Efficiency through Uncertainty: Scalable Formal Synthesis for Stochastic Hybrid Systems

Nathalie Cauchi[1] Luca Laurenti[1] Morteza Lahijanian[2],
Alessandro Abate[1] Marta Kwiatkowska[1] Luca Cardelli[1,3]

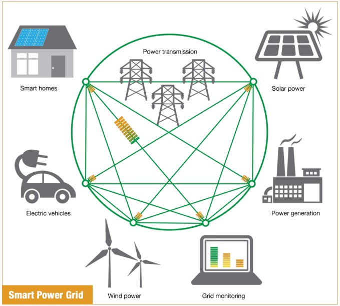
HSCC 2019

[1]University of Oxford

[2]University of Colorado Boulder

[3]Microsoft Research at Cambridge

Introduction



Safety critical systems with a multitude of requirements

Introduction

System dynamics:

- complex continuous dynamics
- discrete modes
- uncertainty

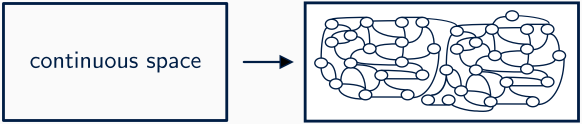
Specifications:

- can be formally defined

stochastic hybrid systems + formal methods

Classical approach

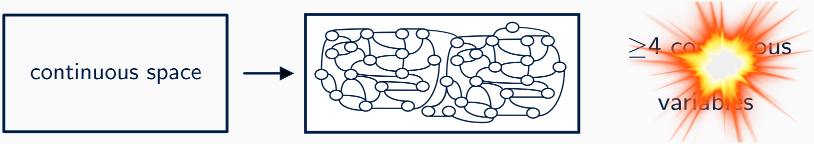
- abstract continuous variables to finite Markov Processes
- errors between original and abstract model (treated as separate parameter)



$\geq$4 continuous variables

Classical approach

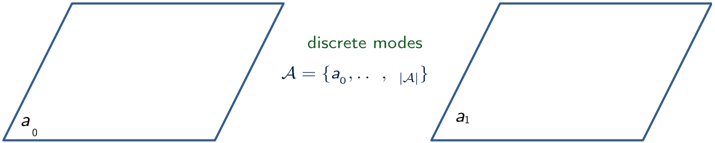
- abstract continuous variables to finite Markov Processes
- errors between original and abstract model (treated as separate parameter)
- generate conservative error bounds
  - error increases linearly with time
  - state-space explosion due to error explosion
- limited to finite time properties
- synthesis over abstract model (classical MDP synthesis)



Problem formulation: stochastic hybrid systems

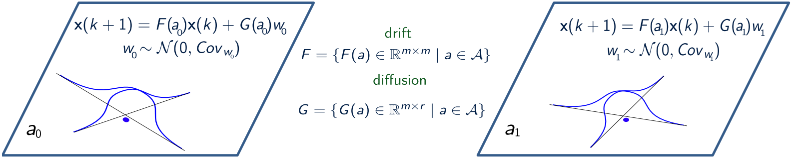
A (discrete-time) linear stochastic hybrid system (SHS) is a tuple

$$\mathcal{H} = (\mathcal{A}, F, G, \Upsilon, L)$$



A (discrete-time) linear stochastic hybrid system (SHS) is a tuple

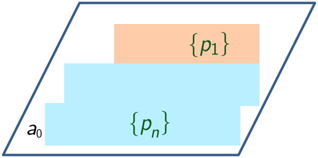
$$\mathcal{H} = (\mathcal{A}, F, G, \Upsilon, L)$$



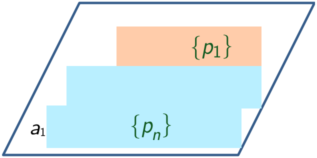
Problem formulation: stochastic hybrid systems

A (discrete-time) linear stochastic hybrid system (SHS) is a tuple

$$\mathcal{H} = (\mathcal{A}, F, G, \Upsilon, L)$$



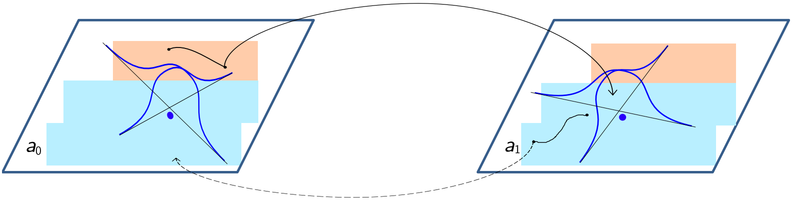
$\Upsilon = \{p_1, \ldots, p_n\}$
atomic propositions
$L : S \rightarrow 2^{\Upsilon}$
labeling function



Problem formulation: stochastic hybrid systems

A (discrete-time) linear stochastic hybrid system (SHS) is a tuple

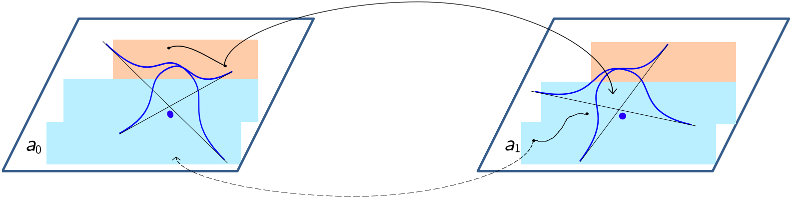
$$\mathcal{H} = (\mathcal{A}, F, G, \Upsilon, L)$$



Problem formulation: stochastic hybrid systems

A (discrete-time) linear stochastic hybrid system (SHS) is a tuple

$$\mathcal{H} = (\mathcal{A}, F, G, \Upsilon, L)$$



stochastic process

- a hybrid state of $\mathcal{H}$ is a pair $s = (a, x) \in S$
- evolution of $\mathcal{H}$ for $k \in \mathbb{Z}_{\geq 0}$ is a stochastic process $\mathsf{s}(k) = (\mathsf{a}(k), x(k)) \in S$

Problem formulation: stochastic hybrid systems

switching strategy

a function that assigns a discrete mode $a \in \mathcal{A}$ to a finite path $\omega_{\mathcal{H}}$ of the process $\mathbf{s}$

Problem formulation: stochastic hybrid systems

switching strategy

a function that assigns a discrete mode $a \in \mathcal{A}$ to a finite path $\omega_{\mathcal{H}}$ of the process **s**

transition kernel

for any measurable set $B \subseteq \mathbb{R}^m$, $x \in \mathbb{R}^m$, and $a \in \mathcal{A}$

$$T(B \mid x, a) = \int_B \mathcal{N}(t \mid F(a)x, G(a)^T Cov_w G(a))\, dt$$

Problem formulation: stochastic hybrid systems

switching strategy

a function that assigns a discrete mode $a \in \mathcal{A}$ to a finite path $\omega_{\mathcal{H}}$ of the process $\mathbf{s}$

transition kernel

for any measurable set $B \subseteq \mathbb{R}^m$, $x \in \mathbb{R}^m$, and $a \in \mathcal{A}$

$$T(B \mid x, a) = \int_B \mathcal{N}(t \mid F(a)x, G(a)^T Cov_w G(a))\, dt$$

temporal logic specifications

- deals with complex formal properties with boolean and temporal constraints (bounded and unbounded)

Problem statement

Given:

1. SHS $\mathcal{H}$
2. a compact set X
3. property expressed as a formula φ defined over regions of X
4. φ also requires system not to leave X

Problem statement

Given:

1. SHS $\mathcal{H}$
2. a compact set X
3. property expressed as a formula φ defined over regions of X
4. φ also requires system not to leave X

Find:

1. a switching strategy that maximizes the probability of satisfying φ for all initial states $s_0 \in \mathcal{A} \times X$

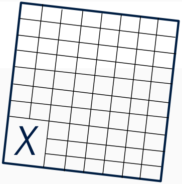
Proposed approach

We construct abstract model that captures all behaviours of SHS with to X and regions of interest

Proposed approach

We construct abstract model that captures all behaviours of SHS with to X and regions of interest

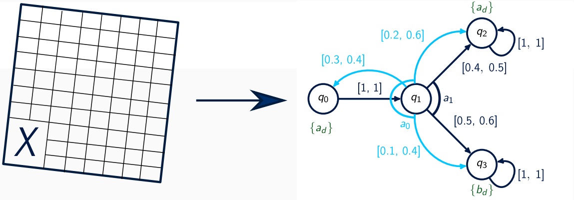
1. discretize set X according to dynamics of each mode



Proposed approach

We construct abstract model that captures all behaviours of SHS with to X and regions of interest

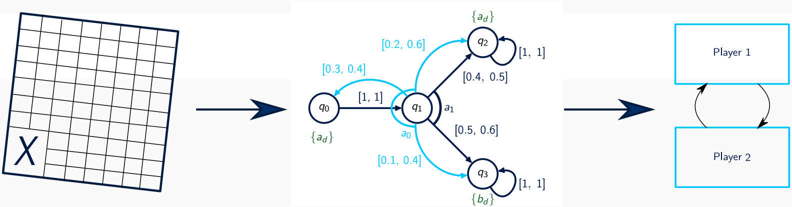
1. discretize set X according to dynamics of each mode
2. quantify error of abstraction and represent it as uncertainty in the abstraction



Proposed approach

We construct abstract model that captures all behaviours of SHS with to X and regions of interest

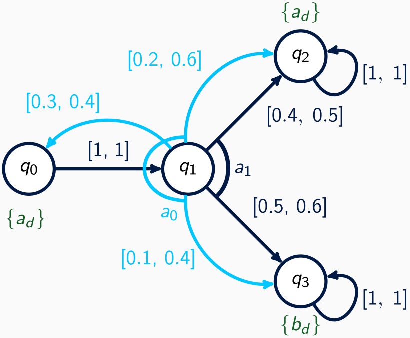
1. discretize set X
2. quantify error of abstraction and represent it as uncertainty
3. synthesise optimal strategy via stochastic games
4. strategy can be mapped back to SHS



Preliminaries: models

An IMDP is a tuple $\mathcal{I} = (Q, A, \check{P}, \hat{P})$

- Q is a finite set of states
- A is a finite set of actions
- $\check{P}(q, a, q')$ defines the lower bound of the transition probability
- $\hat{P}(q, a, q')$ defines the upper bound of the transition probability
- Υ is a finite set of atomic propositions
- $L : Q \to 2^{\Upsilon}$ is a labeling function

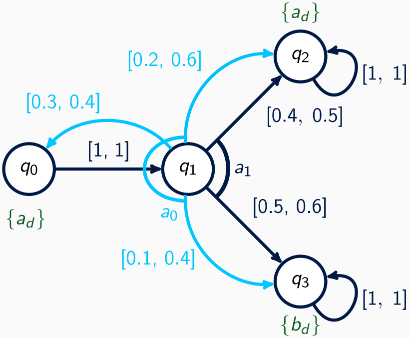


An IMDP is a tuple $\mathcal{I} = (Q, A, \check{P}, \hat{P})$

- a feasible distribution reachable from q by a if

$$\check{P}(q, a, q') \leq \gamma_q^a(q') \leq \hat{P}(q, a, q')$$

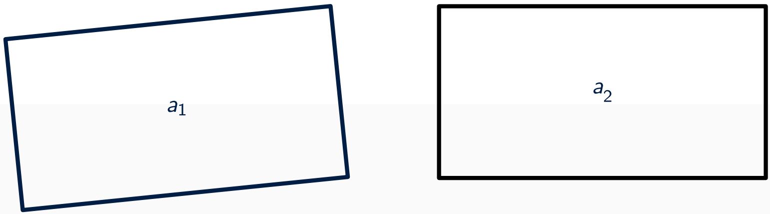
for each state $q' \in Q$



SHS abstractions

We abstract the SHS to an IMDP

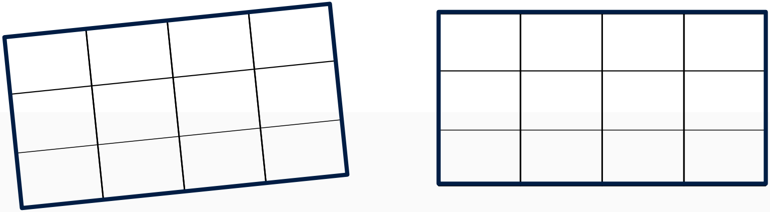
$$\mathcal{I} = (Q, A, \check{P}, \hat{P}, \Upsilon, L)$$



SHS abstractions

We abstract the SHS to an IMDP

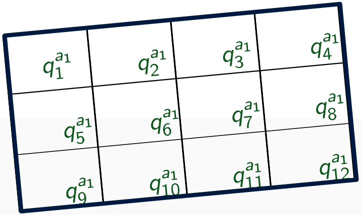
$$\mathcal{I} = (Q, A, \check{P}, \hat{P}, \Upsilon, L)$$

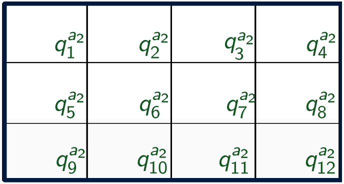


SHS abstractions

We abstract the SHS to an IMDP

$$\mathcal{I} = (Q, A, \check{P}, \hat{P}, \Upsilon, L)$$





$\mathcal{Q}^{a_1} = \{q_1^{a_1}, \ldots, q_{12}^{a_1}\}$

$\mathcal{Q}^{a_2} = \{q_1^{a_2}, \ldots, q_{12}^{a_2}\}$

$$\mathcal{Q} = \{\mathcal{Q}^{a_1} \cup \mathcal{Q}^{a_1} \cup \{q_u\}\}$$

SHS abstractions

We abstract the SHS to an IMDP

$$\mathcal{I} = (Q, A, \check{P}, \hat{P}, \Upsilon, L)$$

- A are the set of modes of $\mathcal{H}$: $A(q) = A \; \forall q \in Q$

SHS abstractions

We abstract the SHS to an IMDP

$$\mathcal{I} = (Q, A, \check{P}, \hat{P}, \Upsilon, L)$$

- A are the set of modes of $\mathcal{H}$: $A(q) = A \ \forall q \in Q$
- one-step transition probability : $T(q|x, a)$
- but $q \in Q$ correspond to regions in $\mathcal{H}$
  - range of feasible transition probabilities to region q
  - bound feasible transitions to get $\check{P}, \hat{P}$

SHS abstractions

We abstract the SHS to an IMDP

$$\mathcal{I} = (Q, A, \check{P}, \hat{P}, \Upsilon, L)$$

- A are the set of modes of $\mathcal{H}$: $A(q) = A \; \forall q \in Q$
- one-step transition probability : $T(q|x, a)$

$$\gamma_{q_i}^{a}(q_j) \leq \min_{x \in q_i} T(q_j|x, a)$$

$$\gamma_{q_i}^{a}(q_j) \geq \max_{x \in q_i} T(q_j|x, a)$$

Computation of imdp

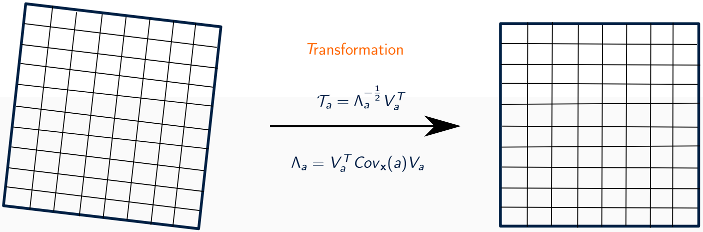
How do we efficiently compute

$$\min_{x \in \boldsymbol{q}_i} T(q_j \mid x, a), \quad \max_{x \in \boldsymbol{q}_i} T(q_j \mid x, a)?$$

Computation of imdp

How do we efficiently compute

$$\min_{x \in \boldsymbol{q}_i} T(q_j \mid x, a), \quad \max_{x \in \boldsymbol{q}_i} T(q_j \mid x, a)?$$



For process $\mathbf{x}$ in mode $a \in \mathcal{A}$

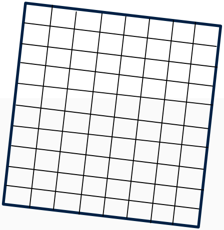
Hyper rectangles

$[v_l^{(1)}, v_u^{(1)}] \times \cdots \times [v_l^{(m)}, v_u^{(m)}]$

Computation of imdp

How do we efficiently compute

$$\min_{x \in \boldsymbol{q}_i} T(q_j \mid x, a), \quad \max_{x \in \boldsymbol{q}_i} T(q_j \mid x, a)?$$

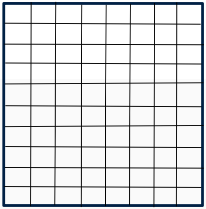


For process $\mathbf{x}$ in mode $a \in \mathcal{A}$

Transformation

$$\mathcal{T}_a = \Lambda_a^{-\frac{1}{2}} V_a^T$$

$$\Lambda_a = V_a^T Cov_{\mathbf{x}}(a) V_a$$



Analytical solution

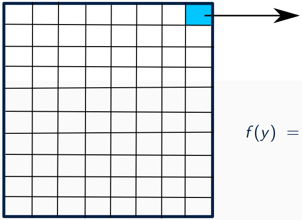
$$[v_l^{(1)}, v_u^{(1)}] \times \cdots \times [v_l^{(m)}, v_u^{(m)}]$$

$$T(q_j \mid x, a) = \frac{1}{2^m} \prod_{\mathrm{n}=1}^{m} \Big(\mathrm{erf}\big(\frac{y^{(\mathrm{n})} - v_l^{(\mathrm{n})}}{\sqrt{2}}\big) - \mathrm{erf}\big(\frac{y^{(\mathrm{n})} - v_u^{(\mathrm{n})}}{\sqrt{2}}\big) \Big)$$

Computation of imdp

How do we efficiently compute

$$\min_{x \in q_i} T(q_j \mid x, a), \quad \max_{x \in q_i} T(q_j \mid x, a)?$$



$$\min_{x \in q_i} T(q_j \mid x, a) = \min_{y \in Post(q_i', \mathcal{T}_a)} f(y),$$
$$\max_{x \in q_i} T(q_j \mid x, a) = \max_{y \in Post(q_i', \mathcal{T}_a)} f(y).$$

$$f(y) = \frac{1}{2^m} \prod_{n=1}^{m} \left(\operatorname{erf}\left(\frac{y^{(n)} - v_l^{(n)}}{\sqrt{2}}\right) - \operatorname{erf}\left(\frac{y^{(n)} - v_u^{(n)}}{\sqrt{2}}\right) \right)$$

Optimisation via KKT or Gradient descent

Computation of imdp

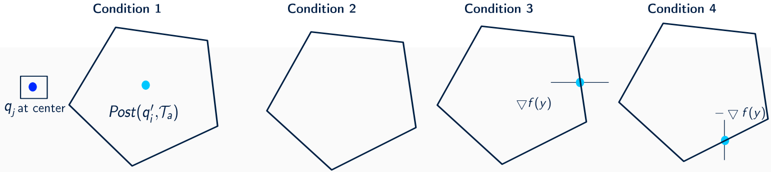
Optimisation via KKT conditions

- solving systems of non-linear equations
- efficient and exact for low-dimensional system
- number of vertices to check grows exponentially with dimensions

Computation of imdp

Optimisation via KKT conditions

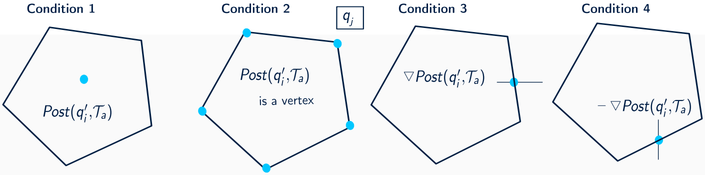
- solving systems of non-linear equations
- efficient and exact for low-dimensional system
- number of vertices to check grows exponentially with dimensions



Computation of imdp

Optimisation via KKT conditions

- solving systems of non-linear equations
- efficient and exact for low-dimensional system
- number of vertices to check grows exponentially with dimensions



Computation of imdp

Optimisation via gradient descent method

$$f(y) = \frac{1}{2^m}\left[\prod_{i=1}^{m}\left(\mathrm{erf}\left(\frac{y^{(i)} - v_l^{(i)}}{\sqrt{2}}\right) - \mathrm{erf}\left(\frac{y^{(i)} - v_u^{(i)}}{\sqrt{2}}\right)\right)\right]$$

Computation of imdp

Optimisation via gradient descent method

$$f(y) = -\frac{1}{2^m}\log\left[\prod_{i=1}^{m}\left(\mathrm{erf}\left(\frac{y^{(i)} - v_l^{(i)}}{\sqrt{2}}\right) - \mathrm{erf}\left(\frac{y^{(i)} - v_u^{(i)}}{\sqrt{2}}\right)\right)\right]$$

- $f(y)$ has the property of being log-concave
- can use standard convex optimisation techniques
- allows for scaling to high dimensions

SHS abstractions

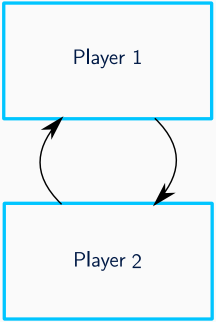
We abstract the SHS to an IMDP

$$\mathcal{I} = (Q, A, \check{P}, \hat{P}, \Upsilon, L)$$

- associate labels with corresponding region R in X
- when discretisation does not respect R
  - add extra labels (conservatively)
  - converting φ into negation normal form (NNF)
  - associate labels with negation of propositions
  - under approximate this region

Strategy synthesis

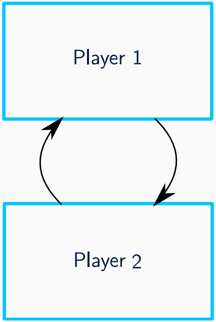
- uncertainties in $\mathcal{I}$: nondeterministic choice of transition probability from one IMDP state to another under a given action
- synthesis task interpreted as a $2\,\frac{1}{2}$ player stochastic game
- the set of actions of player 2 is continuous



1. chooses an action $a \in A$ at state $q \in Q$
2. chooses a feasible transition probability distribution

Strategy synthesis

- uncertainties in $\mathcal{I}$: nondeterministic choice of transition probability from one IMDP state to another under a given action
- synthesis task interpreted as a $2\ \frac{1}{2}$ player stochastic game
- the set of actions of player 2 is continuous



objectives of Players 1:

maximize probability of satisfying φ

adversarial game

objectives of Players 2 :

minimize probability of satisfying φ

Case studies

- implemented the abstraction and synthesis algorithms
- test performance using three case studies
- analysis on the abstraction error generated

$$\varepsilon_{max} = \max_{q \in Q} \left(\hat{p}(q) - \check{p}(q) \right)$$

  min $\check{p}(q)$ and max $\hat{p}(q)$ probabilities of satisfaction for each state q

- case studies are run on an Intel Core i7-8550U CPU at 1.80GHz $\times$ 8 machine with 8 GB of RAM

Case study 1: formal verification

model

1 discrete mode: $A = \{a_1\}$

$$x(k+1) = \begin{pmatrix} 0.85 & 0 \\ 0 & 0.90 \end{pmatrix} x(k) + \begin{pmatrix} 0.15 & 0 \\ 0 & 0.05 \end{pmatrix} w(k)$$

Case study 1: formal verification

model

1 discrete mode: $A = \{a_1\}$

$$x(k+1) = \begin{pmatrix} 0.85 & 0 \\ 0 & 0.90 \end{pmatrix} x(k) + \begin{pmatrix} 0.15 & 0 \\ 0 & 0.05 \end{pmatrix} w(k)$$

verification of a safety property

$$\varphi_1 = \mathcal{G}^{\leq K}\{[-1,1] \times [-1,1]\}$$

comparison against state of the art tool FAUST[2]

Case study 1: formal verification

results:

$$\varphi_1 = \mathcal{G}^{\leq K=2}\{[-1,1]\times[-1,1]\}$$

Tool Method	**Impl. Platform**	$\lvert\bar{Q}\rvert$ (states)	**Time taken** (secs)	**Error** ε_{max}
IMDP (KKT)	MATLAB	361	19.789	0.211
FAUST[2]	MATLAB	361	108.265	1.000
IMDP (KKT)	MATLAB	625	145.563	0.163
FAUST[2]	MATLAB	625	285.795	1.000
IMDP (KKT)	MATLAB	2601	28127.256	0.082
FAUST[2]	MATLAB	2601	5274.578	0.995
IMDP (KKT)	MATLAB	3721	Time out[a]	-
FAUST[2]	MATLAB	3721	11285.313	0.832

[a] 9 hours+ and no solution

Case study 1: formal verification

results:

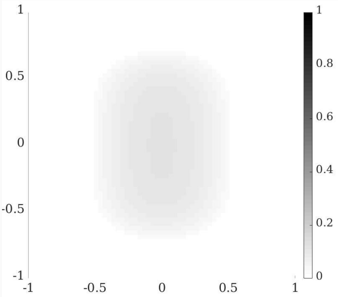
$$\varphi_1 = \mathcal{G}^{\leq K=2}\{[-1,1]\times[-1,1]\}$$

Tool Method	Impl. Platform	$\lvert\bar{Q}\rvert$ (states)	Time taken (secs)	Error $\varepsilon_{\max}$
IMDP (KKT)	MATLAB	361	19.789	0.211
FAUST[2]	MATLAB	361	108.265	1.000
IMDP (KKT)	MATLAB	625	145.563	0.163
FAUST[2]	MATLAB	625	285.795	1.000
IMDP (KKT)	MATLAB	2601	28127.256	0.082
FAUST[2]	MATLAB	2601	5274.578	0.995
IMDP (KKT)	MATLAB	3721	Time out[a]	-
FAUST[2]	MATLAB	3721	11285.313	0.832

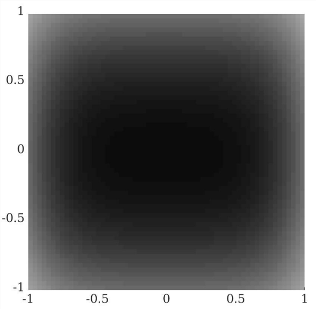
[a] 9 hours+ and no solution

Tool Method	Impl. Platform	$\lvert\bar{Q}\rvert$ (states)	Time taken (secs)	Error $\varepsilon_{\max}$
IMDP (GD)	C++	361	29.003	0.211
FAUST[2]	C++	361	136.71	1.000
IMDP (GD)	C++	625	117.741	0.163
FAUST[2]	C++	625	302.900	1.000
IMDP (GD)	C++	2601	2939.050	0.082
FAUST[2]	C++	2601	3305.490	0.995
IMDP (GD)	C++	3721	3973.28	0.068
FAUST[2]	C++	3721	7537.750	0.832

results:



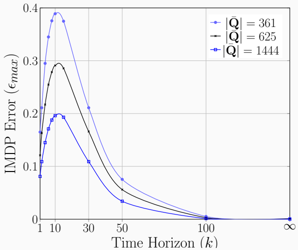
FAUST[2]

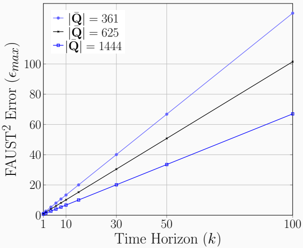


IMDP

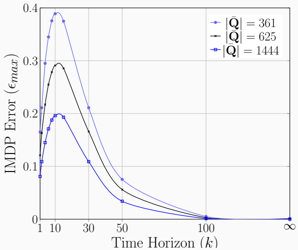
Case study 1: formal verification

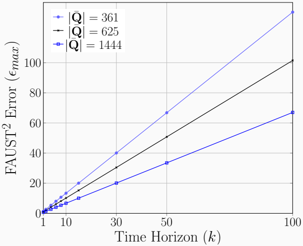
results: $\varphi_1 = \mathcal{G}^{\leq k}\{[-1,1] \times [-1,1]\}$





results: $\varphi_1 = \mathcal{G}^{\leq k}\{[-1,1]\times[-1,1]\}$





- error embedded within abstraction
- performs computations according to feasible transition probabilities
- abstraction errors does not explode with time

Case study 2: strategy synthesis

model

2 discrete modes: $A = \{a_1, a_2\}$

$$x(k+1)_{a_1} = \begin{pmatrix} 0.1 & 0.9 \\ 0.8 & 0.2 \end{pmatrix} x(k)_{a_1} + \begin{pmatrix} 0.3 & 0.1 \\ 0.1 & 0.2 \end{pmatrix} w_{a_1}(k)$$

$$x(k+1)_{a_2} = \begin{pmatrix} 0.8 & 0.2 \\ 0.1 & 0.9 \end{pmatrix} x(k)_{a_2} + \begin{pmatrix} 0.2 & 0 \\ 0 & 0.1 \end{pmatrix} w_{a_2}(k)$$

Case study 2: strategy synthesis

model

2 discrete modes: $A = \{a_1, a_2\}$

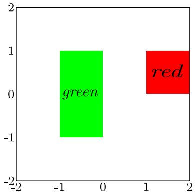
$$x(k+1)_{a_1} = \begin{pmatrix} 0.1 & 0.9 \\ 0.8 & 0.2 \end{pmatrix} x(k)_{a_1} + \begin{pmatrix} 0.3 & 0.1 \\ 0.1 & 0.2 \end{pmatrix} w_{a_1}(k)$$

$$x(k+1)_{a_2} = \begin{pmatrix} 0.8 & 0.2 \\ 0.1 & 0.9 \end{pmatrix} x(k)_{a_2} + \begin{pmatrix} 0.2 & 0 \\ 0 & 0.1 \end{pmatrix} w_{a_2}(k)$$

synthesis of a maximising switching strategy

$$\varphi_2 = \neg red \; \mathcal{U} \; green$$

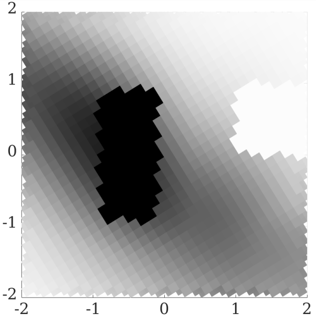
within the set $X = [-2, 2] \times [-2, 2]$



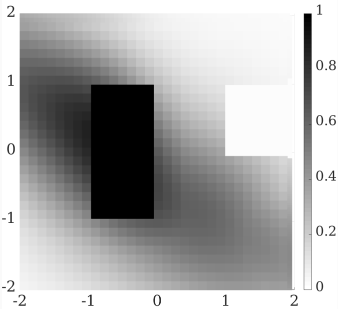
Case study 2: strategy synthesis

results:

- total number of strates: 3612
- computational time: 5434 [s]

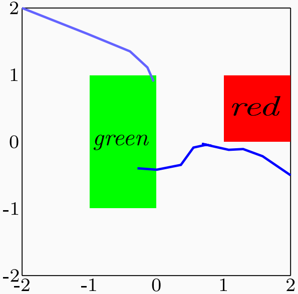


lower bound probability for a_1

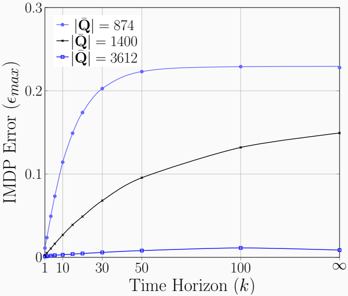


lower bound probability for a_2

results:



original set X with simulated trajectories



maximum error incurred in satisfying φ_2

Case study 3: scaling in continuous dimension of model

model with d continuous variables

1 discrete mode: $A = \{a_1\}$

$$x(k+1) = -0.95\mathbf{1}_d x(k) + 0.1\mathbf{1}_d w(k)$$

Case study 3: scaling in continuous dimension of model

model with d continuous variables

1 discrete mode: $A = \{a_1\}$

$$x(k+1) = -0.95\mathbf{1}_d x(k) + 0.1\mathbf{1}_d w(k)$$

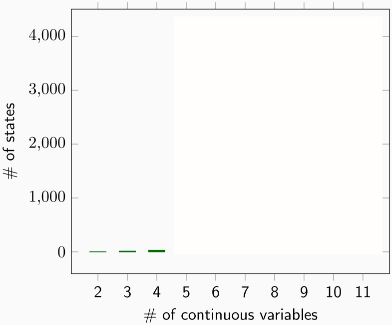
verification of a safety property

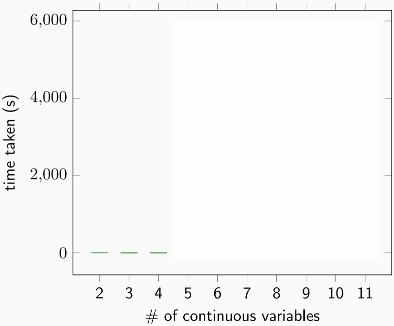
$$\varphi_1 = \mathcal{G}^{\leq k=50}[-1, 1]^d$$

Case study 3: scaling in continuous dimension of model

results:

- current state of the art

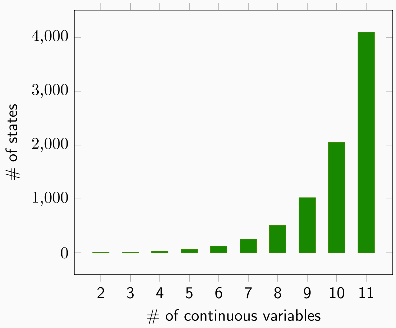


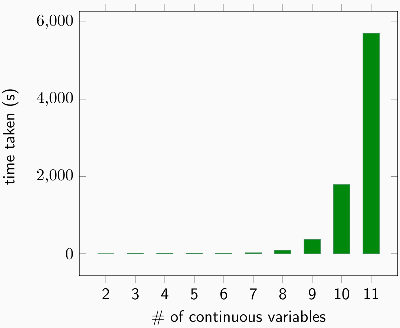


Case study 3: scaling in continuous dimension of model

results:

- number of dimensions that can now be analysed
- adaptive refinement resulting in max error of 0.03

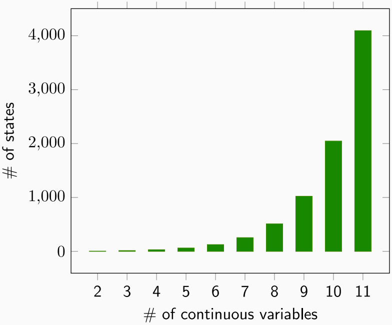


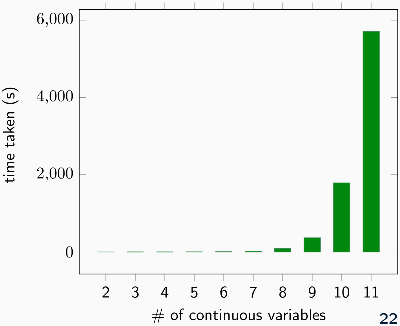


Case study 3: scaling in continuous dimension of model

results:

- number of dimensions that can now be analysed
- adaptive refinement resulting in max error of 0.03
- manageable state spaces
- remarkable improvement





Conclusion

contributions

- theoretical & computational technique for analysis of SHS
- precise and compact abstractions
- algorithms embedded within new tool StocHy presented at TACAS'19

Conclusion

contributions

- theoretical & computational technique for analysis of SHS
- precise and compact abstractions
- algorithms embedded within new tool StocHy presented at TACAS'19

future work

- synthesis for more complex and even multi-objective properties
- analysis of continuous-time SHS (current work in progress)

Conclusion

contributions

- theoretical & computational technique for analysis of SHS
- precise and compact abstractions
- algorithms embedded within new tool StocHy presented at TACAS'19

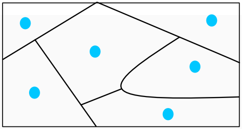
Thank you!

FAUST[2]

abstraction into MDP

- define desired abstraction error
- grid state-space (uniform or adaptive)
- compute transition probabilities via marginalisation
- hinges on computation of Lipschitz constants (h_s)
- N step error
$$\varepsilon = h_s \delta \mathcal{L}(A) N$$

  δ max diameter of partition, $\mathcal{L}(A)$ volume of set



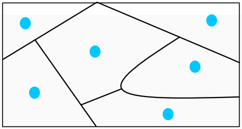
FAUST^2

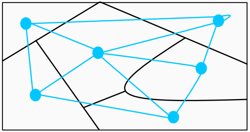
abstraction into MDP

- define desired abstraction error
- grid state-space (uniform or adaptive)
- compute transition probabilities via marginalisation
- hinges on computation of Lipschitz constants (h_s)
- N step error

$$\varepsilon = h_s \delta \mathcal{L}(A) N$$

δ max diameter of partition, $\mathcal{L}(A)$ volume of set



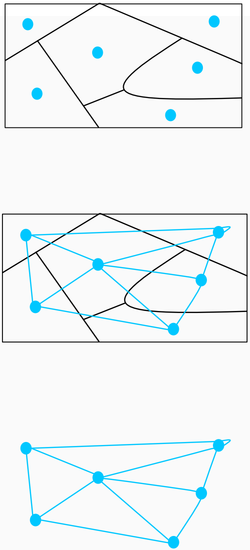


FAUST^2

abstraction into MDP

- define desired abstraction error
- grid state-space (uniform or adaptive)
- compute transition probabilities via marginalisation
- hinges on computation of Lipschitz constants (h_s)
- N step error
$$\varepsilon = h_s \delta \mathcal{L}(A) N$$

  δ max diameter of partition, $\mathcal{L}(A)$ volume of set



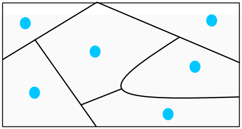
FAUST^2

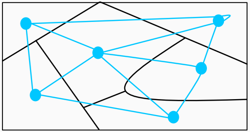
abstraction into MDP

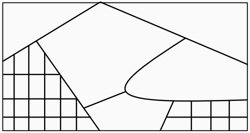
- define desired abstraction error
- grid state-space (uniform or adaptive)
- compute transition probabilities via marginalisation
- hinges on computation of Lipschitz constants (h_s)
- N step error

$$\varepsilon = h_s \delta \mathcal{L}(A) N$$

δ max diameter of partition, $\mathcal{L}(A)$ volume of set







Labelling

- conservatively overapproximate discretizations of $\mathcal{A} \times X$ that donot respect regions in $R = \{r_1, \ldots, r_n\}$
- represent each region by its complement relative to X
- Let $r_{n+i} = X \setminus r_i$ be the complement region of r_i with respect to X. We associate to each r_{n+i} a new atomic proposition p_{n+i} for $1 \leq i \leq n$.

$$\bar{\Upsilon} = \Upsilon \cup \{p_{n+1}, \ldots, p_{2n}\}$$

Then, we design $L : Q \to 2^{\bar{\Upsilon}}$ of $\mathcal{I}$ such that

$$p_i \in L(q) \quad \Leftrightarrow \quad q \subseteq r_i$$

for all $q \in \bar{Q}$ and $0 \leq i \leq 2n$, and $L(q_u) = \emptyset$