Automatic Verification of Stochastic Processes: Certification of Building Automation Systems

Nathalie Cauchi

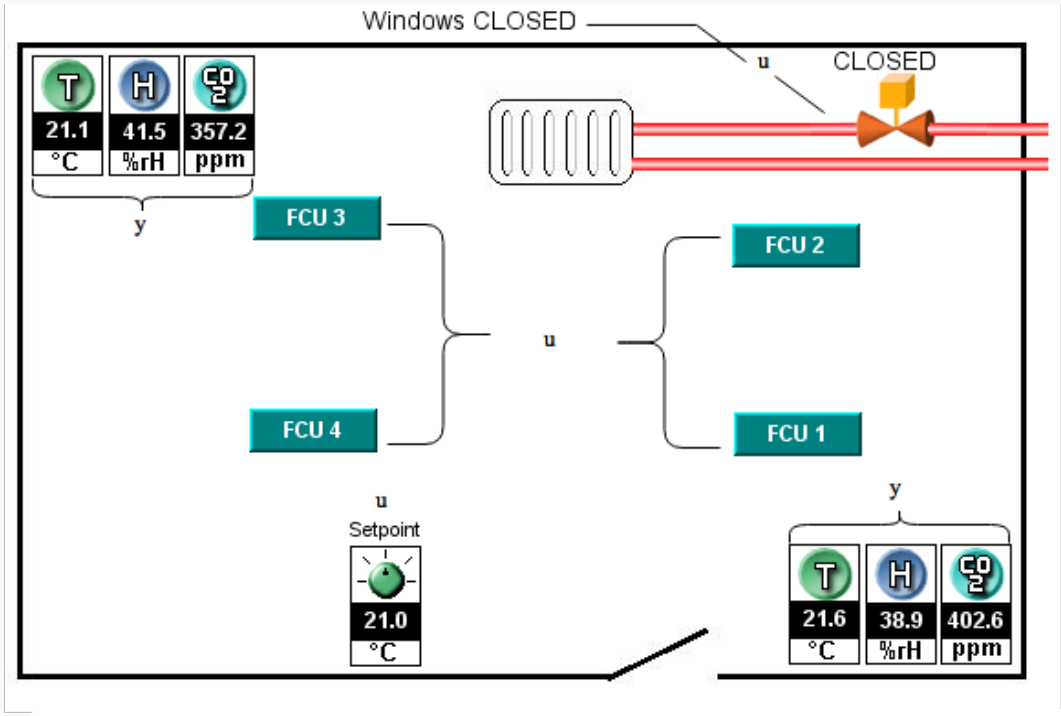
Michaelmas 2019

Introduction

Complex systems with a multitude of requirements

Introduction

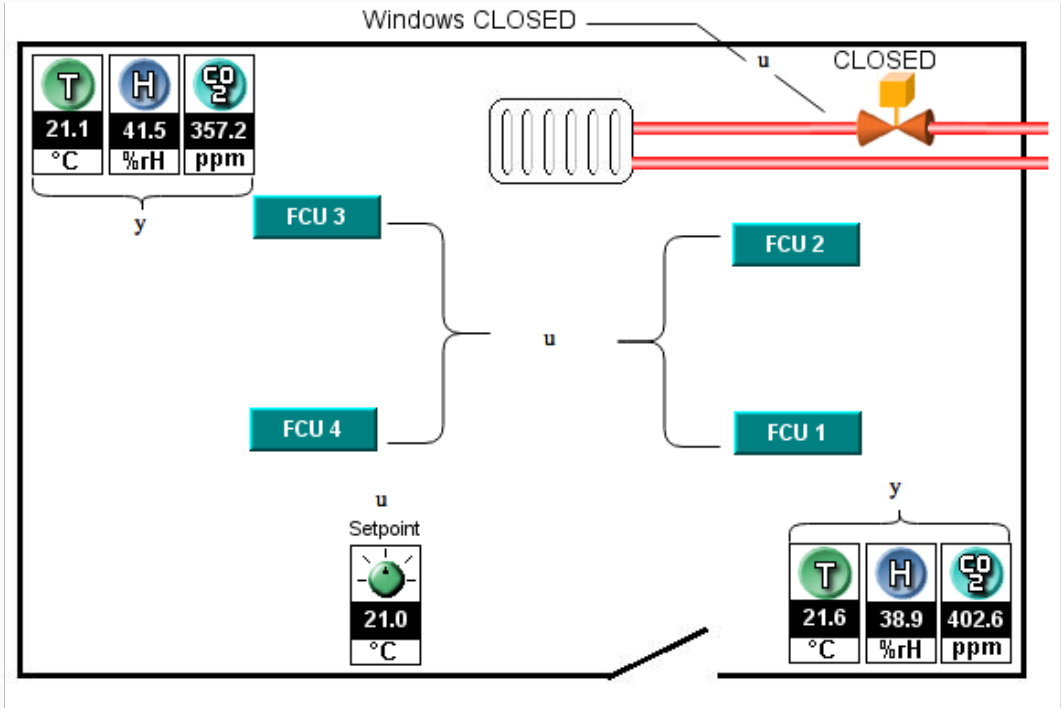
How do we construct a general modelling framework?



modelling uncertainty

Introduction

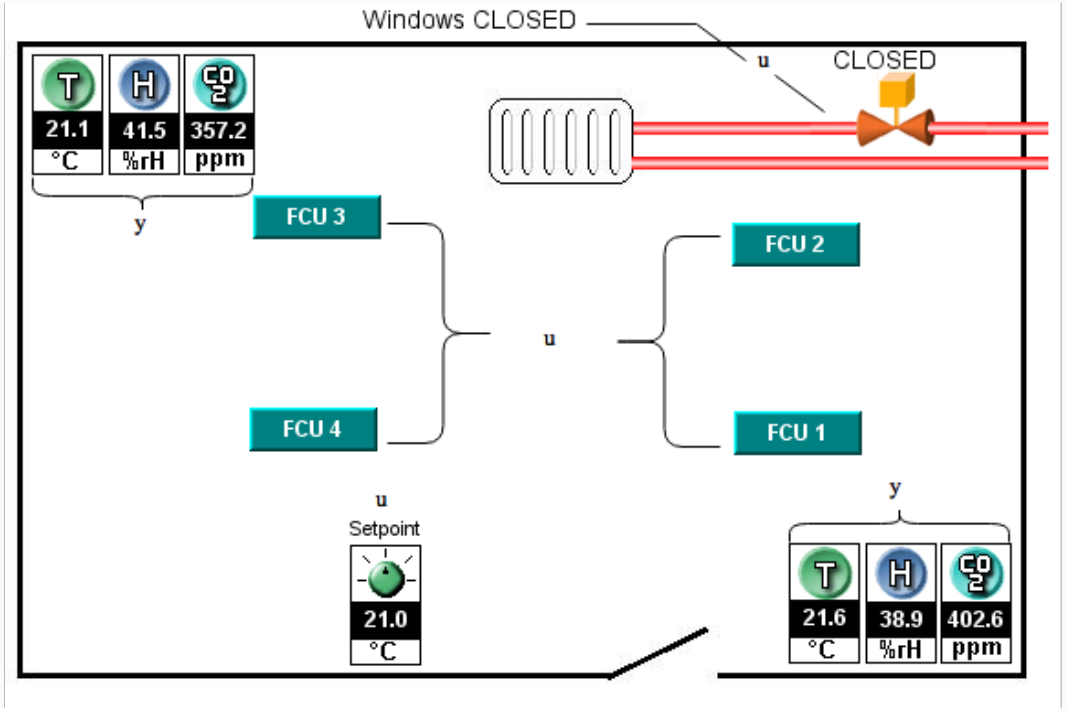
How do we construct a general modelling framework?



digital controls & continuous processes

Introduction

How do we construct a general modelling framework?



both theoretically sound & practical

Introduction

How do we couple certification goals with the modelling framework?

Introduction

How do we couple certification goals with the modelling framework?

System dynamics:

- complex continuous dynamics
- discrete modes
- uncertainty

Specifications:

- can be formally defined

stochastic hybrid systems + formal methods

Certification

From certification requirements into formal specifications?

Certification

From certification requirements into formal specifications?

We split reasoning over different levels:

- high-level:
  - reasoning over the Building Automation System (BAS) as a whole
  - employ key performance metrics
- low-level
  - reasoning over individual subcomponents
  - probabilistic reachability based

High-level certification

High-level certification: maintenance application

- reason over the whole structure of BAS
- zoom into the maintenance application

High-level certification: maintenance application

- reason over the whole structure of BAS
- zoom into the maintenance application

Maintenance

An activity in which repairing is carried out at certain intervals to extend the useful life of the machine

Preventative maintenance

Implemented at predetermined time intervals and prescribed guidelines to prevent the degradation of function

High-level certification: maintenance application

- reason over the whole structure of BAS
- zoom into the maintenance application

Maintenance

An activity in which repairing is carried out at certain intervals to extend the useful life of the machine

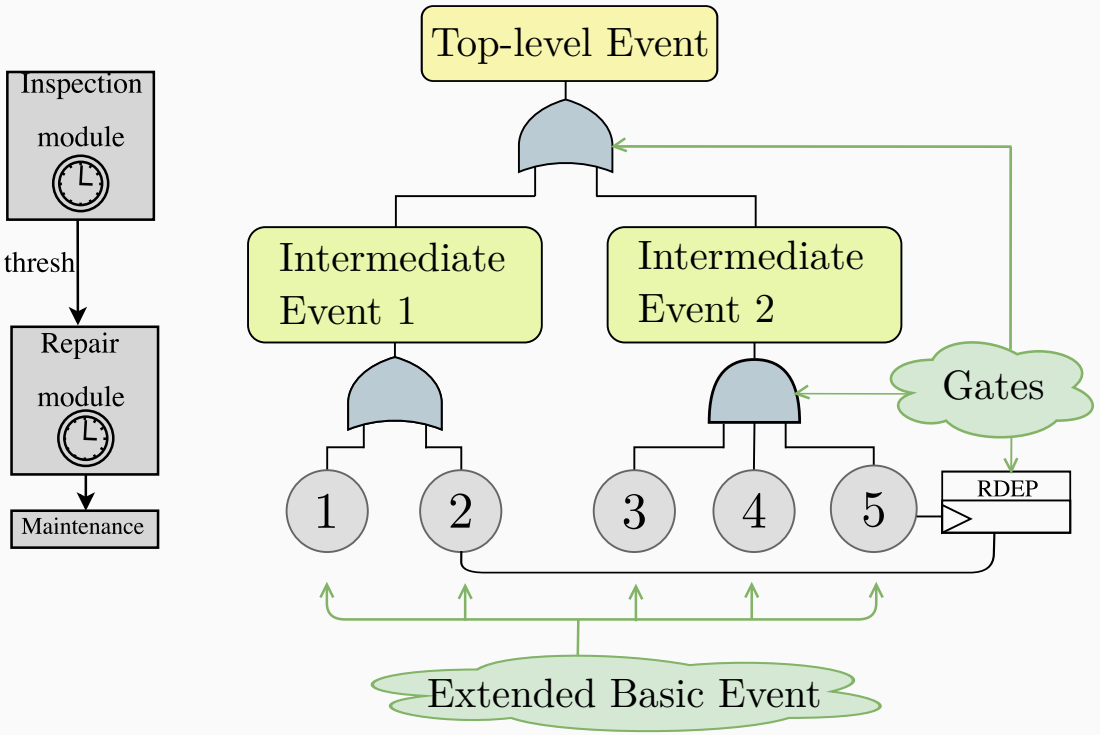
Preventative maintenance

Implemented at predetermined time intervals and prescribed guidelines to prevent the degradation of function

- novel framework via fault maintenance trees

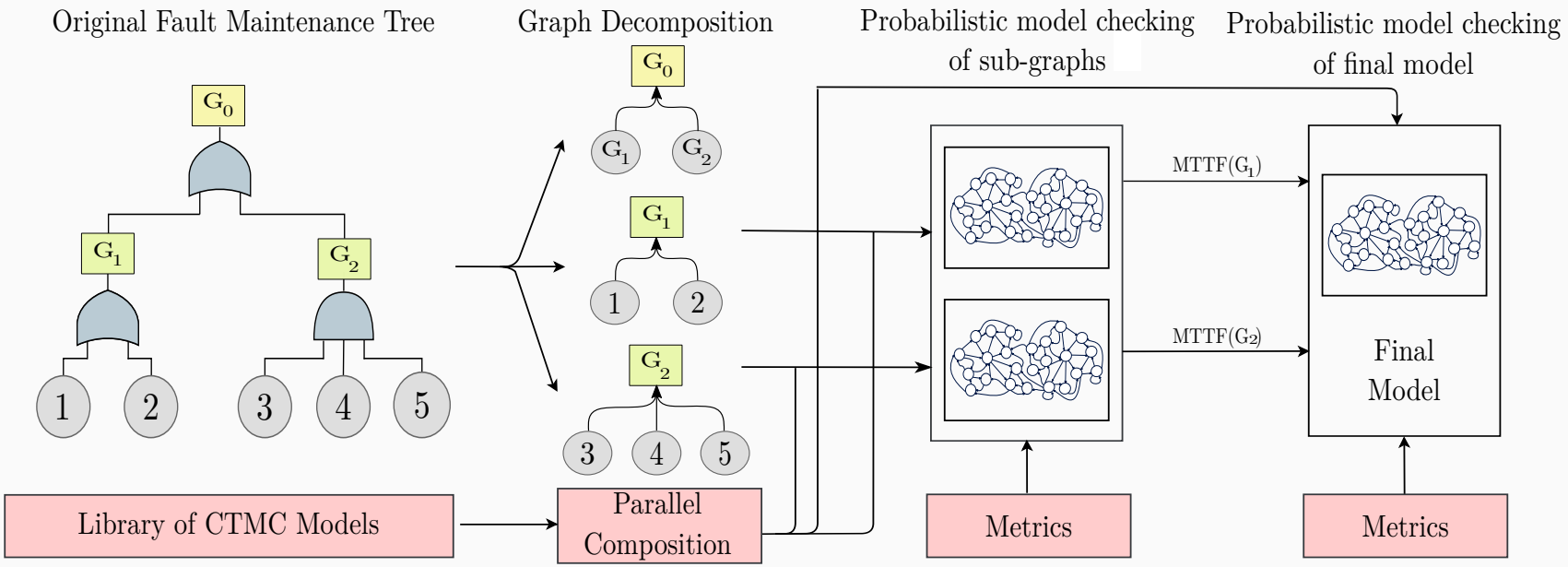
High-level certification: fault maintenance trees

- quantitative and qualitative reasoning
- encompass both degradation models and maintenance actions



High-level certification: fault maintenance trees

How do we reason over fault maintenance trees?



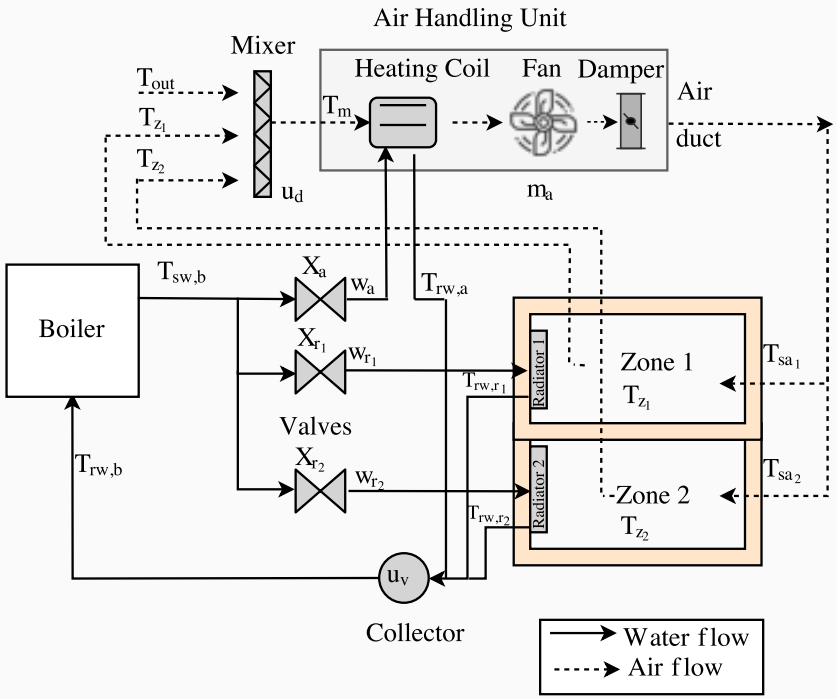
High-level certification: fault maintenance trees

How do we reason over fault maintenance trees?

1. **Reliability** -“probability of system not failing within a given time window"
2. **Availability** - “expected time fraction of time window that the system is functioning"
3. **Expected cost** - “ expected costs (operational + maintenance + inspection + failure cost)"
4. **Expected number of failures** - “expected failures over time window"

High-level certification: fault maintenance trees

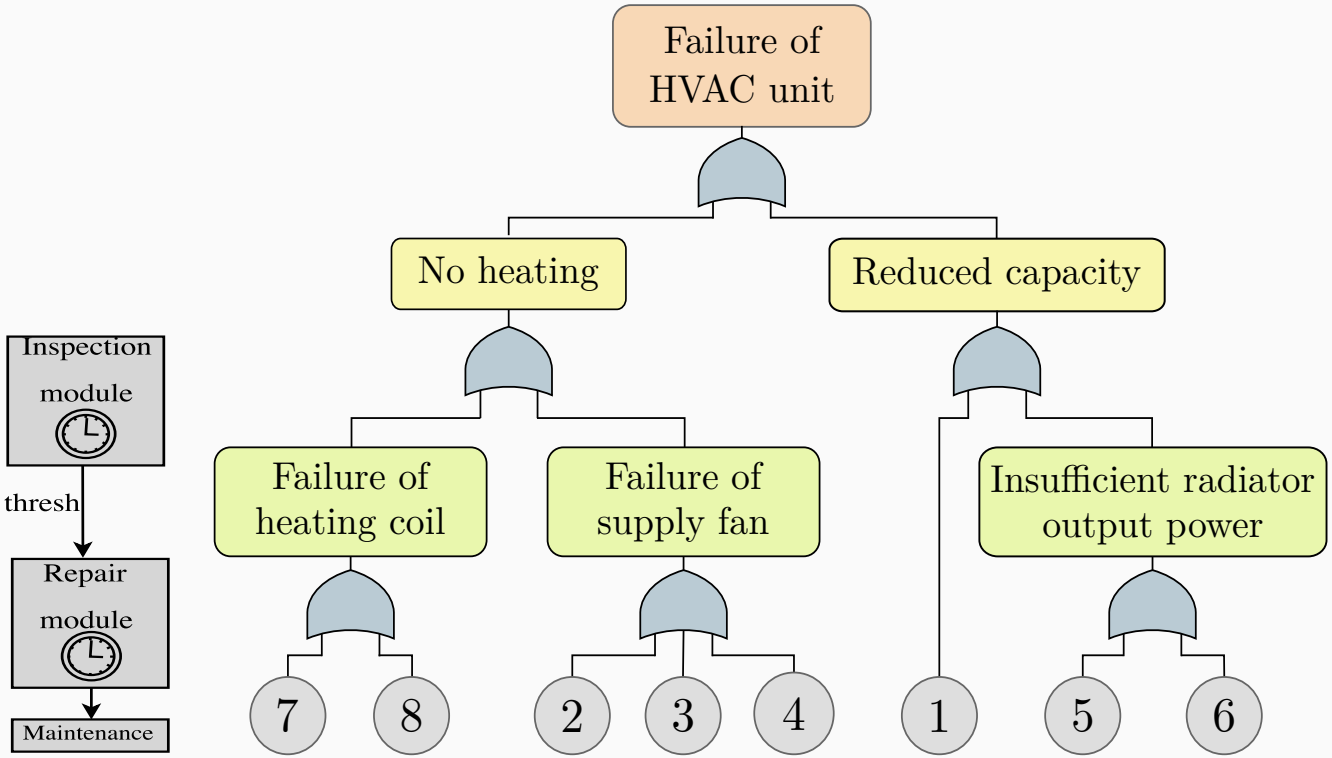
How do we apply framework to BAS?



ID	Component	N	MTTF
1	AHU damper broken	4	20
2	Fan motor failure	3	35
3	Supply fan obstructed	4	31
4	Fan bearing failure	6	17
5	Radiator failure	4	25
6	Radiator stuck valve	2	10
7	Heater stuck valve	2	10
8	Heat pump failure	4	20

High-level certification: fault maintenance trees

How do we apply framework to BAS?



High-level certification: fault maintenance trees

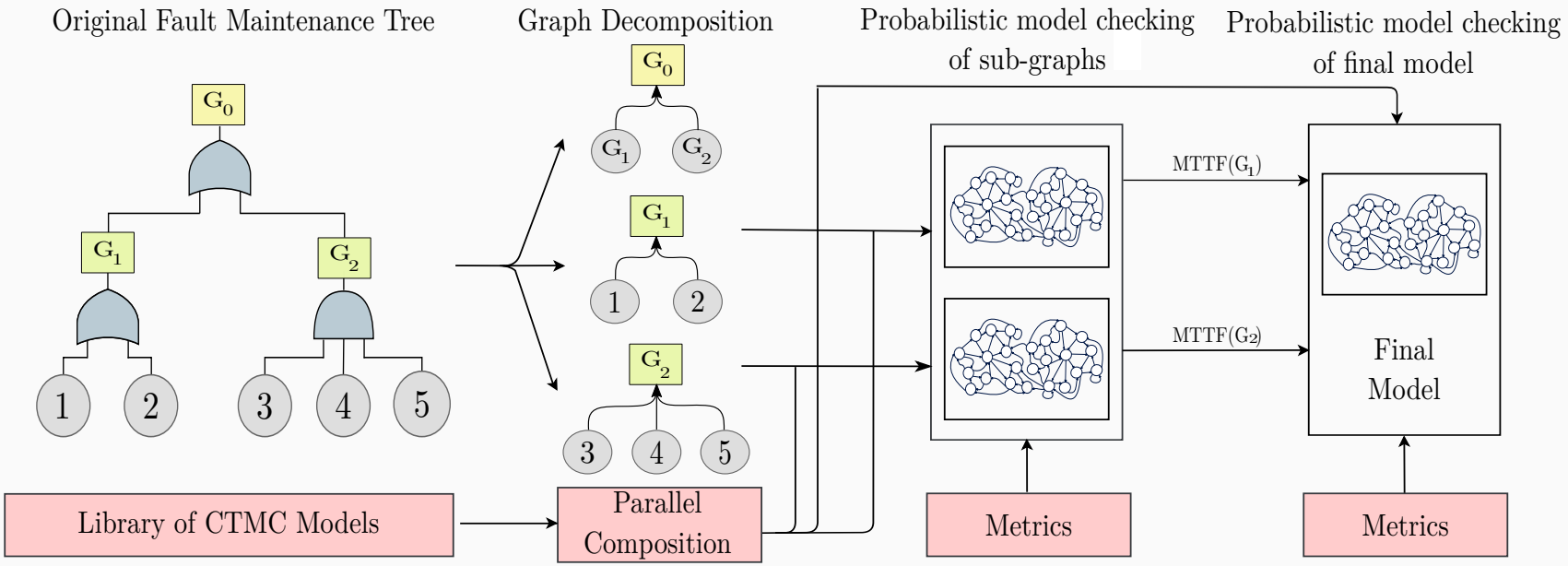
How do we apply framework to BAS?

- preventative maintenance strategies

Strategy index	T_{rep}	T_{oh}	T_{insp}
M_0	2 years	-	1 year
M_1	5 years	-	2 years
M_2	2 years	5 years	-
M_3	2 years	15 years	0.5 years
M_4	4 years	30 years	1 year

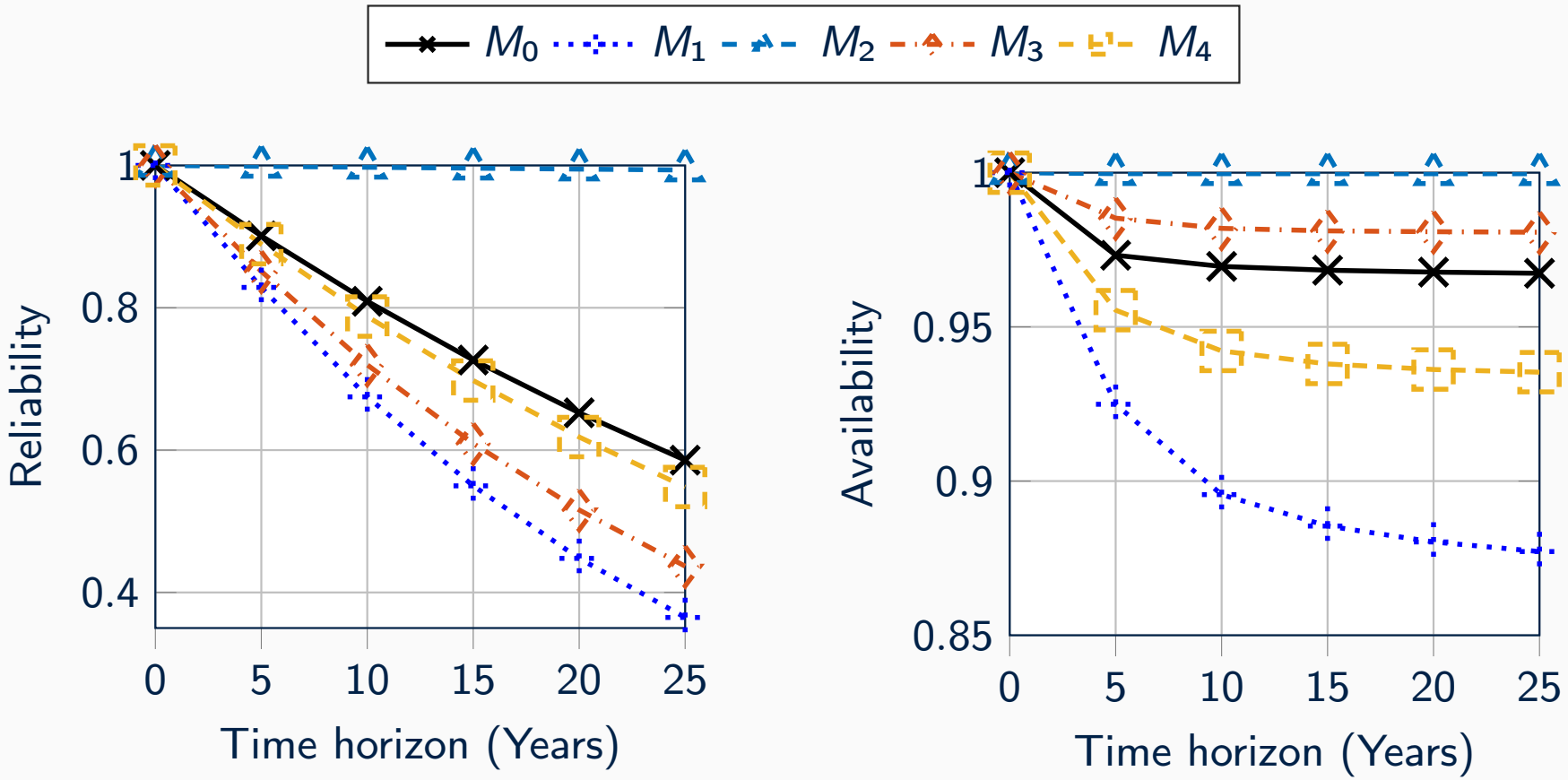
High-level certification: fault maintenance trees

How do we apply framework to BAS?



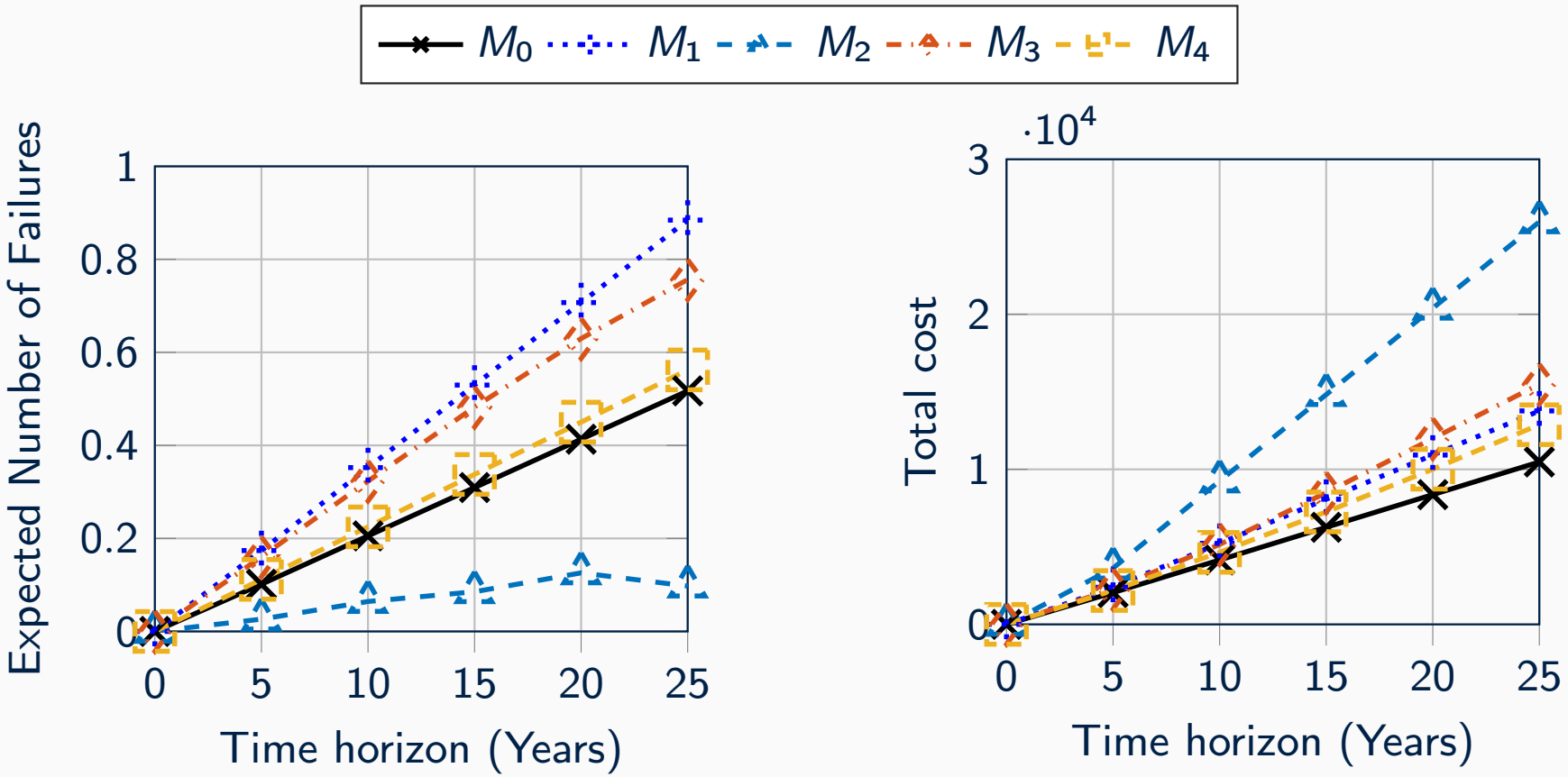
High-level certification: fault maintenance trees

How do we apply framework to BAS?



High-level certification: fault maintenance trees

How do we apply framework to BAS?



High-level certification: fault maintenance trees

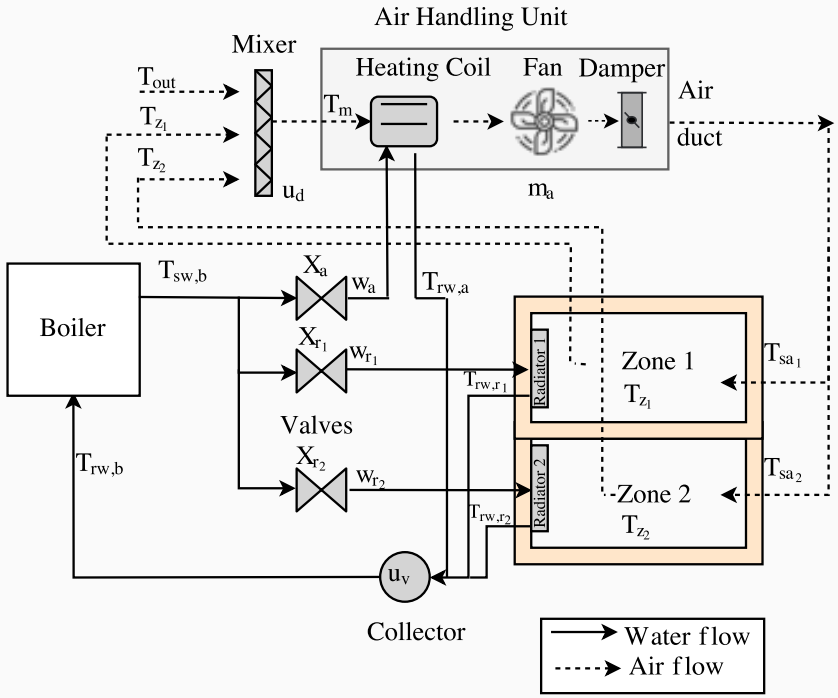
How do we apply framework to BAS?

- compare against statistical model checking

T	Metric	PMC		SMC		Metric	PMC		SMC	
		M_3	M_4	M_3	M_4		M_3	M_4	M_3	M_4
5	**Reliability**	0.6	0.8	6.3	6.1	**Availability**	1.0	1.1	7.9	13.1
10		1.6	1.2	14.4	17.0		2.3	2.1	15.5	25.4
15		2.0	1.7	17.4	29.0		3.2	3.5	24.9	35.8
20		2.3	2.9	19.5	38.6		4.4	4.1	31.2	43.2
25		3.0	3.2	17.9	46.2		5.5	3.4	31.6	52.2
5	**ENF**	1.1	1.2	7.9	12.9	**Total costs**	5.4	5.2	29.8	47.9
10		2.2	2.3	15.4	24.0		10.5	9.4	56.4	90.0
15		3.2	3.5	21.3	34.2		14.0	13.9	78.5	119.4
20		4.9	4.0	27.7	43.3		17.6	17.1	94.7	145.6
25		5.6	4.4	31.6	50.1		22.1	19.7	105.7	168.3

Low-level certification

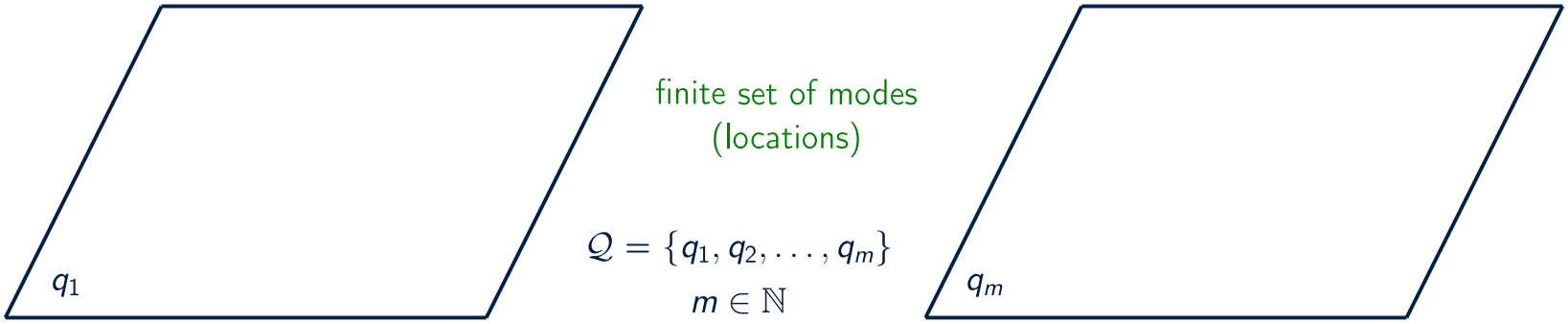
Low-level certification: setup



Low-level certification: stochastic hybrid systems

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

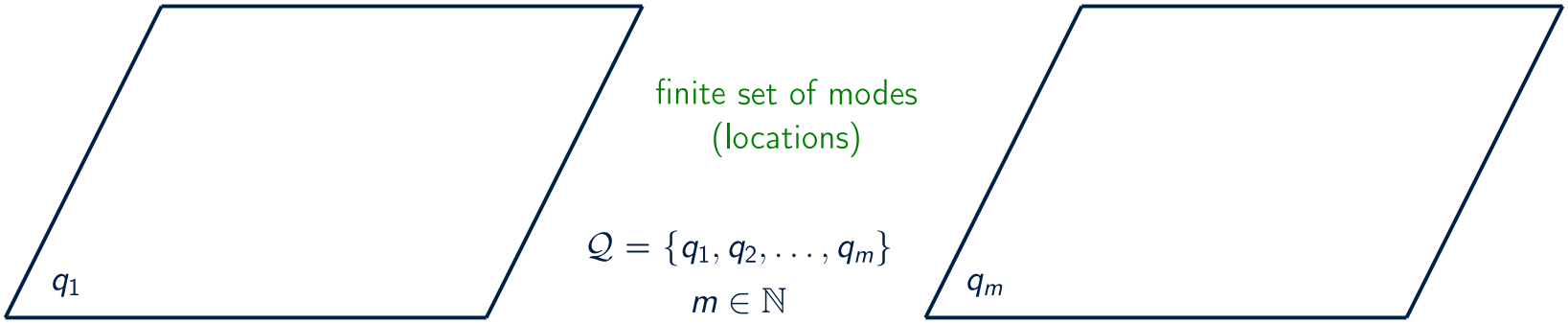
$$\mathcal{H} = (\mathcal{Q}, n, \mathcal{U}, T_q, T_x, \Theta, L)$$



Low-level certification: stochastic hybrid systems

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

$$\mathcal{H} = (\mathcal{Q}, n, \mathcal{U}, T_q, T_x, \Theta, L)$$

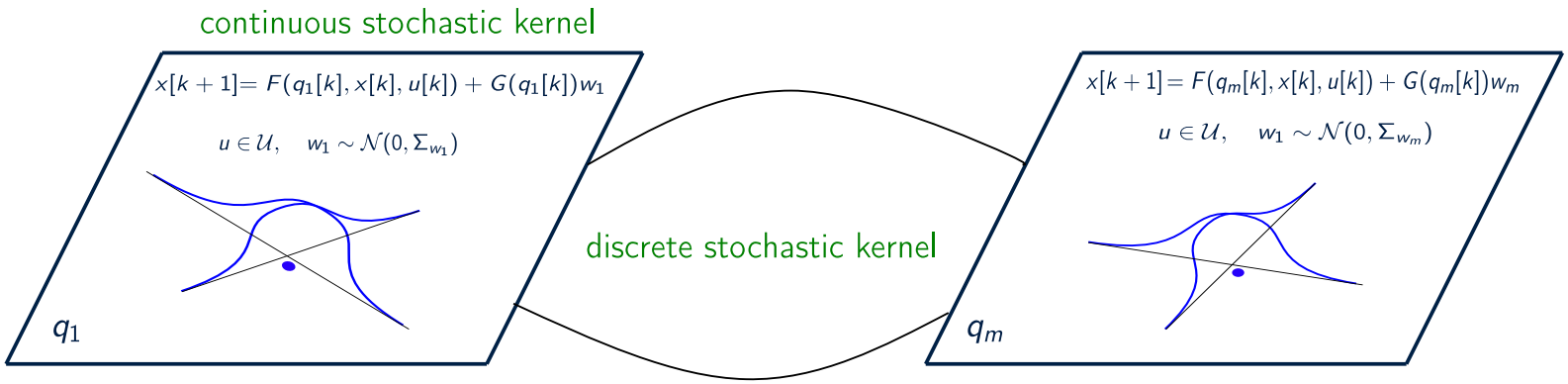


- $n \in \mathbb{N}$: dimension of the continuous space;
- $\mathcal{D} = \cup_{q \in \mathcal{Q}} \{q\} \times \mathbb{R}^n$: hybrid state space
- $\mathcal{U}$: a continuous set of actions

Low-level certification: stochastic hybrid systems

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

$$\mathcal{H} = (\mathcal{Q}, n, \mathcal{U}, T_q, T_x, \Theta, L)$$

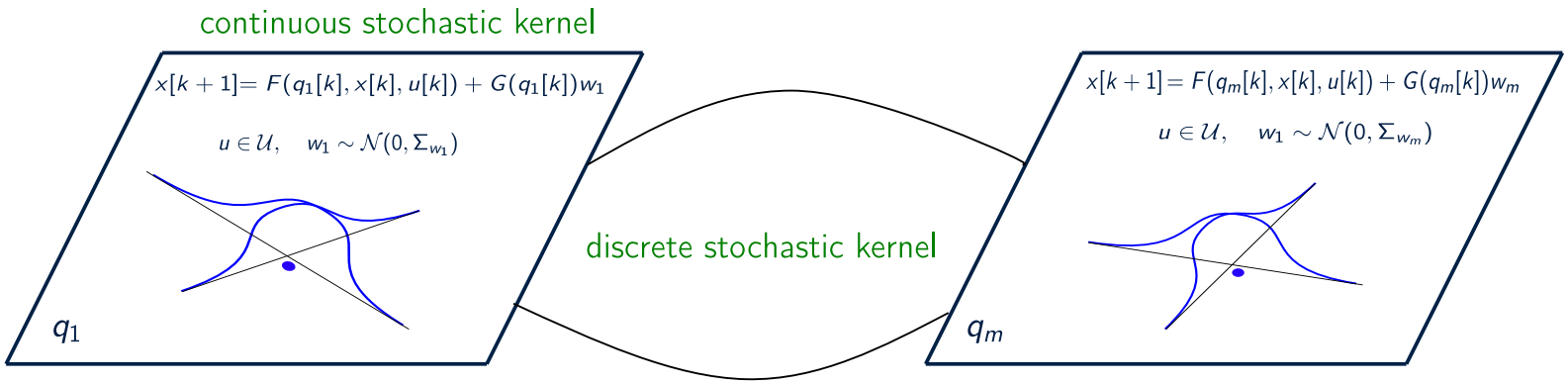


- $\mathcal{Q} : T_q(\cdot|d, u)$: discrete stochastic kernel
- $(\mathbb{R}^n, \mathcal{B}(\mathbb{R}^n)) : T_x(\cdot|d, u)$: continuous stochastic kernel

Low-level certification: stochastic hybrid systems

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

$$\mathcal{H} = (\mathcal{Q}, n, \mathcal{U}, T_q, T_x, \Theta, L)$$

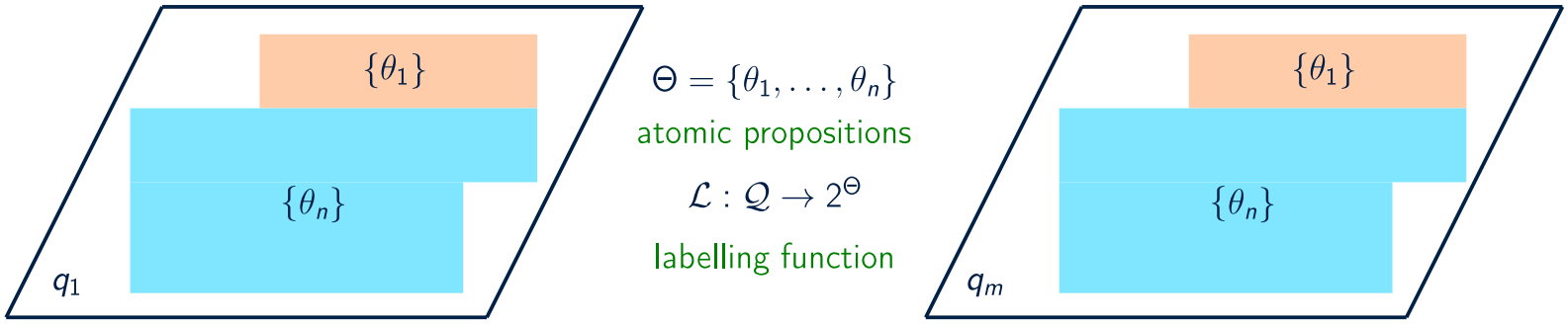


$$x[k+1] = F(q[k], x[k], u[k]) + G(q[k])w,$$
$$u \in \mathcal{U}, \quad w \sim \mathcal{N}(0, \Sigma_w),$$

Low-level certification: stochastic hybrid systems

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

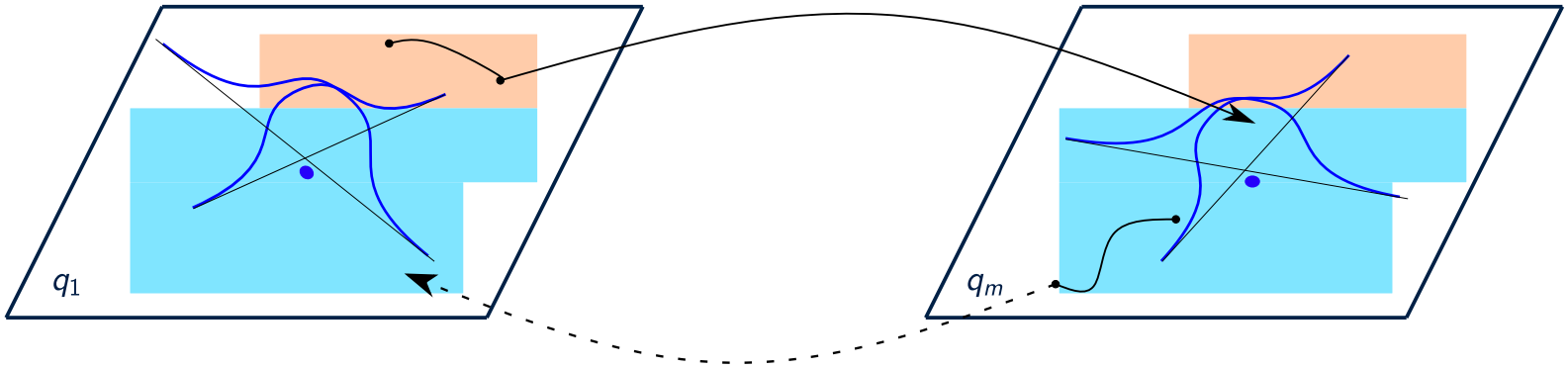
$$\mathcal{H} = (\mathcal{Q}, n, \mathcal{U}, T_q, T_x, \Theta, L)$$



Low-level certification: stochastic hybrid systems

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

$$\mathcal{H} = (\mathcal{Q}, n, \mathcal{U}, T_q, T_x, \Theta, L)$$



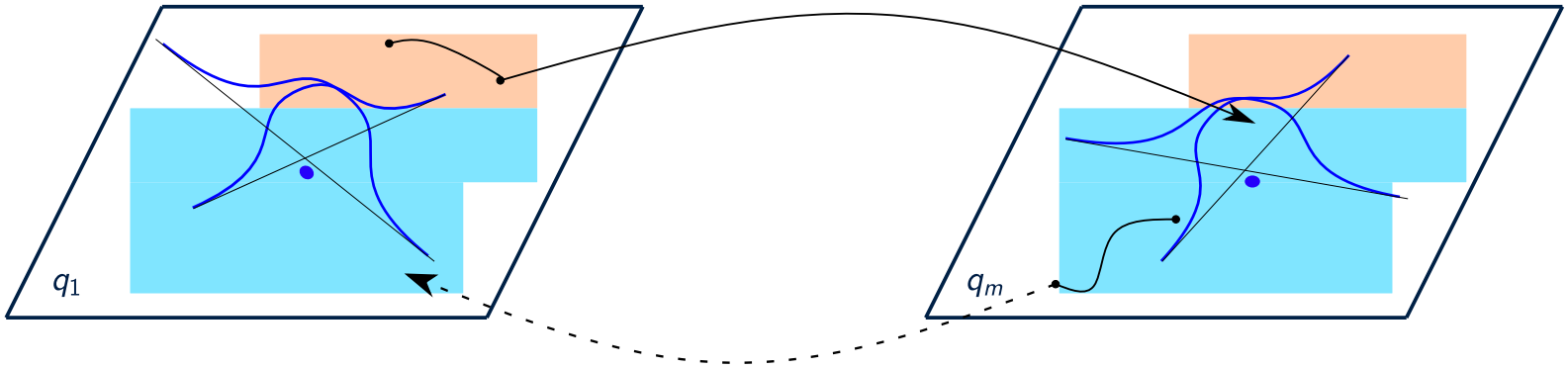
Stochastic process

- a hybrid state of $\mathcal{H}$ is a pair $d = (q, x) \in \mathcal{D}$
- evolution of $\mathcal{H}$ for $k \in \mathbb{N}$ is a stochastic process $\mathbf{d}[k] = (q[k], x[k]) \in \mathcal{H}$

Low-level certification: stochastic hybrid systems

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

$$\mathcal{H} = (\mathcal{Q}, n, \mathcal{U}, T_q, T_x, \Theta, L)$$



Stochastic process

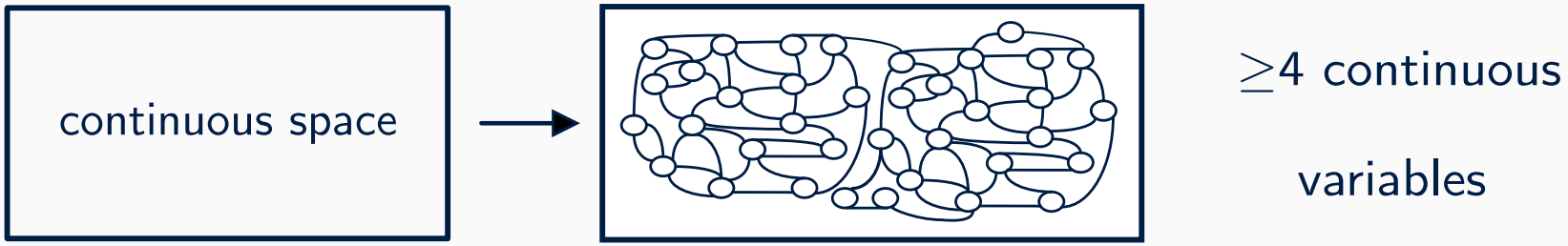
- SHS evolve over uncountable space
  - need to perform abstractions
- quantitative and finite abstractions into Markov processes

Low-level certification: MDP abstractions

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

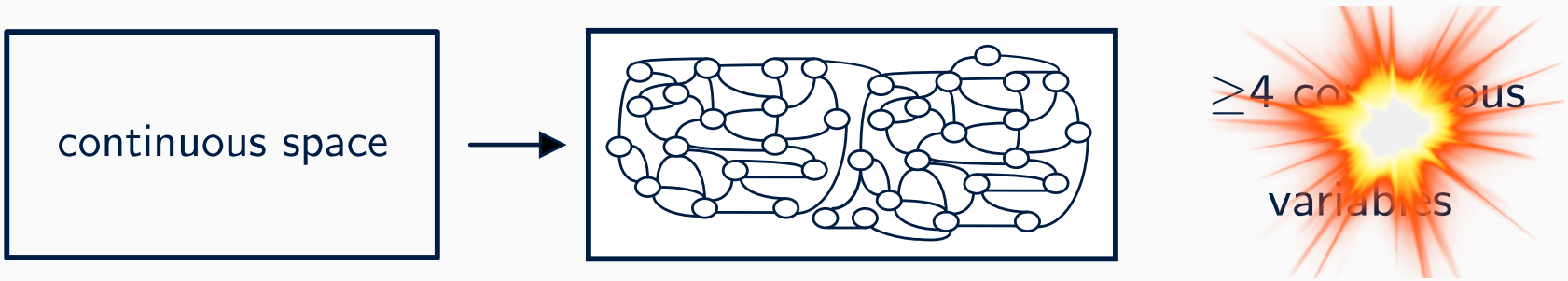
Low-level certification: MDP abstractions

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



Low-level certification: MDP abstractions

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

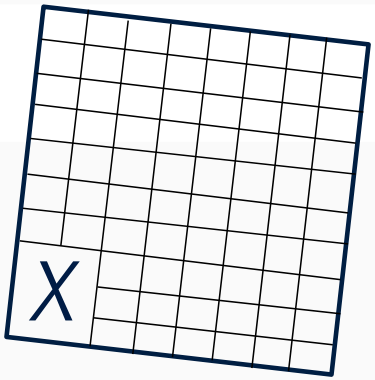


Low-level certification: IMDP abstractions

How do we embed uncertainty in abstraction?

Low-level certification: IMDP abstractions

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



Low-level certification: IMDP abstractions

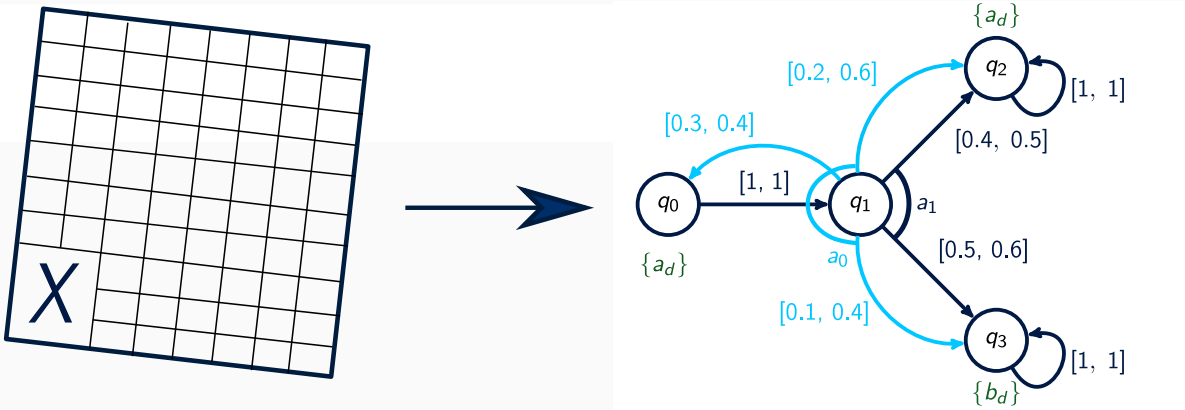
We construct abstract model that captures all behaviours of SHS with respect to X and regions of interest

1. discretise set X according to dynamics of each mode

Low-level certification: IMDP abstractions

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

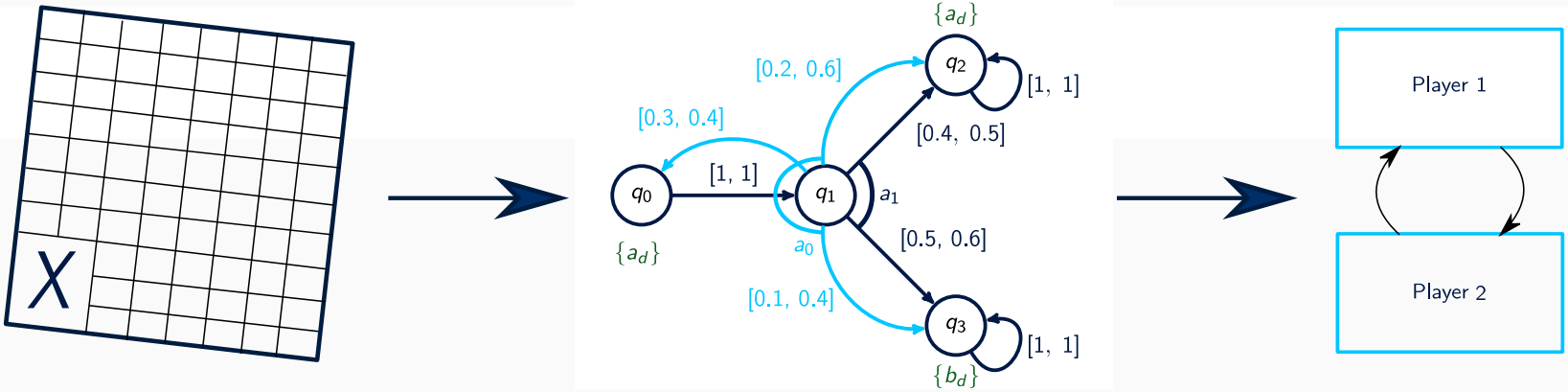
1. discretise set X according to dynamics of each mode
2. quantify error of abstraction and represent it as uncertainty in the abstraction



Low-level certification: IMDP abstractions

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

1. discretise set X according to dynamics of each mode
2. quantify error of abstraction and represent it as uncertainty
3. perform verification or synthesis via stochastic games
4. refine back to SHS



Low-level certification: comparison

Model with varying dimensions

- 1 discrete mode, d continuous variables, no control action
- continuous variable evolve according to

$$X[k+1] = 0.8\mathbf{1}_d X[k] + 0.2\mathbf{1}_d W[k]$$

Low-level certification: comparison

Model with varying dimensions

- 1 discrete mode, d continuous variables, no control action
- continuous variable evolve according to

$$X[k+1] = 0.8\mathbf{1}_d X[k] + 0.2\mathbf{1}_d W[k]$$

Model checking for different d dimensions

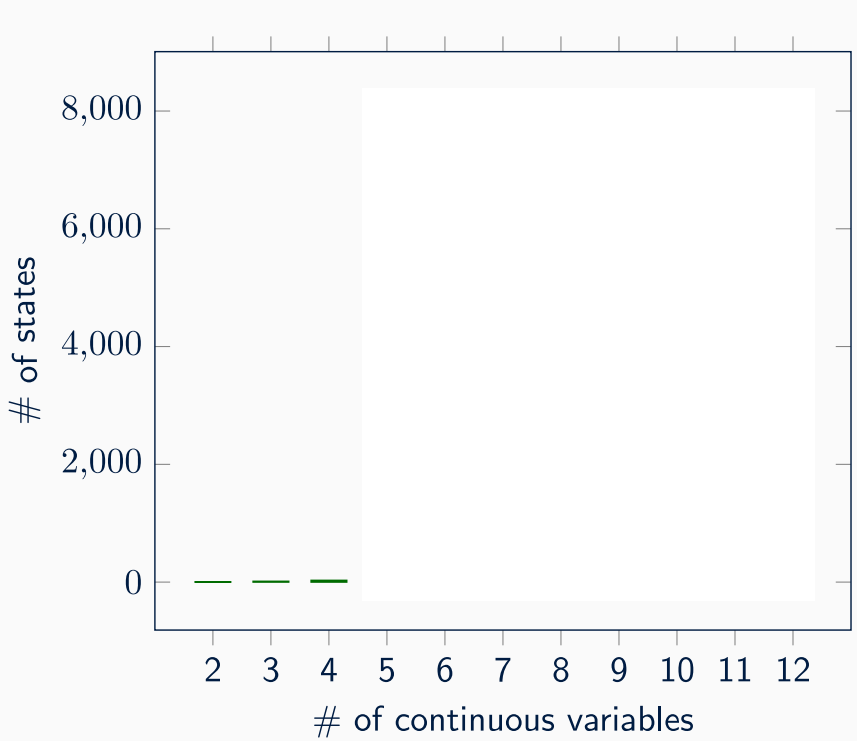
- safety requirement

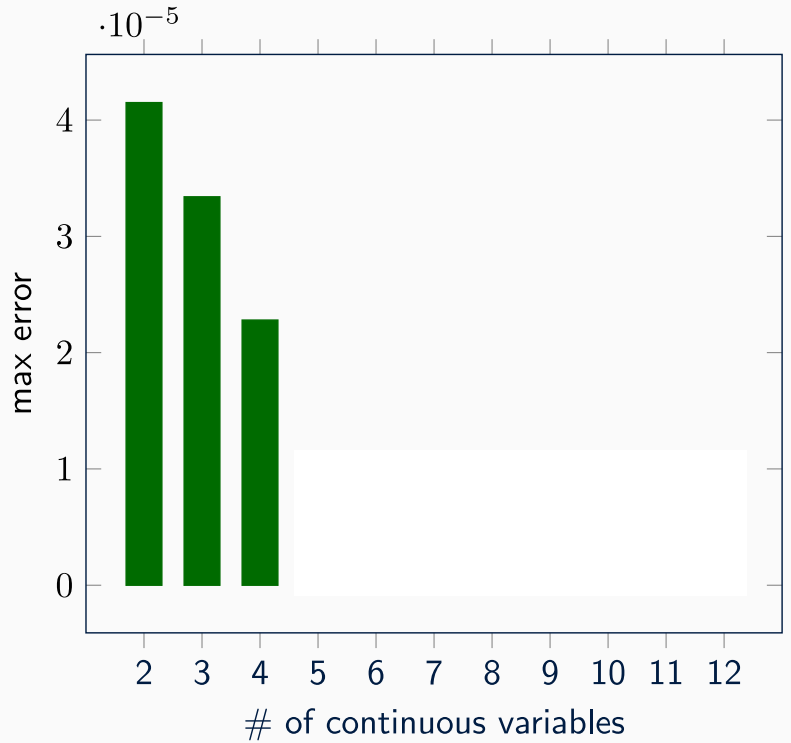
$$\varphi_1 := \mathrm{P}_{=?}\left(\mathbf{G}^{\leq K=10}\, [-1,1]^d\right)$$

Low-level certification: comparison

Results:

- abstract models with manageable state spaces
- scalability with respect to the continuous dimension d
- marked improvement over state-of-the-art tools

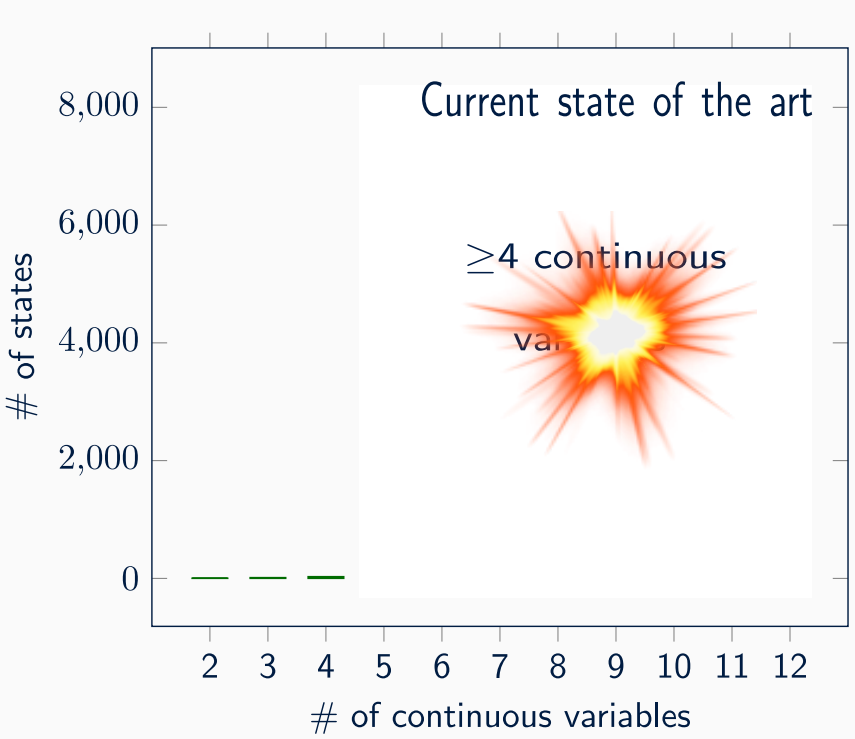


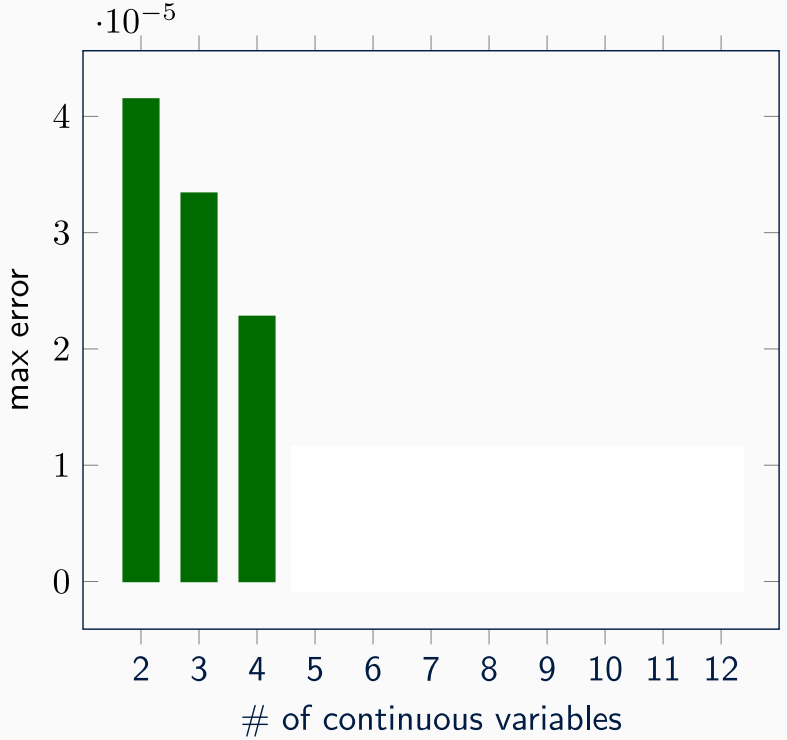


Low-level certification: comparison

Results:

- abstract models with manageable state spaces
- scalability with respect to the continuous dimension d
- marked improvement over state-of-the-art tools

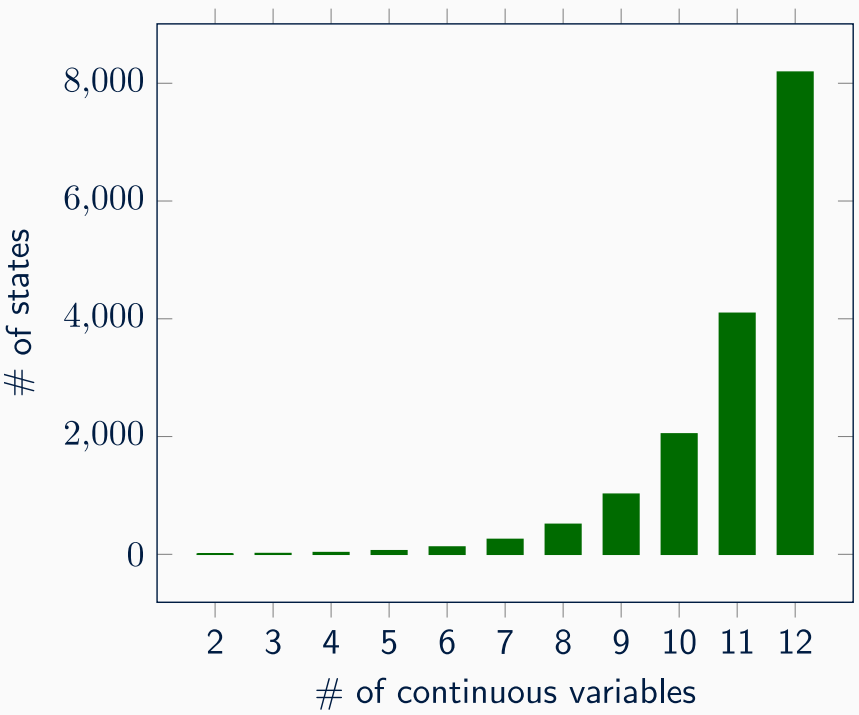


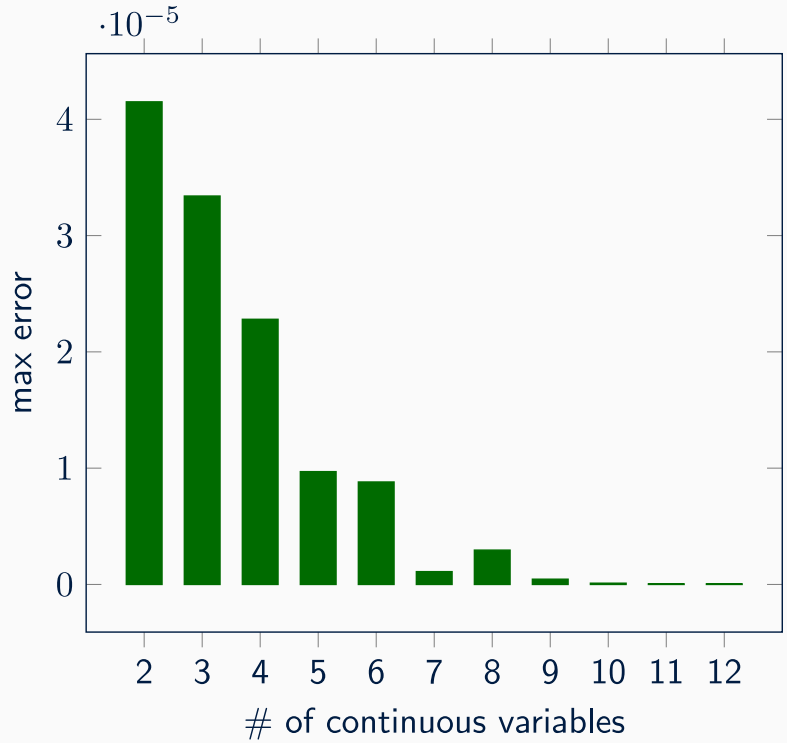


Low-level certification: comparison

Results:

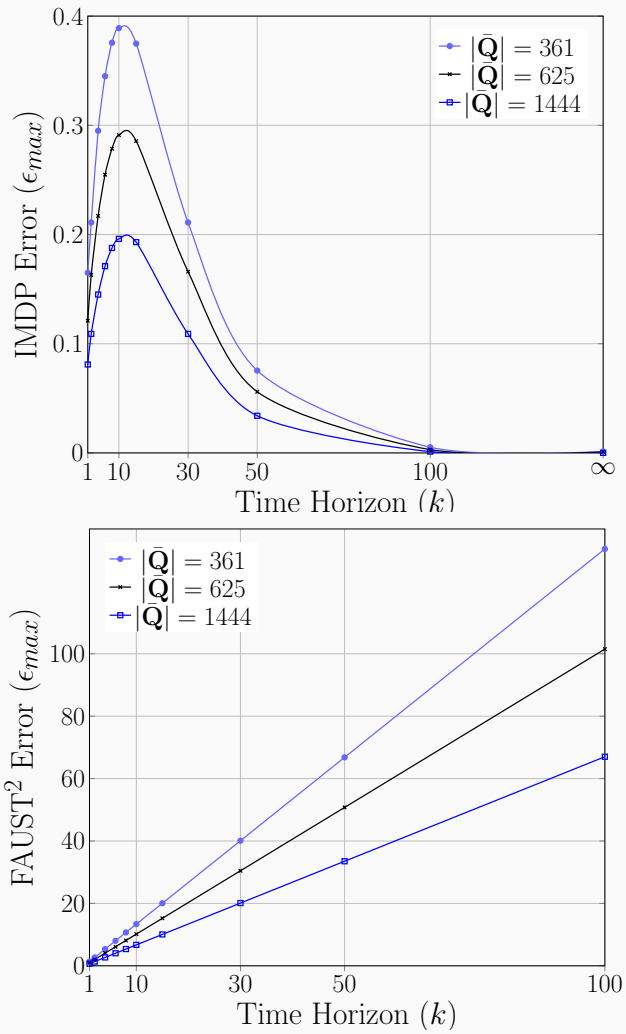
- abstract models with manageable state spaces
- scalability with respect to the continuous dimension d
- marked improvement over state-of-the-art tools





Low-level certification: comparison

Error analysis: $\varphi_1 := \mathbf{G}^{\leq K=10}[-1,1]^{d=2}$



- error embedded within abstraction
- performs computations according to feasible transition probabilities
- abstraction errors does not explode with time
- significant reduction in computational time and effort

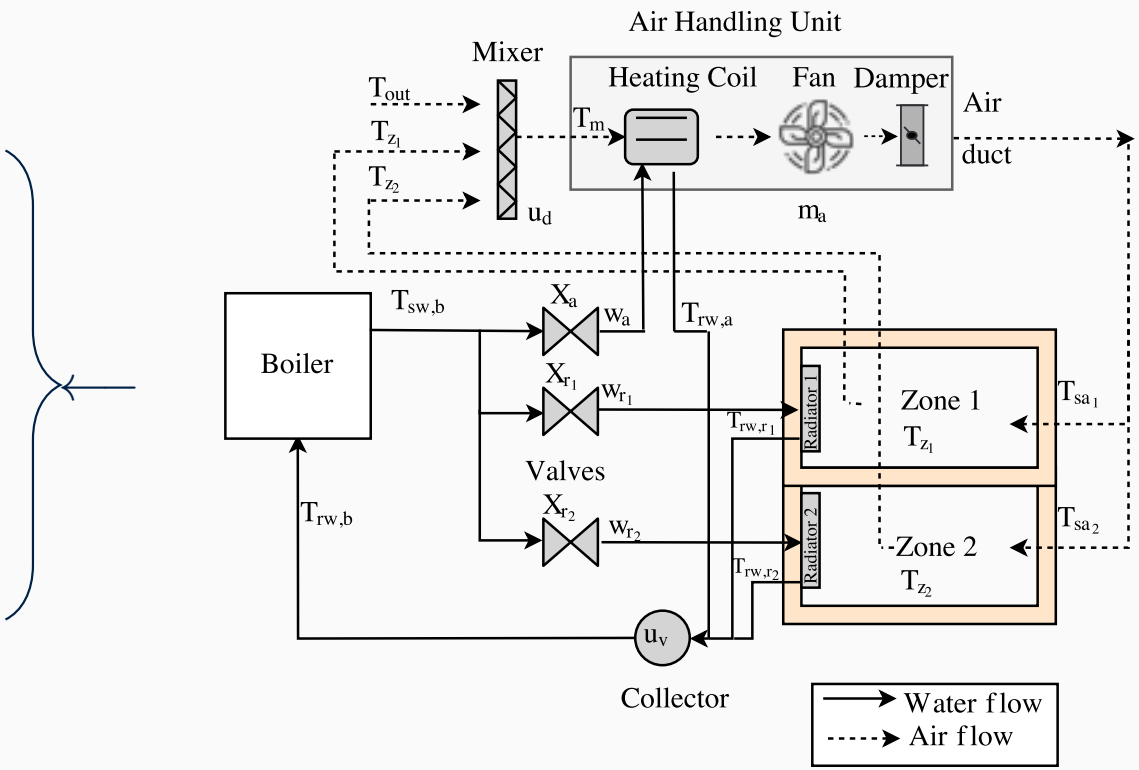
Low-level certification: library of models

Components

- boiler
- valve
- heating coil
- mixer

...

- zone



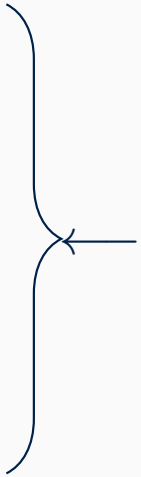
Low-level certification: library of models

Components

- boiler
- valve
- heating coil
- mixer

...

- zone



Model description

- 1 discrete variable: B_{en}
- 1 continuous variable: $T_{sw,b}$
- 1 output: $T_{sw,b}$
- process noise: σ_{sw}
- constants: τ_{sw}, k_b

Model dynamics

$$dT_{sw,b}(t) = \begin{cases} 0 & B_{en}(t) = 0 \\ (\tau_{sw})^{-1}[(-T_{sw,b}(t) + k_b)dt] + \sigma_{sw}dW & B_{en}(t) = 1 \end{cases}$$

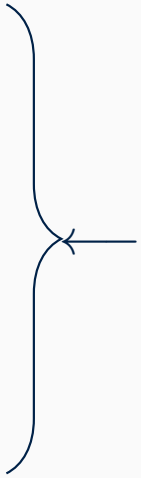
Low-level certification: library of models

Components

- boiler
- valve
- heating coil
- mixer

...

- zone



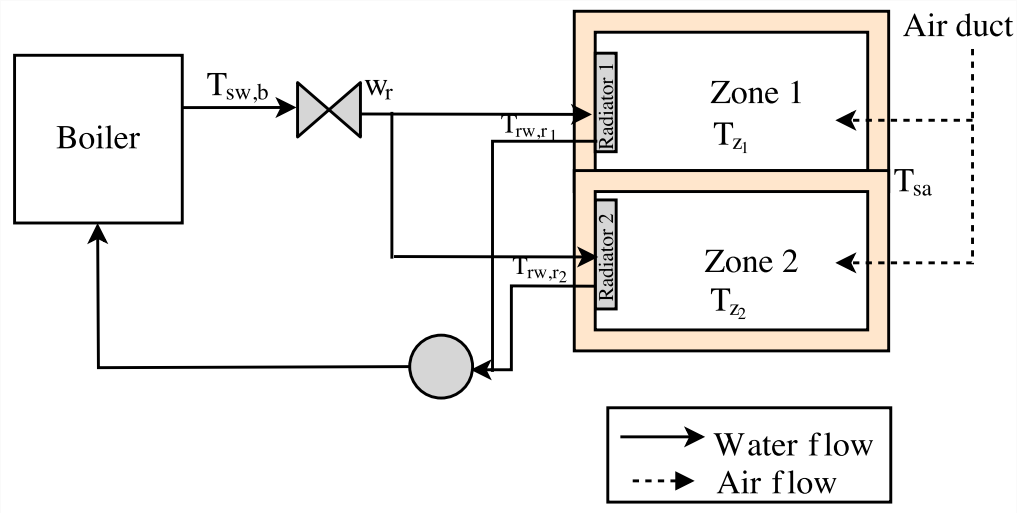
Types

- algebraic / differential
- non / linear
- continuous variables
- discrete modes
- noise

Coupling

- input-output
- control laws

Low-level certification: formal verification case study



Assuming slow air movement (less than $12.5 m/min$) and 50% indoor relative humidity, the operative temperatures recommended range from 20 $^{\circ}C$ and 24 $^{\circ}C$ in the winter

Low-level certification: formal verification case study

- 1 discrete mode, 4 continuous variables, 1 control action

$$X[k+1] = AX[k] + BU[k] + Q + \Sigma W[k]$$

$$Y[k] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} X[k].$$

Low-level certification: formal verification case study

- 1 discrete mode, 4 continuous variables, 1 control action

$$X[k+1] = AX[k] + BU[k] + Q + \Sigma W[k]$$
$$Y[k] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} X[k].$$

- safety requirement

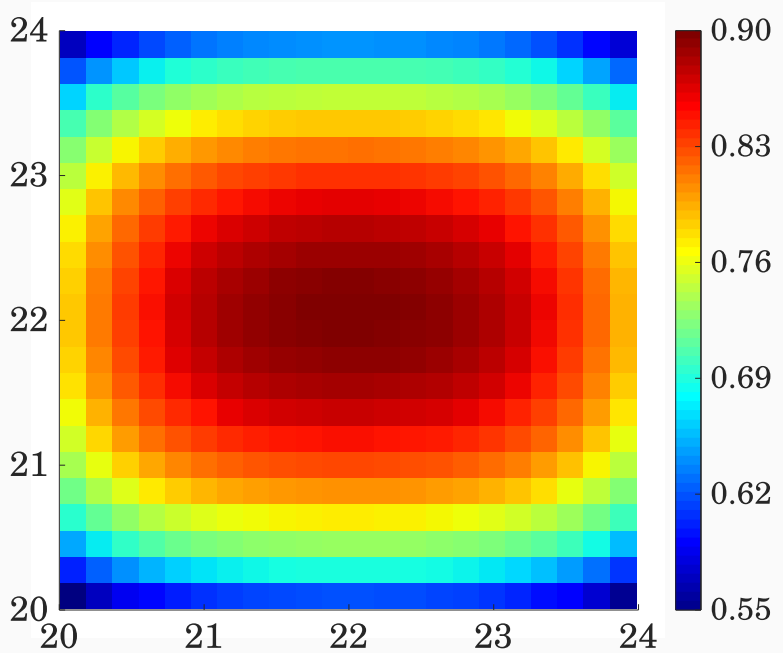
$$\varphi_2 := \mathrm{P}_{=?}\left(\mathbf{G}^{\leq K=6}\ X_{safe}\right)$$

- $K = 6 \times \Delta = 1.5$ hours

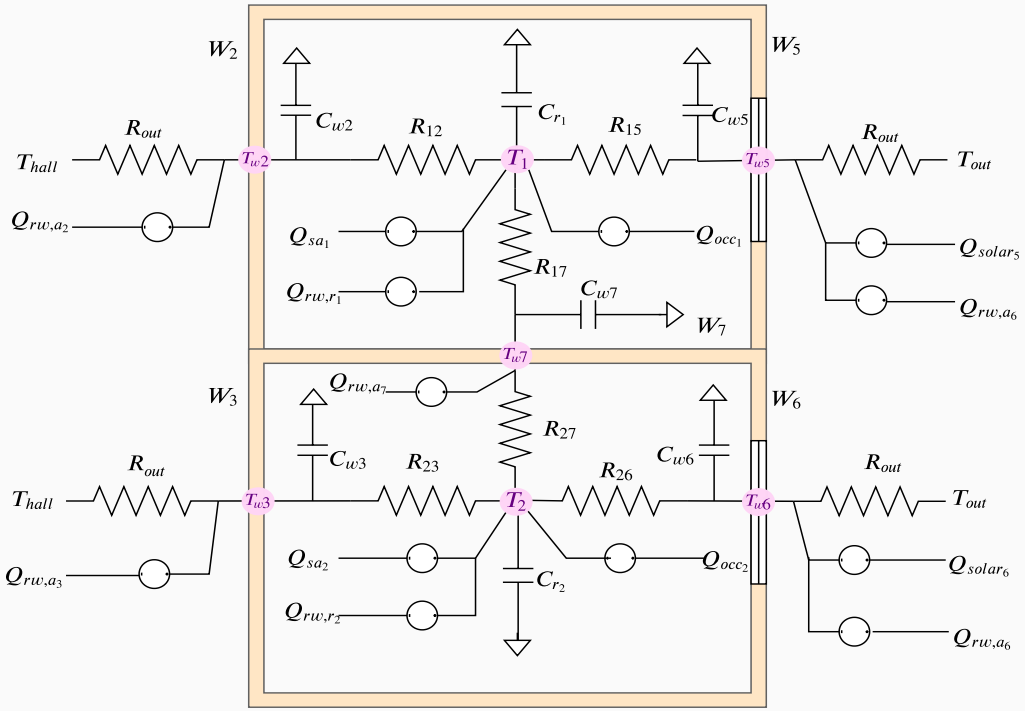
Low-level certification: formal verification case study

Solving using IMDP abstractions

	$\lvert\mathcal{S}\rvert$ [states]	**Time taken** [s]	$\mathbf{p}_{\varphi_2}(\mathbf{s})$	**Error** ε_{max}
2D	484	56.9	≥ 0.55	0.16
4D	2132	1587.7	≥ 0.55	0.18



Low-level certification: synthesis case study



Temperature within zone 1 does not deviate from the setpoint ($T_{sp} = 22\ [\,^\circ C]$) by more than 1 $[\,^\circ C]$ over a time horizon equal to 2.25 hours (i.e $K = 9$).

Low-level certification: synthesis case study

- 1 discrete mode, 7 continuous variables, 1 control action

$$X[k+1] = AX[k] + BU[k] + Q + \Sigma W[k]$$
$$Y[k] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} X[k].$$

Low-level certification: synthesis case study

- 1 discrete mode, 7 continuous variables, 1 control action

$$X[k+1] = AX[k] + BU[k] + Q + \Sigma W[k]$$
$$Y[k] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} X[k].$$

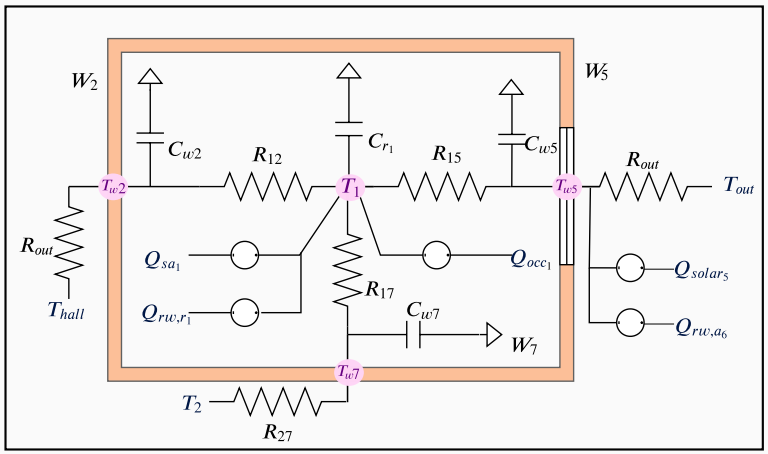
- synthesis requirement

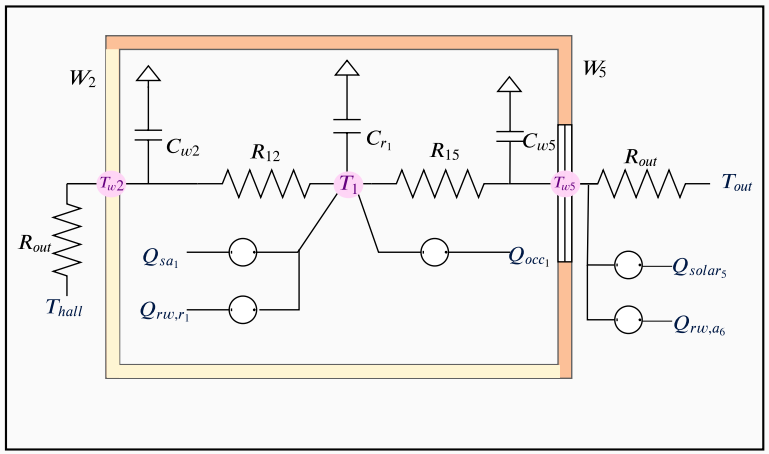
$$\varphi_3 := \mathrm{P}_{\geq p}\left(\mathbf{G}^{\leq K=9}\ [|T_{sp} - T_{z_1}| \leq 1]\right)$$

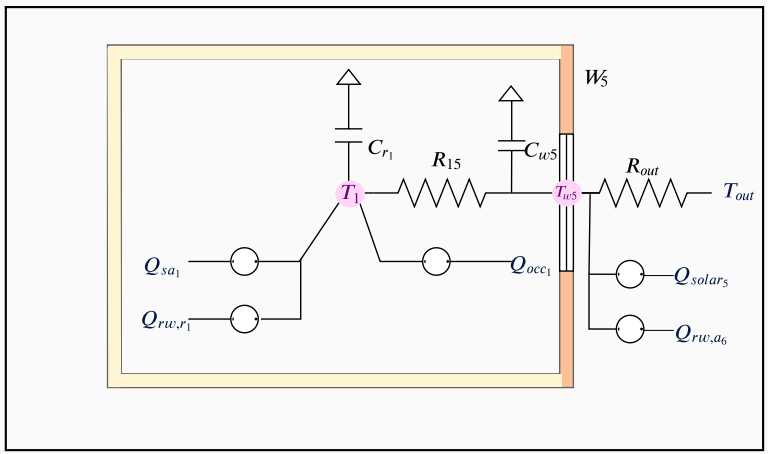
- $K = 9 \times \Delta = 2.25$ hours

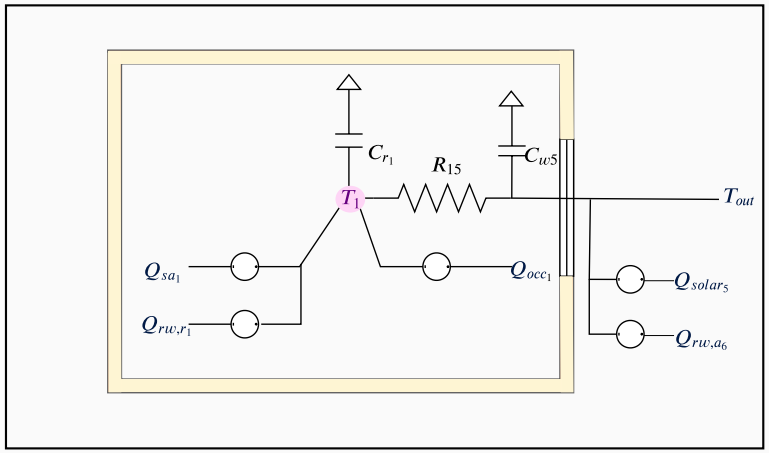
Low-level certification: synthesis case study

Model order reduction



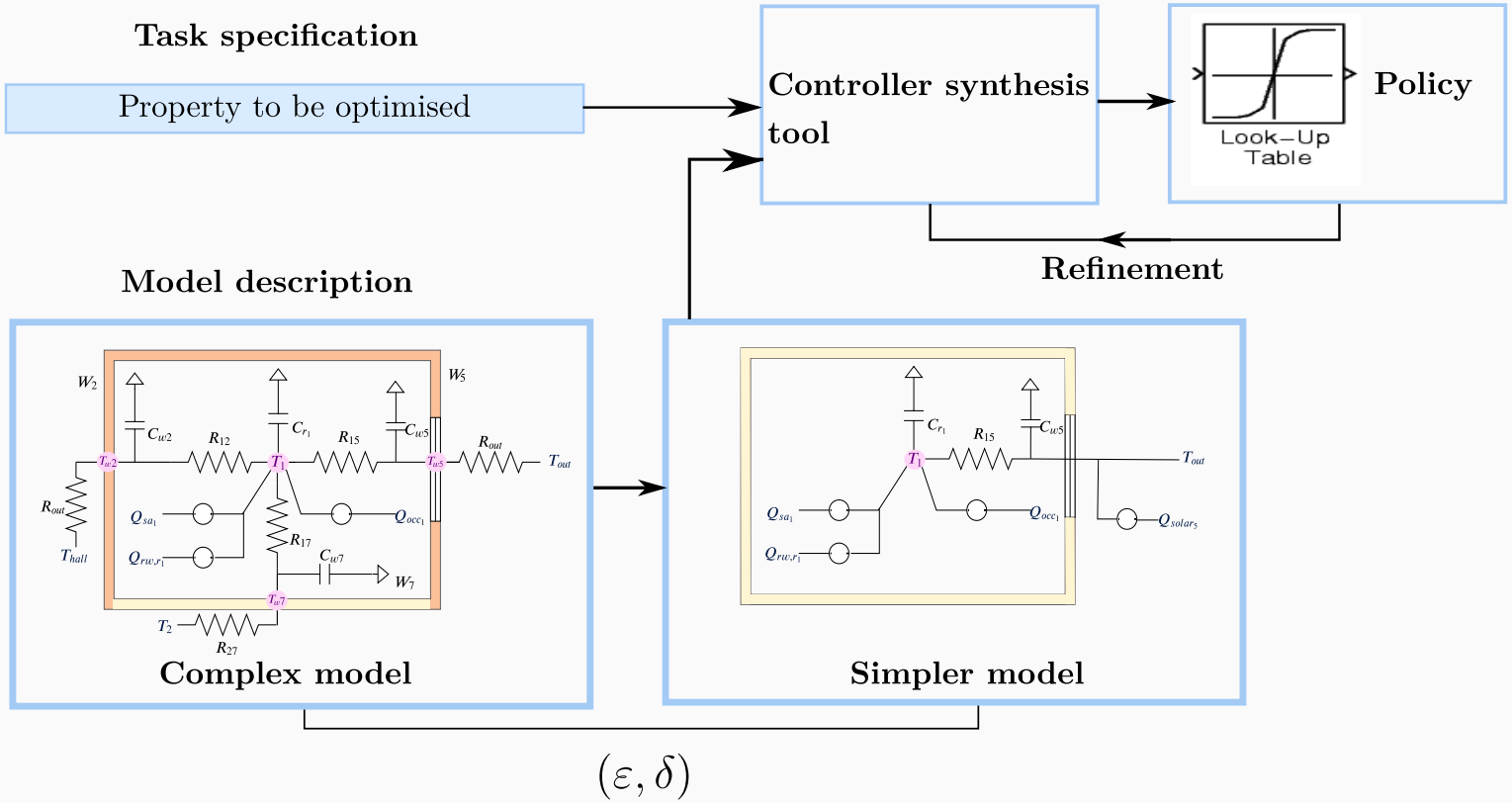






Low-level certification: synthesis case study

Employ synthesis via (ϵ, δ) relations



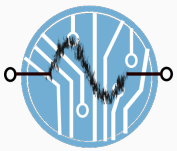
resulting policy: safety probability of $p = 0.7657$

Software tool

StocHy: **related work**

Tool	continuous variables	discrete variables	time	scope
The Modest Toolset	deterministic	multiple	continuous	modelling & verification & JANI support
Storm	-	multiple	continuous or discrete	probabilistic model checking
	-	multiple	continuous or discrete	probabilistic model checking
FAUST^2	stochastic	single	discrete	probabilistic reachability analysis

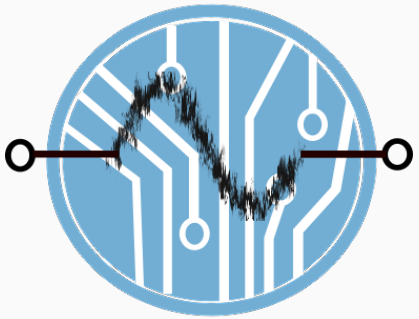
StocHy: **related work**

Tool	continuous variables	discrete variables	time	scope
The Modest Toolset	deterministic	multiple	continuous	modelling & verification & JANI support
Storm	-	multiple	continuous or discrete	probabilistic model checking
	-	multiple	continuous or discrete	probabilistic model checking
FAUST^2	stochastic	single	discrete	probabilistic reachability analysis
 StocHy	stochastic	multiple	discrete	probabilistic reachability, simulation

StocHy: **contribution**

verification

- abstraction based
- novel algorithm with tighter bounds and more scalability



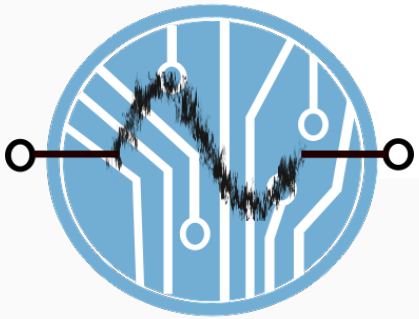
StocHy: **contribution**

verification

- abstraction based
- novel algorithm with tighter bounds and more scalability

synthesis

- abstraction based
- optimisation via sparse matrices



StocHy

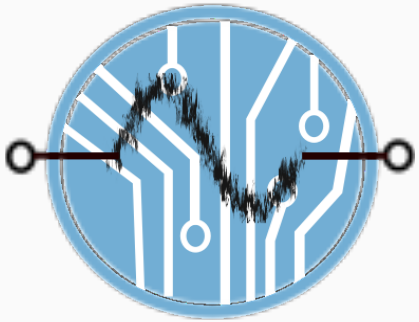
StocHy: **contribution**

verification

- abstraction based
- novel algorithm with tighter bounds and more scalability

simulation

- automatically generates statistics
- visualisation via time varying histograms



synthesis

- abstraction based
- optimisation via sparse matrices

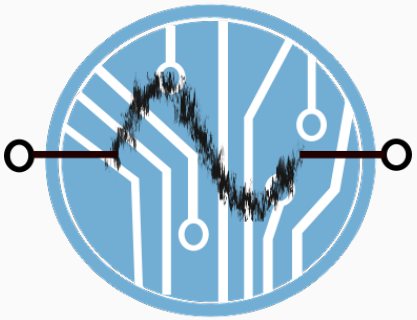
StocHy: **contribution**

verification

- abstraction based
- novel algorithm with tighter bounds and more scalability

simulation

- automatically generates statistics
- visualisation via time varying histograms



StocHy

synthesis

- abstraction based
- optimisation via sparse matrices

features

- modular
- C++ implementation
- extendable
- multiple options

- 1 discrete mode, 2 continuous variables, no control action
- continuous variable evolve according to

$$x_1[k+1] = x_1[k] + \frac{\Delta}{V}(-\rho_m x_1[k] + \rho_c(C_{out} - x_1[k])) + \sigma_1 w_1[k]$$

$$x_2[k+1] = x_2[k] + \frac{\Delta}{C_z}(\rho_m C_{pa}(T_{set} - x_2[k]) + \frac{\rho_c}{R}(T_{out} - x_2[k])) + \sigma_2 w_2[k]$$

x_1: zone CO_2 level

x_2 : zone temperature level

$\sigma_{(\cdot)}$: variance of noise $w_{\cdot,k} \sim \mathcal{N}(0,1)$

$\rho_{(\cdot)}$: fixed air flow

C_{out} :outside CO_2 level

T_{set}: set temperature

T_{out}: outside temperature

StocHy: **formal verification**

- 1 discrete mode, 2 continuous variables, no control action
- continuous variable evolve according to

$$x_1[k+1] = x_1[k] + \frac{\Delta}{V}(-\rho_m x_1[k] + \rho_c(C_{out} - x_1[k])) + \sigma_1 w_1[k]$$

$$x_2[k+1] = x_2[k] + \frac{\Delta}{C_z}(\rho_m C_{pa}(T_{set} - x_2[k]) + \frac{\rho_c}{R}(T_{out} - x_2[k])) + \sigma_2 w_2[k]$$

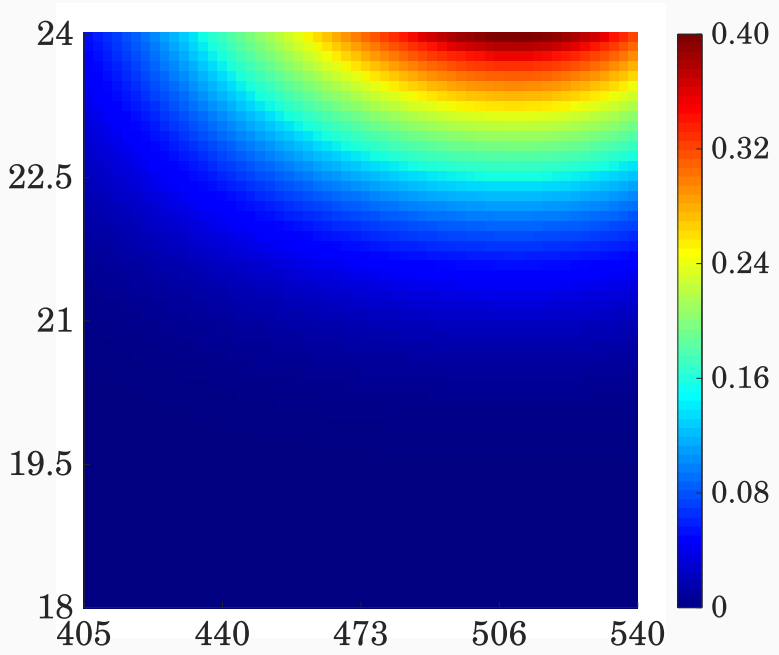
- check if CO_2 and temperature levels remain within

$$\varphi_4 := \mathrm{P}_{=?}\left(\mathcal{G}^{\leq K}\left([405, 540] \times [18, 24]\right)\right)$$

- verification task via IMDP library

Results:

- maximal safety probability: 0.5
- total number of states: 3481
- computational time: 946.29 [s]



probability of satisfiability

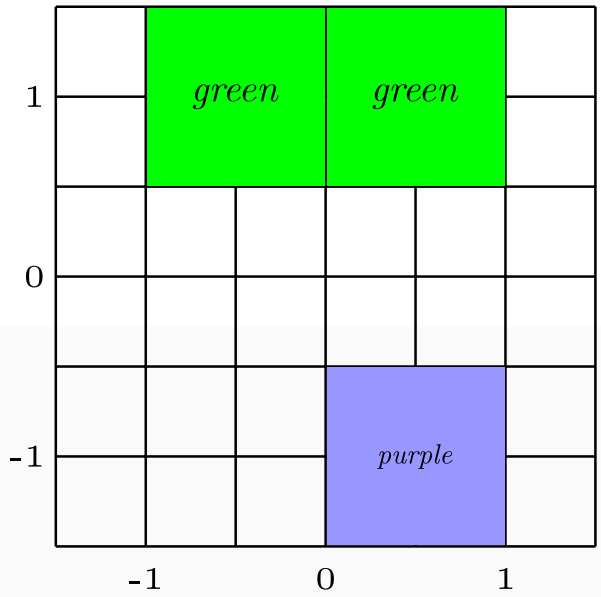
StocHy: **strategy synthesis**

- 2 discrete modes: $\mathcal{Q} = \{q_0, q_1\}$
- 2 continuous variables evolve according to:

$$X[k+1] = A_{q_i} X[k+1] + \Sigma_{q_i} W[k]$$

$$A_{q_0} = \begin{bmatrix} 0.43 & 0.52 \\ 0.65 & 0.12 \end{bmatrix} \Sigma_{q_0} = \begin{bmatrix} 1 & 0.1 \\ 0 & 0.1 \end{bmatrix}$$

$$A_{q_1} = \begin{bmatrix} 0.65 & 0.12 \\ 0.52 & 0.43 \end{bmatrix} \Sigma_{q_1} = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}$$



StocHy: **strategy synthesis**

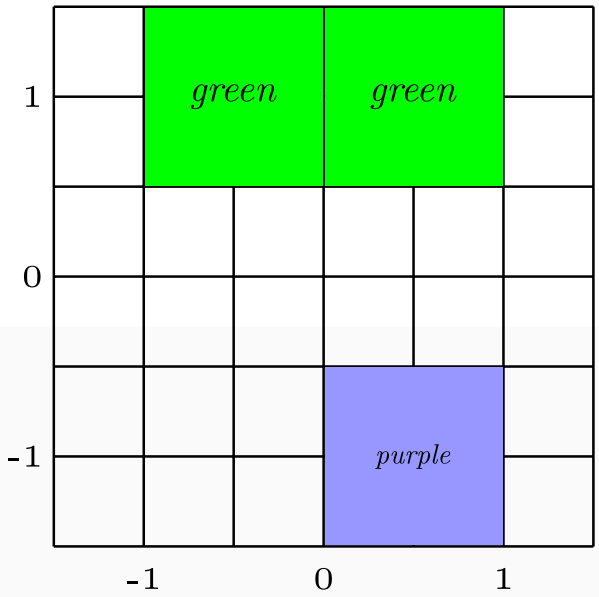
- 2 discrete modes: $\mathcal{Q} = \{q_0, q_1\}$
- 2 continuous variables evolve according to:

$$X[k+1] = A_{q_i} X[k+1] + \Sigma_{q_i} W[k]$$

- optimal strategy to satisfy

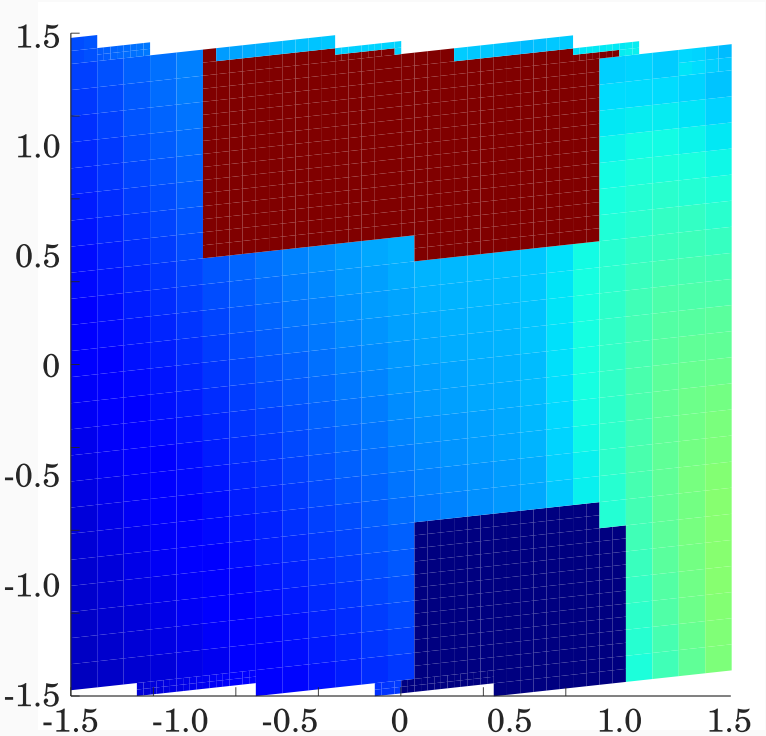
$$\varphi_5 := \mathrm{P}_{\geq p}\left[(\neg \mathit{purple})\ \mathrm{U}\ \mathit{green}\right]$$

- synthesis task via IMDP library

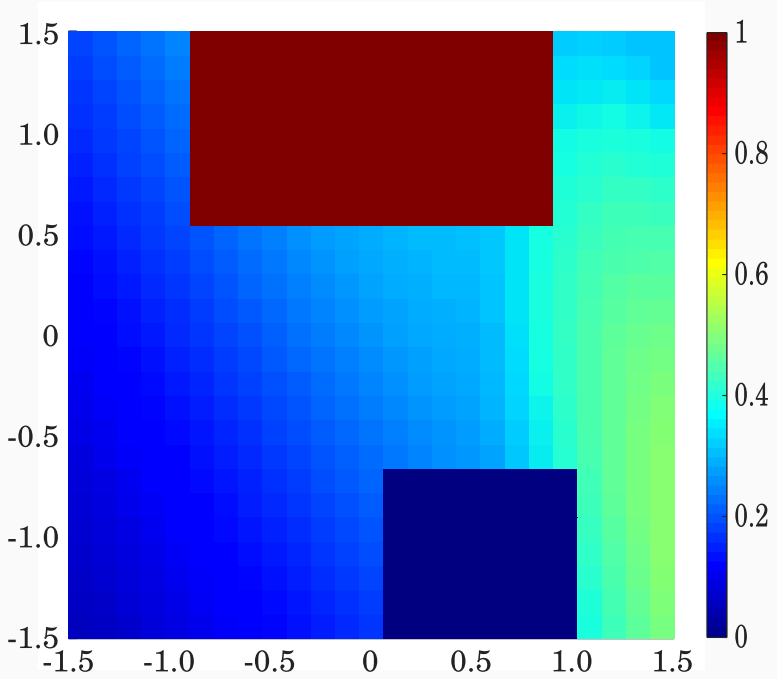


Results:

- maximal abstraction error: 0.21
- total number of states: 2410
- computational time: 1639.30 [s]



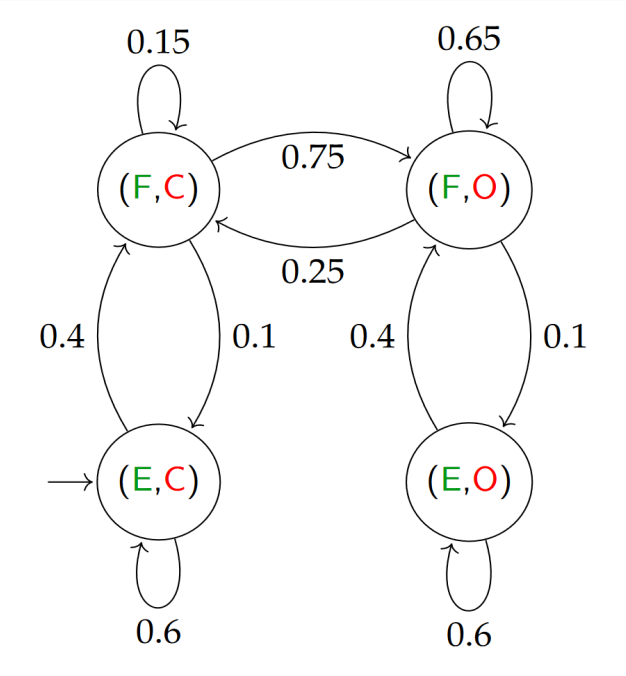
lower bound probability for q_0

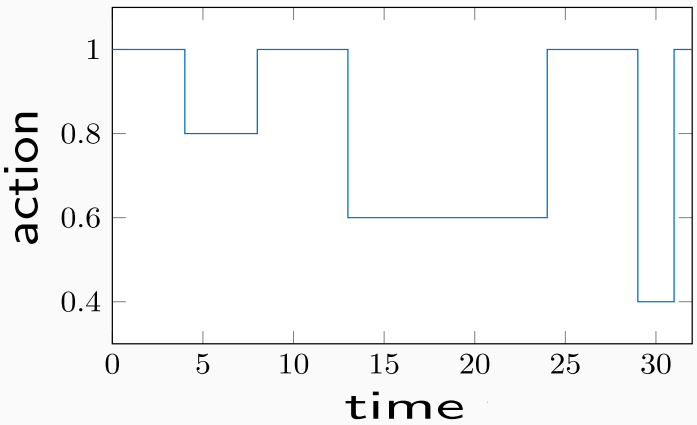


lower bound probability for q_1

- 4 discrete modes, 2 continuous variables

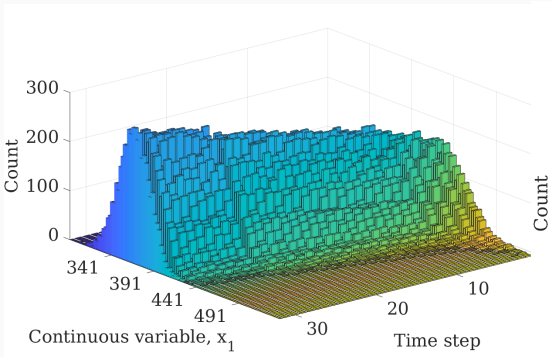
$$X[k+1] = A_{q_i} X[k] + B_{q_i} U[k] + X[k] \sum_{i=1}^{v} N_{q,i} U_{k,i} + Q_q + \Sigma_{q_i} W[k]$$

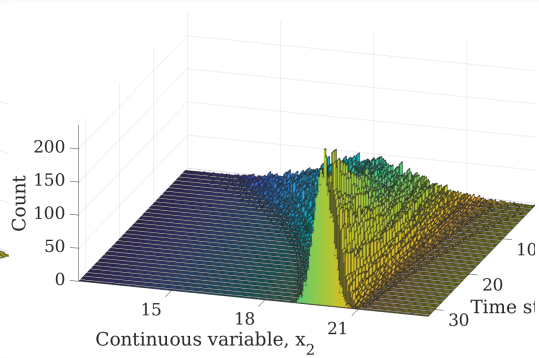


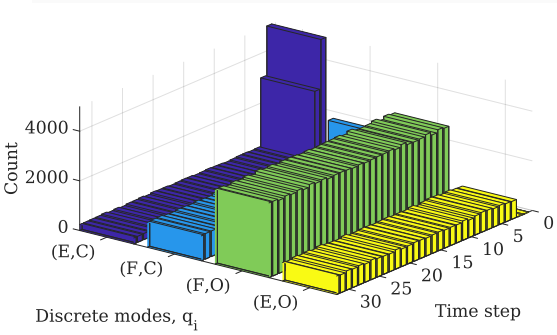


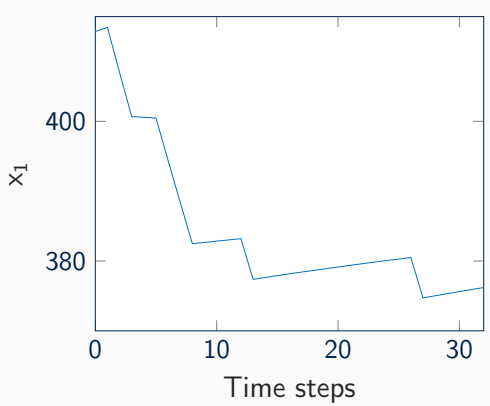
Results:

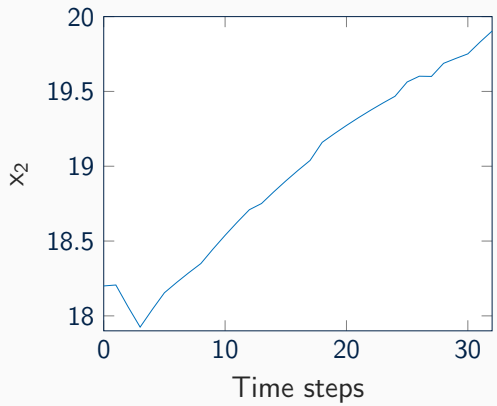
- computational time: 48.6 [s]

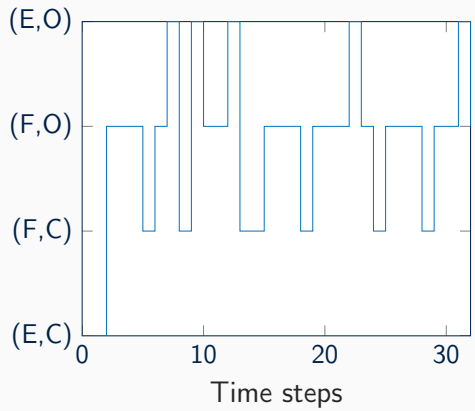








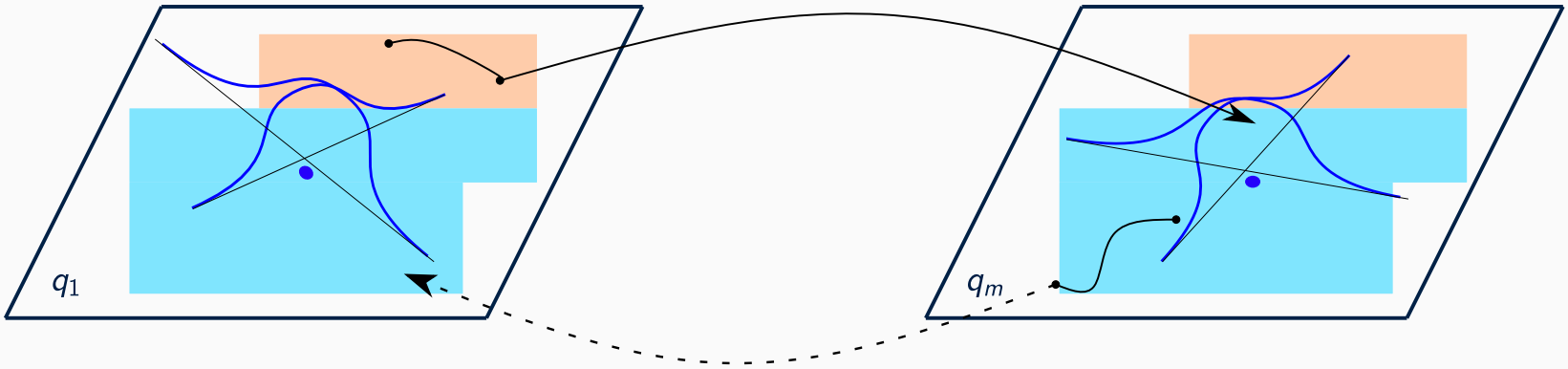




Conclusions: highlights

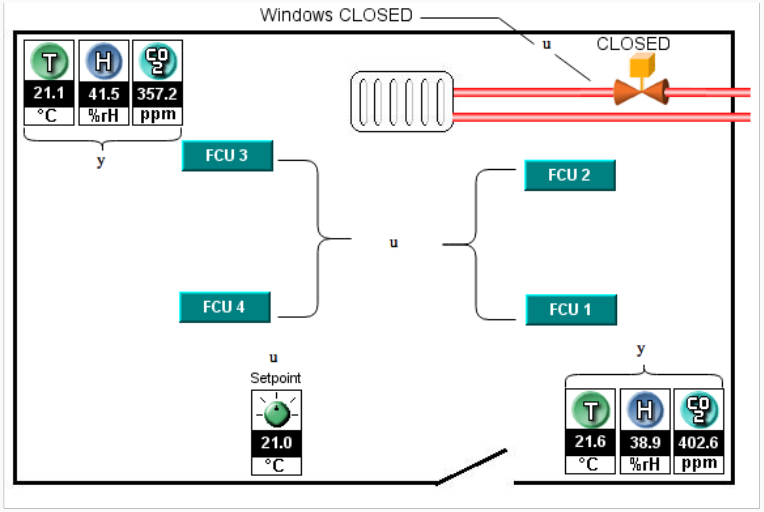
- integrated approach for certification of systems
- framework for computing guarantees and performance levels for BAS
- *StocHy*: aims to simplify complexity of modelling using SHS
- library of models for BAS: serve as benchmarks
- application of control policy refinement
- analysis of fault maintenance trees using probabilistic model checking

Recommendations for future research



- analysis of continuous-time stochastic hybrid systems
- encoding of the transition probabilities within the abstractions
- formal modelling language for stochastic hybrid systems

Recommendations for future research



- predictive maintenance via fault maintenance trees
- analysis of fault maintenance trees via StocHy
- connect the library of models for BAS with the Brick schema
- expand library of models for BAS

Associated publications

Chapter 3

- **N. Cauchi**, K. A. Hoque, A. Abate, and M. Stoelinga, "Efficient probabilistic model checking of smart building maintenance using fault maintenance trees," in Proceedings of the 4th ACM International Conference on Systems for Energy-Efficient Built Environments (**BuildSys 2017**). ACM, 2017
- A. Abate, C. E. Budde, **N. Cauchi**, A. V. Harmelen, K. A. Hoque, and M. Stoelinga, "Modelling smart buildings using fault maintenance trees," in 15th European Performance Engineering Workshop (**EPEW 2018**), Springer, 2018.
- A. Abate, C. E. Budde, **N. Cauchi**, K. A. Hoque, and M. Stoelinga, "Assessment of maintenance policies for smart buildings," in 4th European Conference of the Prognostics and Health Management Society (**PHME 2018**), 2018.
- **N. Cauchi**, K. A. Hoque, A. Abate, and M. Stoelinga, "Maintenance of smart buildings using fault trees,ÄCM Transactions on Sensor Networks, SI: Systems for Smart and Efficient Built Environments (**TOSN**),2018.

Associated publications

Chapter 4 and 5

- S. Haesaert, **N. Cauchi**, and A. Abate,"Certified policy synthesis for general markov decision processes: An application in building automation systems," **Performance Evaluation**, 2017.
- **N. Cauchi** and A. Abate, "Benchmarks for cyber-physical systems: A modular model library for buildings automation," in IFAC Conference on Analysis and Design of Hybrid Systems, (**ADHS**), 2018
- **N. Cauchi**, et al. "ARCH-COMP18 Category Report: Stochastic Modelling," in Proceedings of the 5th International Workshop on Applied Verification for Continuous and Hybrid Systems (**ARCH**), 2018.
- **N. Cauchi**, L. Laurenti, M. Lahijanian, A. Abate, M. Kwiatkowska and L. Cardelli, "Efficiency through uncertainty: scalable formal synthesis for stochastic hybrid systems," ACM International conference on Hybrid Systems: Computation and Control (**HSCC**), 2019.

Associated publications

Chapter 4 and 5

- **N. Cauchi**, et al. "ARCH-COMP19 Category Report: Stochastic Modelling," in Proceedings of the 5th International Workshop on Applied Verification for Continuous and Hybrid Systems (**ARCH**), 2019.

Chapter 6

- **N. Cauchi**, and A. Abate,"StocHy: Automated verification and synthesis of stochastic processes," in Tools and algorithms for the construction and analysis of systems **TACAS**, 2019.

Further

- K. Macek, P. Endel **N. Cauchi** and A. Abate. "Long-term predictive maintenance: a study of optimal cleaning of biomass boilers," in **Energy and Buildings**, 2017.
- **N. Cauchi**, K. Macek and A. Abate. "Model-based predictive maintenance in building automation systems with user discomfort," in **Energy**, 2017.

Thank you!

Questions?

Low-level certification: stochastic hybrid systems

special case

actions maybe associated to a deterministic selection of locations, namely $T_q : \mathcal{U} \to \mathcal{Q}$ and $\mathcal{U}$ is a finite set of actions.

Low-level certification: stochastic hybrid systems

special case

actions maybe associated to a deterministic selection of locations, namely $T_q : \mathcal{U} \to \mathcal{Q}$ and $\mathcal{U}$ is a finite set of actions.

Transition kernel

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

$$T_x(\mathcal{B} \mid q, x, u) = \int_{\mathcal{B}} \mathcal{N}(t \mid F(q, x, u), G(q)\Sigma_w G(q)^T)\, dt.$$

Low-level certification: stochastic hybrid systems

special case

actions maybe associated to a deterministic selection of locations, namely $T_q : \mathcal{U} \to \mathcal{Q}$ and $\mathcal{U}$ is a finite set of actions.

Transition kernel

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

$$T_x(\mathcal{B} \mid q, x, u) = \int_{\mathcal{B}} \mathcal{N}(t \mid F(q, x, u), G(q)\Sigma_w G(q)^T)\, dt.$$

temporal logic specifications

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

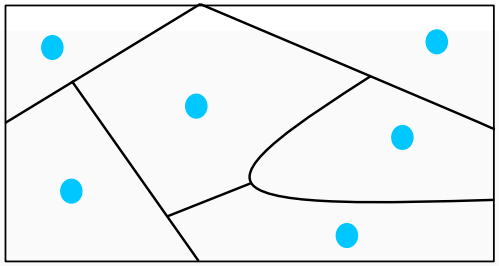
Low-level certification: MDP abstractions

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) of transition kernel
- N-step error

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

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



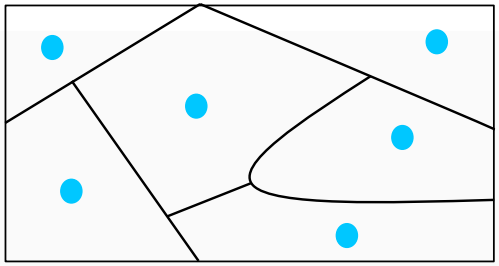
Low-level certification: MDP abstractions

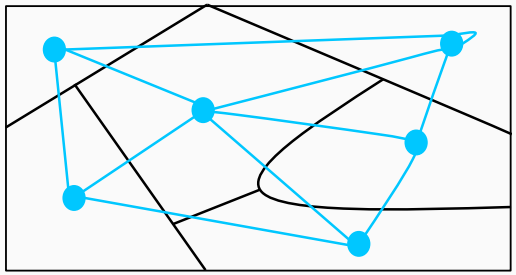
abstraction into MDP

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

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

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





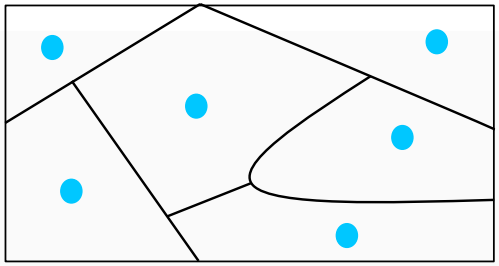
Low-level certification: MDP abstractions

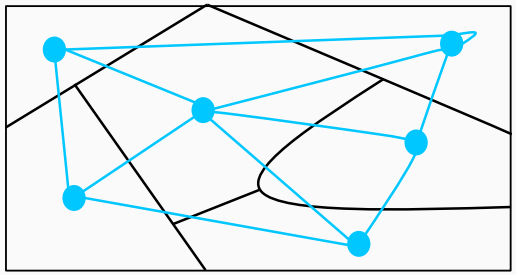
abstraction into MDP

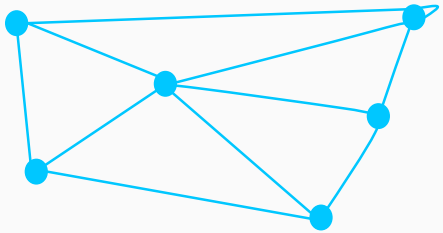
- define desired abstraction error
- grid state-space (uniform or adaptive)
- compute transition probabilities via marginalisation
- hinges on computation of Lipschitz constants (h_s) of transition kernel
- N-step error

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

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







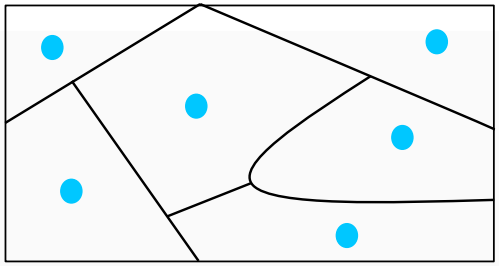
Low-level certification: MDP abstractions

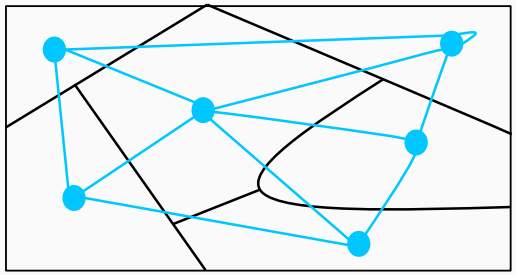
abstraction into MDP

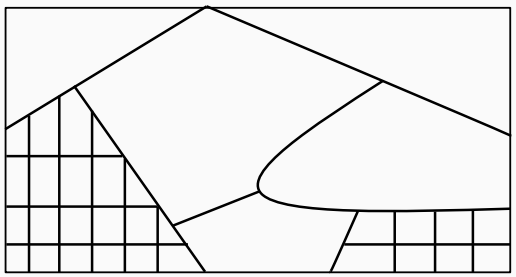
- define desired abstraction error
- grid state-space (uniform or adaptive)
- compute transition probabilities via marginalisation
- hinges on computation of Lipschitz constants (h_s) of transition kernel
- N-step error

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

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



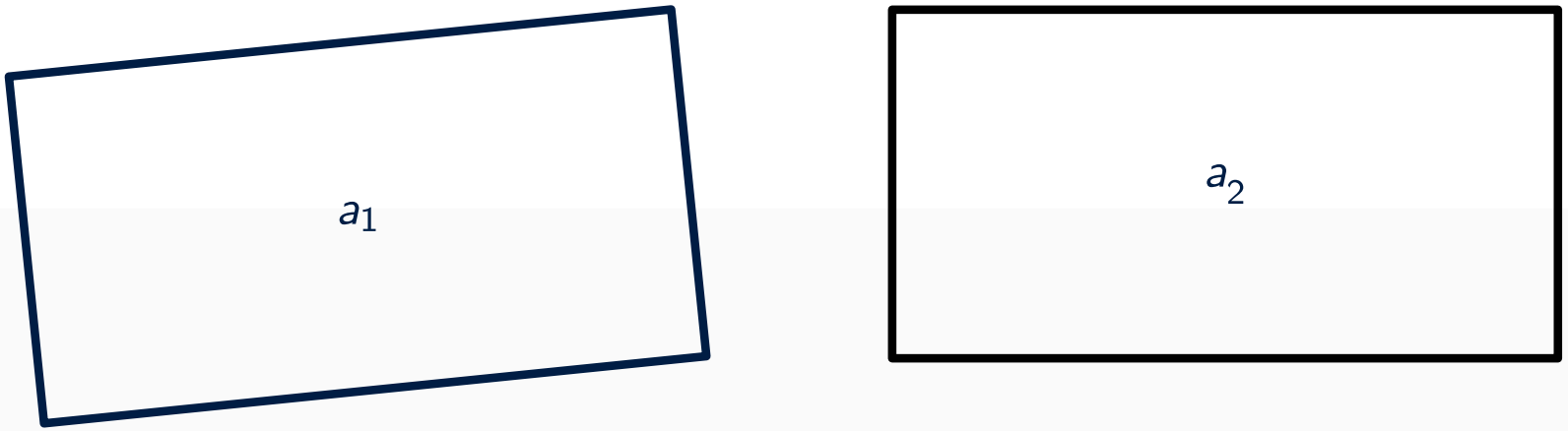




Low-level certification: MDP abstractions

We abstract the shs to an IMDP

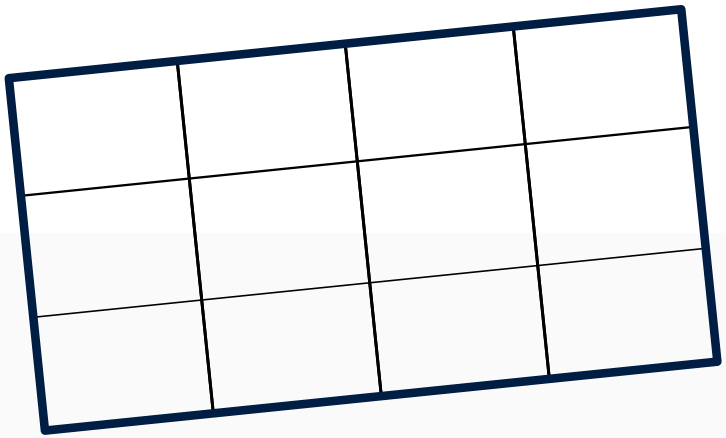
$$\mathcal{I} = (\mathcal{S}, \mathcal{A}, \check{T}_s, \hat{T}_s, \Theta, L)$$

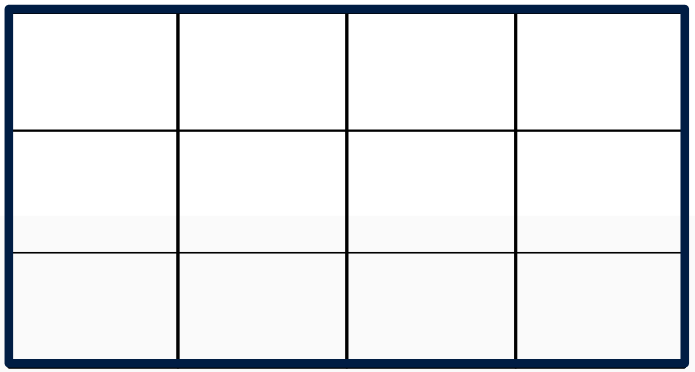


Low-level certification: MDP abstractions

We abstract the shs to an IMDP

$$\mathcal{I} = (\mathcal{S}, \mathcal{A}, \check{T}_s, \hat{T}_s, \Theta, L)$$

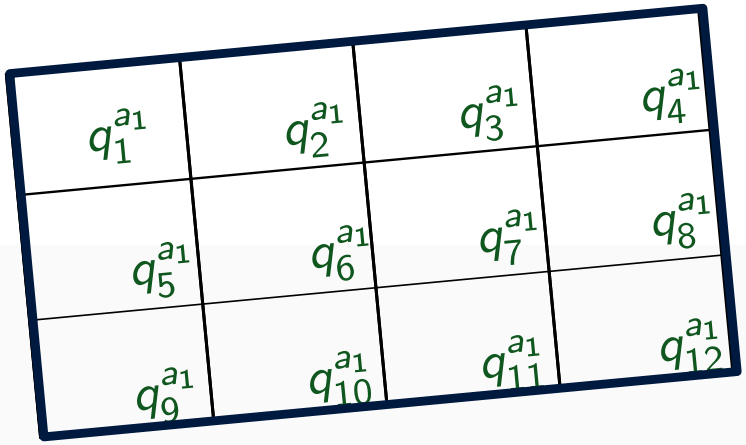


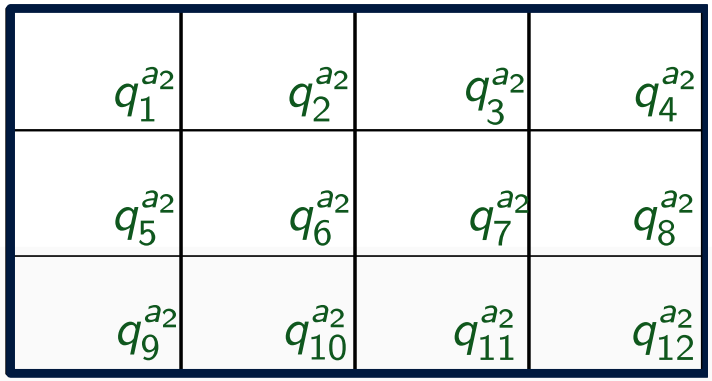


Low-level certification: MDP abstractions

We abstract the shs to an IMDP

$$\mathcal{I} = (\mathcal{S}, \mathcal{A}, \check{T}_s, \hat{T}_s, \Theta, L)$$





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

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

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

Low-level certification: MDP abstractions

We abstract the shs to an IMDP

$$\mathcal{I} = (\mathcal{S}, \mathcal{A}, \check{T}_s, \hat{T}_s, \Theta, L)$$

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

Low-level certification: MDP abstractions

We abstract the shs to an IMDP

$$\mathcal{I} = (\mathcal{S}, \mathcal{A}, \check{T}_s, \hat{T}_s, \Theta, L)$$

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

Low-level certification: MDP abstractions

We abstract the shs to an IMDP

$$\mathcal{I} = (\mathcal{S}, \mathcal{A}, \check{T}_s, \hat{T}_s, \Theta, L)$$

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

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

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

Low-level certification: computation of IMDP

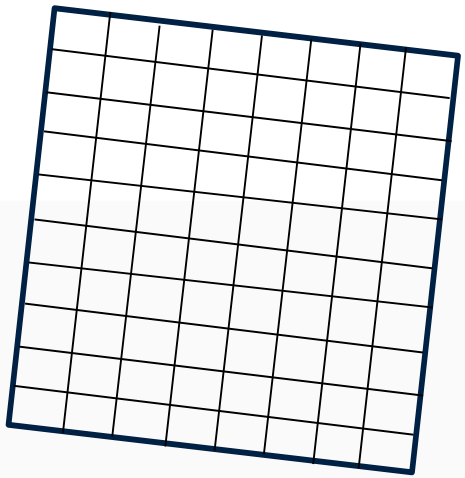
How do we efficiently compute

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

Low-level certification: computation of IMDP

How do we efficiently compute

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

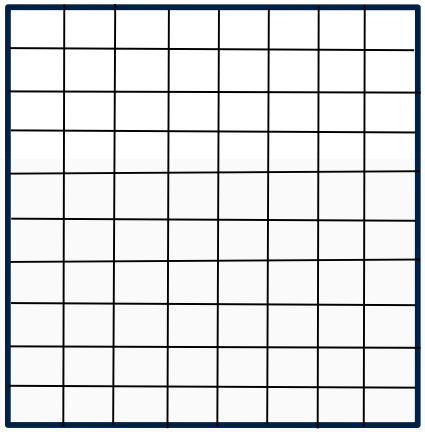


For process $\mathbf{x}$ in mode $a \in \mathcal{A}$

Transformation

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

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



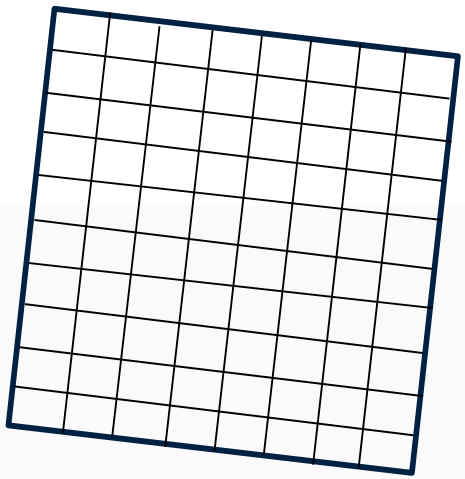
Hyper rectangles

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

Low-level certification: computation of IMDP

How do we efficiently compute

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

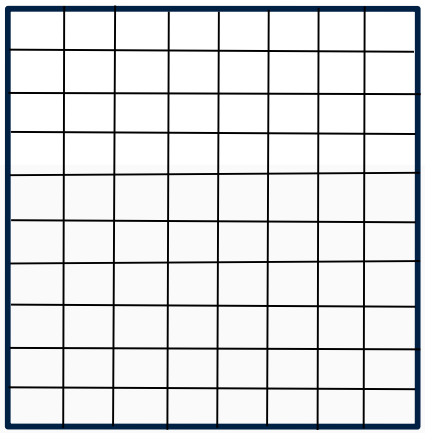


For process x in mode $a \in \mathcal{A}$

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

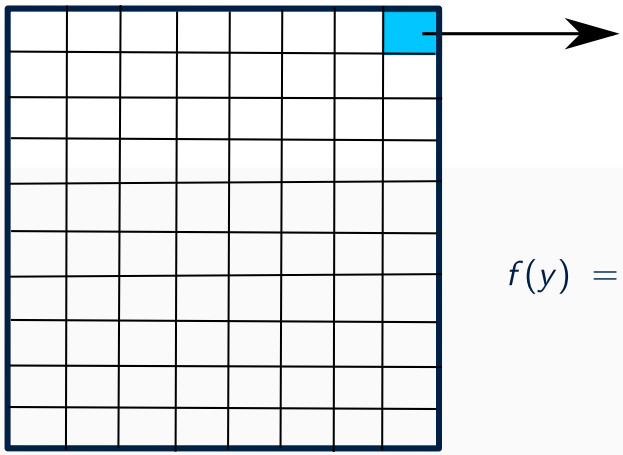
$$[v_l^{(1)}, v_u^{(1)}] \times \cdots \times [v_l^{(m)}, v_u^{(m)}]$$

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

Low-level certification: computation of IMDP

How do we efficiently compute

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



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

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

Optimisation via KKT or Gradient descent

Low-level certification: computation of IMDP

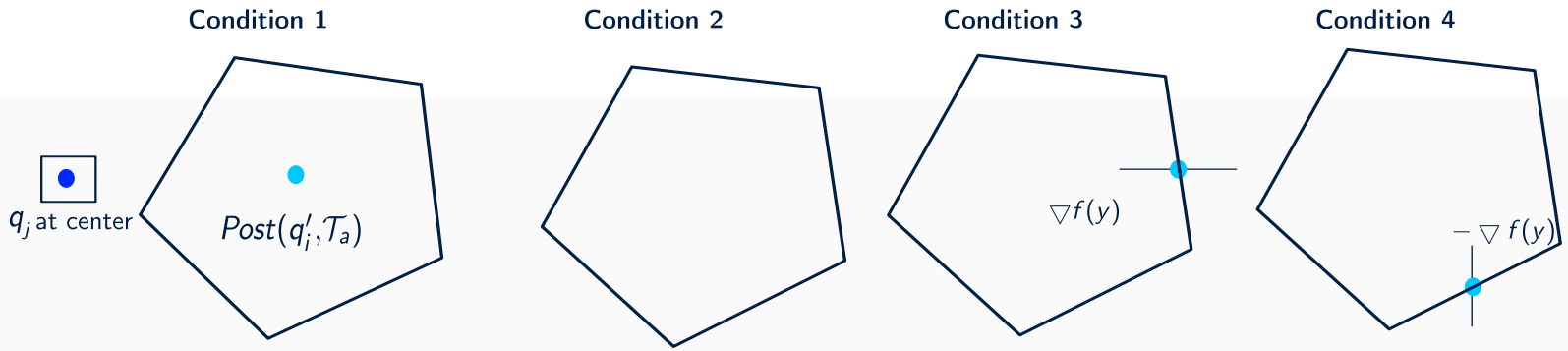
Optimisation via KKT conditions

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

Low-level certification: computation of IMDP

Optimisation via KKT conditions

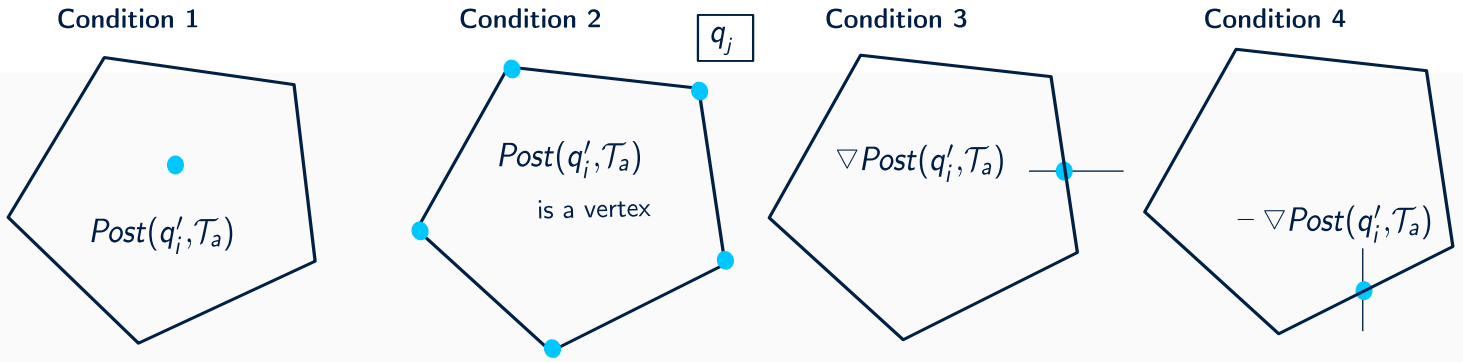
- solving systems of non-linear equations
- efficient and exact for low-dimensional system
- number of vertices to check grows exponentially with dimensions



Low-level certification: 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



Low-level certification: computation of IMDP

Optimisation via gradient descent method

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

Low-level certification: computation of IMDP

Optimisation via gradient descent method

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

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

Low-level certification: computation of IMDP

We abstract the shs to an IMDP

$$\mathcal{I} = (\mathcal{S}, \mathcal{A}, \check{T}_s, \hat{T}_s, \Theta, L)$$

- 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

Labelling

- conservatively overapproximate discretizations of $\mathcal{A} \times X$ that do not 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 $1 \leq i \leq n$.

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

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

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

for all $s \in \bar{\mathcal{S}}$ and $0 \leq i \leq 2n$, and $L(s_u) = \emptyset$