifications, such as safety in [11], stability in [12], [8], [7], [9], [13], and trajectory tracking in [10]. Learning a symbolic abstraction offers several benefits over the aforementioned results. First, our approach is universal and makes it possible to deal with general nonlinear systems subject to constraints. Second, we can deal with complex specifications, such as those expressed in linear temporal logic over finite traces [14].

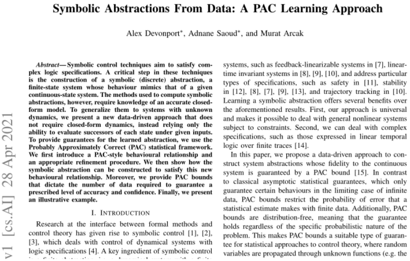
In this paper, we propose a data-driven approach to construct system abstractions whose fidelity to the continuous system is guaranteed by a PAC bound [15]. In contrast to classical asymptotic statistical guarantees, which only guarantee certain behaviours in the limiting case of infinite data, PAC bounds restrict the probability of error that a statistical estimate makes with finite data. Additionally, PAC bounds are distribution-free, meaning that the guarantee holds regardless of the specific probabilistic nature of the problem. This makes PAC bounds a suitable type of guarantee for statistical approaches to control theory, where random variables are propagated through unknown functions (e.g. the

Deterministic blackbox

- transition relation Δ is unknown but deterministic;
- any finite amount of samples.

DEVONPORT, A. et al. Symbolic Abstractions From Data: A PAC Learning Approach. *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021. p. 599-604.

Building an abstraction of a system



Symbolic Abstractions From Data: A PAC Learning Approach

Alex Devonport*, Adnane Saoud*, and Murat Arcak

Abstract— **Symbolic control techniques aim to satisfy complex logic specifications. A critical step in these techniques is the construction of a symbolic (discrete) abstraction, a finite-state system whose behaviour mimics that of a given continuous-state system. The methods used to compute symbolic abstractions, however, require knowledge of an accurate closed-form model. To generalize them to systems with unknown dynamics, we present a new data-driven approach that does not require closed-form dynamics, instead relying only the ability to evaluate successors of each state under given inputs. To provide guarantees for the learned abstraction, we use the Probably Approximately Correct (PAC) statistical framework. We first introduce a PAC-style behavioural relationship and an appropriate refinement procedure. We then show how the symbolic abstraction can be constructed to satisfy this new behavioural relationship. Moreover, we provide PAC bounds that dictate the number of data required to guarantee a prescribed level of accuracy and confidence. Finally, we present an illustrative example.**

I. Introduction

Research at the interface between formal methods and control theory has given rise to symbolic control [1], [2], [3], which deals with control of dynamical systems with logic specifications [4]. A key ingredient of symbolic control

systems, such as feedback-linearizable systems in [7], linear-time invariant systems in [8], [9], [10], and address particular types of specifications, such as safety in [11], stability in [12], [8], [7], [9], [13], and trajectory tracking in [10]. Learning a symbolic abstraction offers several benefits over the aforementioned results. First, our approach is universal and makes it possible to deal with general nonlinear systems subject to constraints. Second, we can deal with complex specifications, such as those expressed in linear temporal logic over finite traces [14].

In this paper, we propose a data-driven approach to construct system abstractions whose fidelity to the continuous system is guaranteed by a PAC bound [15]. In contrast to classical asymptotic statistical guarantees, which only guarantee certain behaviours in the limiting case of infinite data, PAC bounds restrict the probability of error that a statistical estimate makes with finite data. Additionally, PAC bounds are distribution-free, meaning that the guarantee holds regardless of the specific probabilistic nature of the problem. This makes PAC bounds a suitable type of guarantee for statistical approaches to control theory, where random variables are propagated through unknown functions (e.g. the

Deterministic blackbox

- transition relation Δ is unknown but deterministic;
- any finite amount of samples.

Alternating simulation

Prove that the systems capture each other's transitions with a given precision.

DEVONPORT, A. et al. Symbolic Abstractions From Data: A PAC Learning Approach. *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021. p. 599-604.

Outline

1 System abstraction

2 0-player games

3 1-player games

ODE as a Dynamical System

Definitions

- ODE: $\frac{\mathrm{d}}{\mathrm{d}t}\xi(t) = f(\xi(t))$;
- dynamical system: pair $(\mathbb{R}^n, f)$ where f is *regular* and defines an ODE;
- trajectory: map $\xi :]t_0, t_1[\to \mathbb{R}^n$ such that ξ is a solution to the ODE.

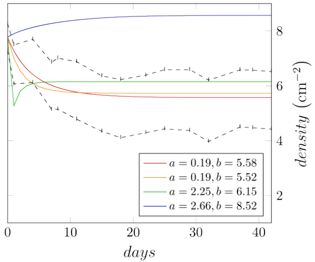
ODE as a Dynamical System

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = ax(t)(1 - \frac{b}{x(t)})$$

$$\rightarrow f(\cdot) = a[\cdot](1 - \frac{b}{[\cdot]}), \quad a, b \in \mathbb{R}$$

Definitions

- ODE: $\frac{\mathrm{d}}{\mathrm{d}t}\xi(t) = f(\xi(t))$;
- dynamical system: pair $(\mathbb{R}^n, f)$ where f is *regular* and defines an ODE;
- trajectory: map $\xi :]t_0, t_1[\rightarrow \mathbb{R}^n$ such that ξ is a solution to the ODE.



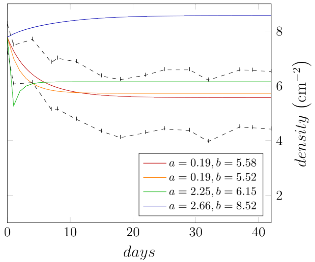
ODE as a Dynamical System

$$\frac{\mathrm{d}}{\mathrm{d}t}x(t) = ax(t)(1 - \frac{b}{x(t)})$$

$$\to f(\cdot) = a[\cdot](1 - \frac{b}{[\cdot]}), \quad a, b \in \mathbb{R}$$

Definitions

- ODE: $\frac{\mathrm{d}}{\mathrm{d}t}\xi(t) = f(\xi(t))$;
- dynamical system: pair $(\mathbb{R}^n, f)$ where f is *regular* and defines an ODE;
- trajectory: map $\xi :]t_0, t_1[\to \mathbb{R}^n$ such that ξ is a solution to the ODE.



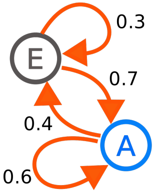
"Predictable" systems

Simple definition (one setting) of isolated systems.
$\to$ Outcome completely determined at launch.

Probabilistic models

Markov chain

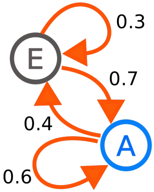
- oriented graph $(E, \mathbb{P})$ where $\mathbb{P} = (p_{i,j})_{(i,j) \in E^2}$ is a transition matrix such that $\sum_j p_{i,j} = 1$;
- transitions are solely determined by the current state of the system;



Probabilistic models

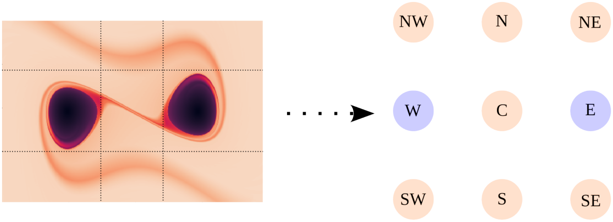
Markov chain

- oriented graph $(E, \mathbb{P})$ where $\mathbb{P} = (p_{i,j})_{(i,j)\in E^2}$ is a transition matrix such that $\sum_j p_{i,j} = 1$;
- transitions are solely determined by the current state of the system;



Useful to describe random, autonomous systems (e.g., nuclear fission or pseudo-random number generator)

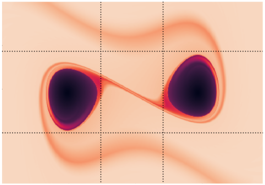
Building a MC with an ODE

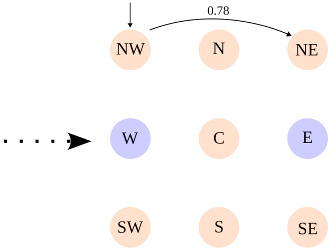


Statistical Model Checking

- $\varphi_{i,j}$ = "the simulation started in region i and stopped in region j";
- one process per pair (i, j);
- caution: need to find the right partition.

Building a MC with an ODE

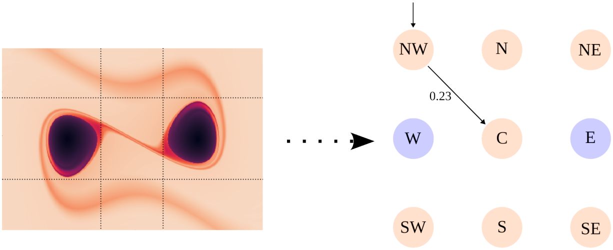




Statistical Model Checking

- $\varphi_{i,j}$ = "the simulation started in region i and stopped in region j";
- one process per pair (i, j);
- caution: need to find the right partition.

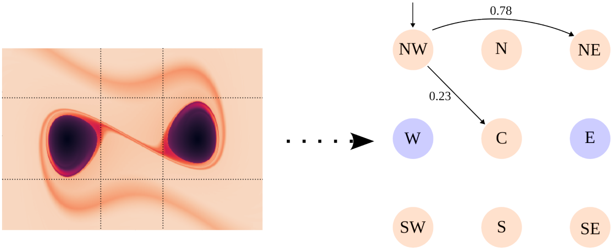
Building a MC with an ODE



Statistical Model Checking

- $\varphi_{i,j}$ = "the simulation started in region i and stopped in region j";
- one process per pair (i, j);
- caution: need to find the right partition.

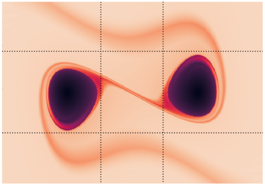
Building a MC with an ODE

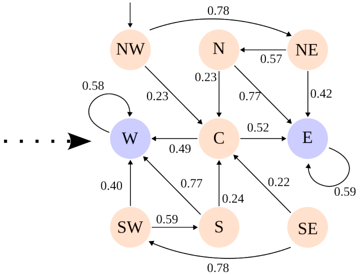


Statistical Model Checking

- $\varphi_{i,j}$ = "the simulation started in region i and stopped in region j";
- one process per pair (i, j);
- caution: need to find the right partition.

Building a MC with an ODE

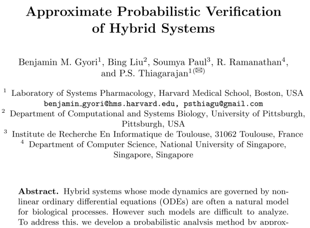




Statistical Model Checking

- $\varphi_{i,j}$ = "the simulation started in region i and stopped in region j";
- one process per pair (i, j);
- caution: need to find the right partition.

Abstracting parametric ODEs with a MC



Approximate Probabilistic Verification of Hybrid Systems

Benjamin M. Gyori[1], Bing Liu[2], Soumya Paul[3], R. Ramanathan[4], and P.S. Thiagarajan[1(✉)]

[1] Laboratory of Systems Pharmacology, Harvard Medical School, Boston, USA
benjamin_gyori@hms.harvard.edu, psthiagu@gmail.com
[2] Department of Computational and Systems Biology, University of Pittsburgh, Pittsburgh, USA
[3] Institute de Recherche En Informatique de Toulouse, 31062 Toulouse, France
[4] Department of Computer Science, National University of Singapore, Singapore, Singapore

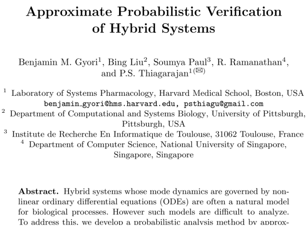
Abstract. Hybrid systems whose mode dynamics are governed by non-linear ordinary differential equations (ODEs) are often a natural model for biological processes. However such models are difficult to analyze. To address this, we develop a probabilistic analysis method by approx-

Problem definition

- hybrid system H (with modes);
- discrete time-step τ;
- trajectory $\pi = q_0 q_1 \ldots q_N$;
- MC $M = \{(\pi, X, \mathbb{P}_X)\}$ where π is a finite trajectory.

GYORI, B. et al. Approximate probabilistic verification of hybrid systems. *International Workshop on Hybrid Systems Biology*, 2015. p. 96–116.

Abstracting parametric ODEs with a MC


Approximate Probabilistic Verification of Hybrid Systems

Benjamin M. Gyori[1], Bing Liu[2], Soumya Paul[3], R. Ramanathan[4], and P.S. Thiagarajan[1(✉)]

[1] Laboratory of Systems Pharmacology, Harvard Medical School, Boston, USA
benjamin_gyori@hms.harvard.edu, psthiagu@gmail.com
[2] Department of Computational and Systems Biology, University of Pittsburgh, Pittsburgh, USA
[3] Institute de Recherche En Informatique de Toulouse, 31062 Toulouse, France
[4] Department of Computer Science, National University of Singapore, Singapore, Singapore

Abstract. Hybrid systems whose mode dynamics are governed by non-linear ordinary differential equations (ODEs) are often a natural model for biological processes. However such models are difficult to analyze. To address this, we develop a probabilistic analysis method by approx-

Problem definition

- hybrid system H (with modes);
- discrete time-step τ;
- trajectory $\pi = q_0 q_1 \dots q_N$;
- MC $M = \{(\pi, X, \mathbb{P}_X)\}$ where π is a finite trajectory.

SMC procedure for MC construction

H satisfies $\phi \Leftrightarrow M$ satisfies ϕ.

GYORI, B. et al. Approximate probabilistic verification of hybrid systems. *International Workshop on Hybrid Systems Biology*, 2015. p. 96–116.

Outline

1 System abstraction

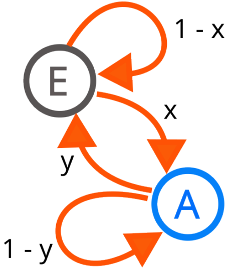
2 0-player games

3 1-player games

Control systems

Markov Decision Process

- enhanced MC $(E, Act, \{\mathbb{P}_a\})$;
- choose an action $a \Leftrightarrow$ choose a probability distribution $\mathbb{P}_a$.

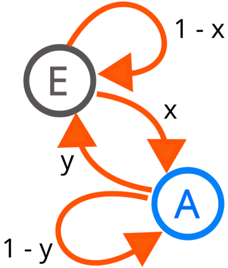


$$\mathbb{P}_a = \begin{bmatrix} 1 - x_a & x_a \\ y_a & 1 - y_a \end{bmatrix}$$

Control systems

Markov Decision Process

- enhanced MC $(E, Act, \{\mathbb{P}_a\})$;
- choose an action $a \Leftrightarrow$ choose a probability distribution $\mathbb{P}_a$.



$$\mathbb{P}_a = \begin{bmatrix} 1 - x_a & x_a \\ y_a & 1 - y_a \end{bmatrix}$$

Control!

Useful to describe partly random systems on which a decision maker has some control (e.g., coin toss or epidemic)

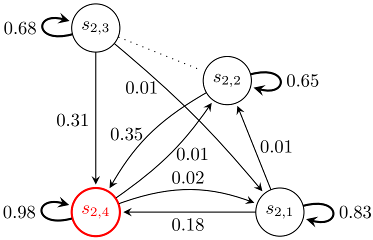
Building an MDP with an ODE...

SIR epidemiological model

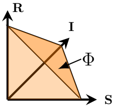
$$\begin{cases} \frac{\mathrm{d}S}{\mathrm{d}t} = \Lambda - \mu S - \omega \frac{IS}{N} \\ \frac{\mathrm{d}I}{\mathrm{d}t} = \omega \frac{IS}{N} - (\gamma + \mu) I \\ \frac{\mathrm{d}R}{\mathrm{d}t} = \gamma I - \mu R. \end{cases}$$

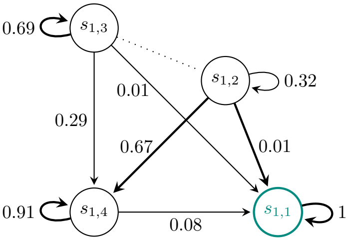
Different scenarios

Each scenario corresponds to a quadruplet $(\Lambda, \mu, \omega, \gamma)$



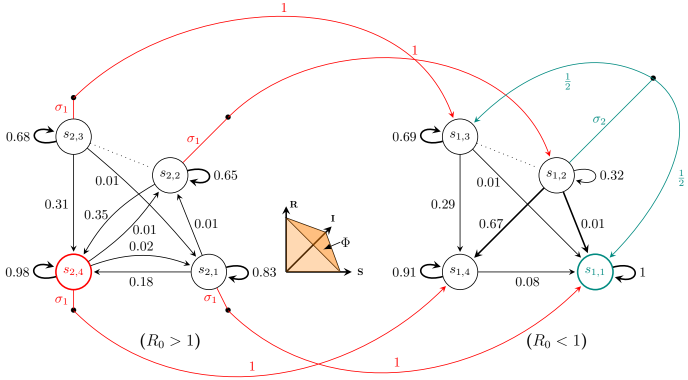
$(R_0 > 1)$





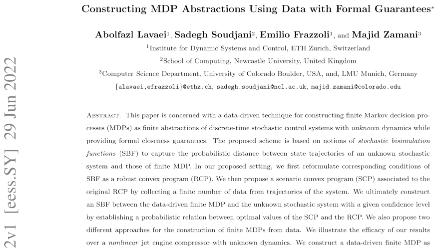
$(R_0 < 1)$

... by switching between MCs



σ_1, σ_2 are two actions that allow instant switching between states and MCs.

Bisimulations and MDP


Constructing MDP Abstractions Using Data with Formal Guarantees*

Abolfazl Lavaei[1], Sadegh Soudjani[2], Emilio Frazzoli[1], and Majid Zamani[3]
[1]Institute for Dynamic Systems and Control, ETH Zurich, Switzerland
[2]School of Computing, Newcastle University, United Kingdom
[3]Computer Science Department, University of Colorado Boulder, USA, and, LMU Munich, Germany
{alavaei,efrazzoli}@ethz.ch, sadegh.soudjani@ncl.ac.uk, majid.zamani@colorado.edu

Abstract. This paper is concerned with a data-driven technique for constructing finite Markov decision processes (MDPs) as finite abstractions of discrete-time stochastic control systems with *unknown* dynamics while providing formal closeness guarantees. The proposed scheme is based on notions of *stochastic bisimulation functions* (SBF) to capture the probabilistic distance between state trajectories of an unknown stochastic system and those of finite MDP. In our proposed setting, we first reformulate corresponding conditions of SBF as a robust convex program (RCP). We then propose a scenario convex program (SCP) associated to the original RCP by collecting a finite number of data from trajectories of the system. We ultimately construct an SBF between the data-driven finite MDP and the unknown stochastic system with a given confidence level by establishing a probabilistic relation between optimal values of the SCP and the RCP. We also propose two different approaches for the construction of finite MDPs from data. We illustrate the efficacy of our results over a *nonlinear* jet engine compressor with unknown dynamics. We construct a data-driven finite MDP as

Discrete-Time Stochastic Control Systems and MDPs

- $\Phi = (X, U, \varsigma, f)$ where
  - $X \subseteq \mathbb{R}^n$ is a set of states;
  - $U \in \mathbb{R}^m$ is a set of inputs;
  - ς is a sequence of *i.i.d.* random variables;
  - $f : X \times U \times \mathcal{V}_\varsigma \to X$ is a measurable function.
- $\hat{\Phi} = (\hat{X}, U, \hat{\mathsf{T}}_x)$ is its corresponding finite MDP.

Stochastic Bisimulation Function

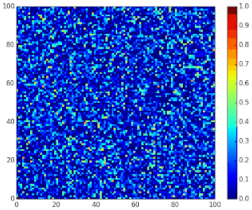
$S : X \times \hat{X} \to \mathbb{R}_+$ ensures the expected distance between systems remains short after a one-step-transition.

LAVAEI, A. et al. *Constructing MDP abstractions using data with formal guarantees.* IEEE Control Systems Letters. 2022, vol 7, p. 460–465.

Future work

Applications

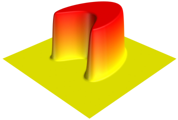
- Biology



Epidemic simulation. Each pixel can infect its eight immediate neighbors.

Extension

- Procedure simplification?
- PDE?
- partition?
- properties?



Visualisation of a solution to the 2D heat equation.

Thank you for your attention!