Efficient Synthesis of Safety Controllers Using Symbolic Models and Lazy Algorithms

Elena Ivanova

LIX, École polytechnique

8 Mars 2022

Cyber-physical systems

Cyber-physical systems are the integration of computational devices with physical processes: embedded computers monitor and control physical processes, which in return affect computations through information feedback loops.

Cyber-physical systems

Cyber-physical systems are the integration of computational devices with physical processes: embedded computers monitor and control physical processes, which in return affect computations through information feedback loops.



Challenges when developing CPS:

- CPS models are heterogeneous. The continuous behavior is described by differential equations, while the discrete behavior is formalized with finite-state automata frameworks.
- Complex control objectives. Reachability, fault-tolerance, LTL formulas etc..

CPS. Safety Specification

Cyber-physical systems are often safety critical.
Safety specification: the behavior of the controlled cyber-physical system should not violate the safety restrictions.

CPS. Safety Specification

Cyber-physical systems are often safety critical.
Safety specification: the behavior of the controlled cyber-physical system should not violate the safety restrictions.

- Climate control should maintain the temperature in an intelligent building into the desirable range.

CPS. Safety Specification

Cyber-physical systems are often safety critical.
Safety specification: the behavior of the controlled cyber-physical system should not violate the safety restrictions.

- Climate control should maintain the temperature in an intelligent building into the desirable range.
- Insulin pumps should protect a diabetic person from hyper or hypoglycemia.

CPS. Safety Specification

Cyber-physical systems are often safety critical.
Safety specification: the behavior of the controlled cyber-physical system should not violate the safety restrictions.

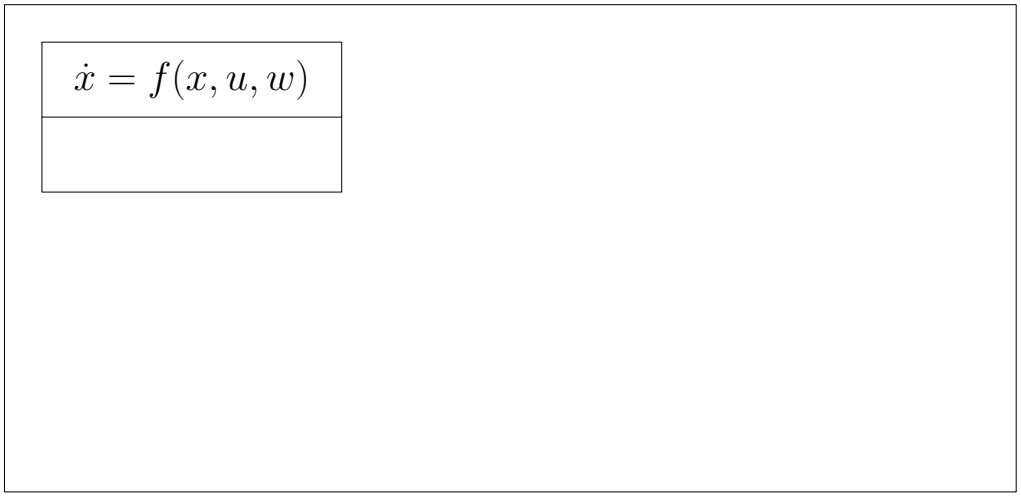

- Climate control should maintain the temperature in an intelligent building into the desirable range.
- Insulin pumps should protect a diabetic person from hyper or hypoglycemia.
- Adaptive cruise control assistants should keep a car at a safe distance from the previous vehicle.

Problem Statement

Continous-time control system:

$$\dot{x} = f(x, u, w), x \in \mathbb{R}^n, u \in U \subset \mathbb{R}^p, w \in W \subset \mathbb{R}^m.$$

Safety specification: design a controller $u(t, x)$ maintaining all trajectories of the closed-loop system within a safe set S.

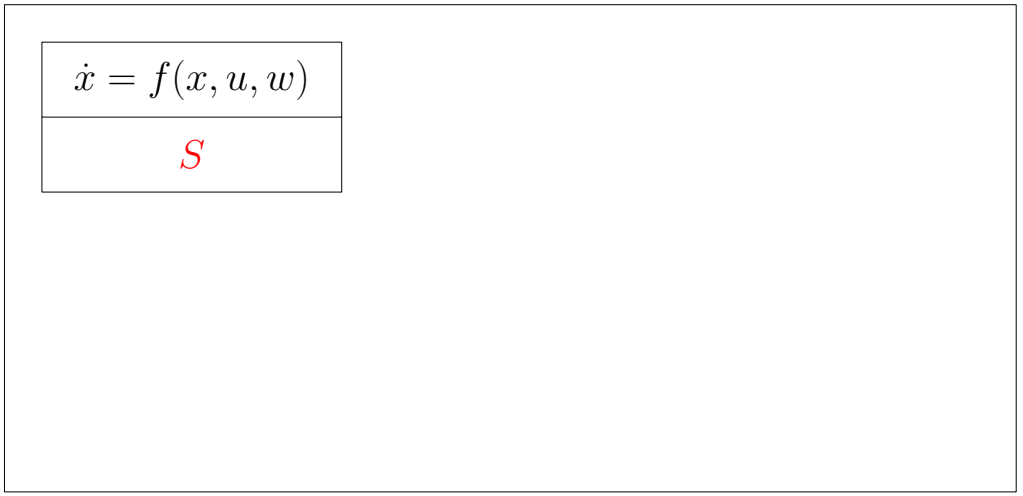


Problem Statement

Continous-time control system:

$$\dot{x} = f(x, u, w), x \in \mathbb{R}^n, u \in U \subset \mathbb{R}^p, w \in W \subset \mathbb{R}^m.$$

Safety specification: design a controller $u(t, x)$ maintaining all trajectories of the closed-loop system within a safe set S.

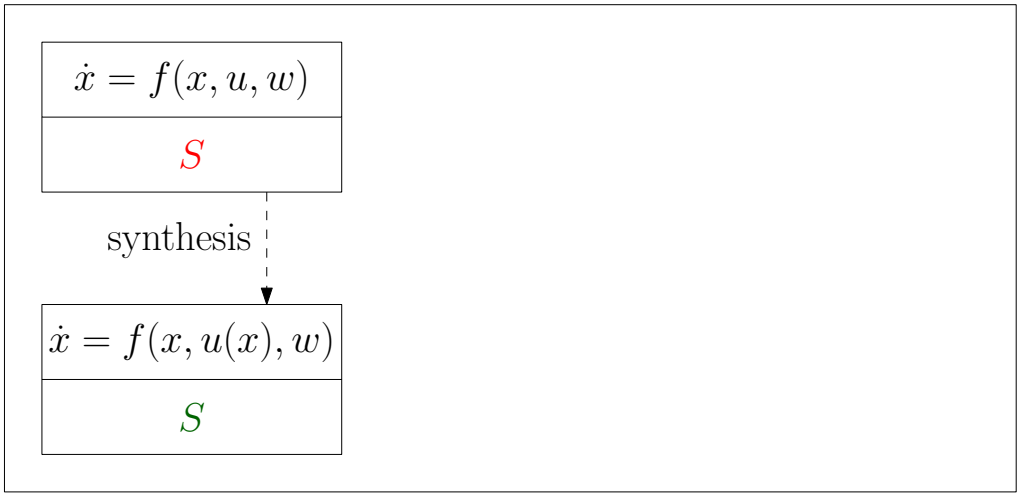


Problem Statement

Continous-time control system:

$$\dot{x} = f(x, u, w), x \in \mathbb{R}^n, u \in U \subset \mathbb{R}^p, w \in W \subset \mathbb{R}^m.$$

Safety specification: design a controller $u(t, x)$ maintaining all trajectories of the closed-loop system within a safe set S.

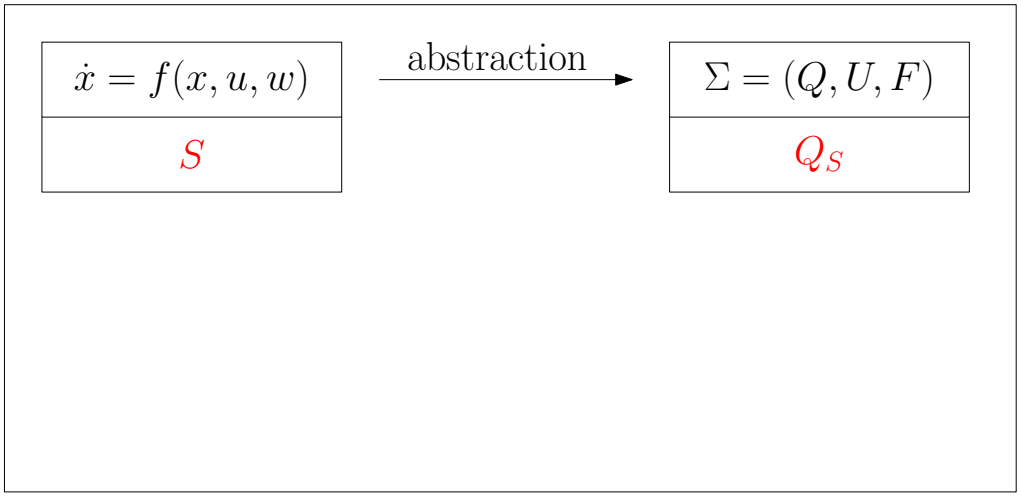


Problem Statement

Continous-time control system:

$$\dot{x} = f(x, u, w), x \in \mathbb{R}^n, u \in U \subset \mathbb{R}^p, w \in W \subset \mathbb{R}^m.$$

Safety specification: design a controller $u(t, x)$ maintaining all trajectories of the closed-loop system within a safe set S.



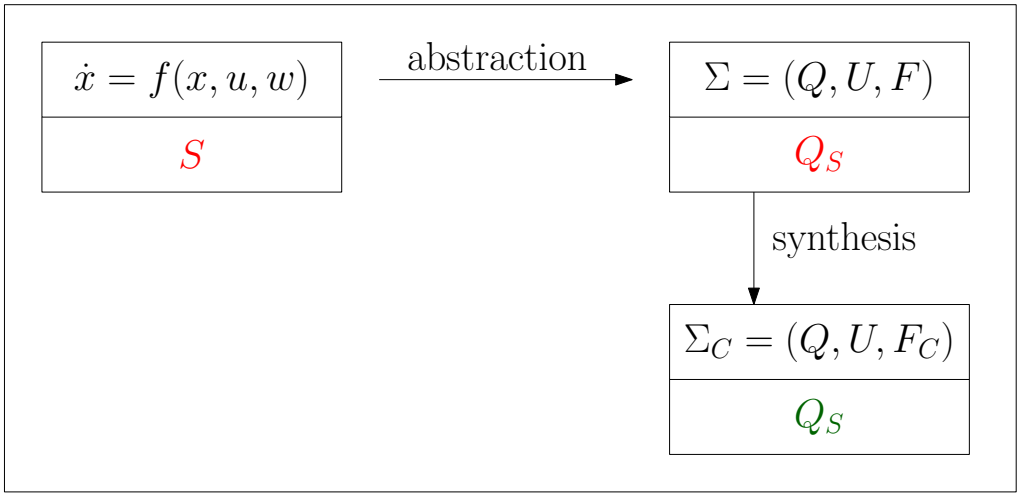
Abstraction-based control synthesis approach.

Problem Statement

Continous-time control system:

$$\dot{x} = f(x, u, w), x \in \mathbb{R}^n, u \in U \subset \mathbb{R}^p, w \in W \subset \mathbb{R}^m.$$

Safety specification: design a controller $u(t, x)$ maintaining all trajectories of the closed-loop system within a safe set S.



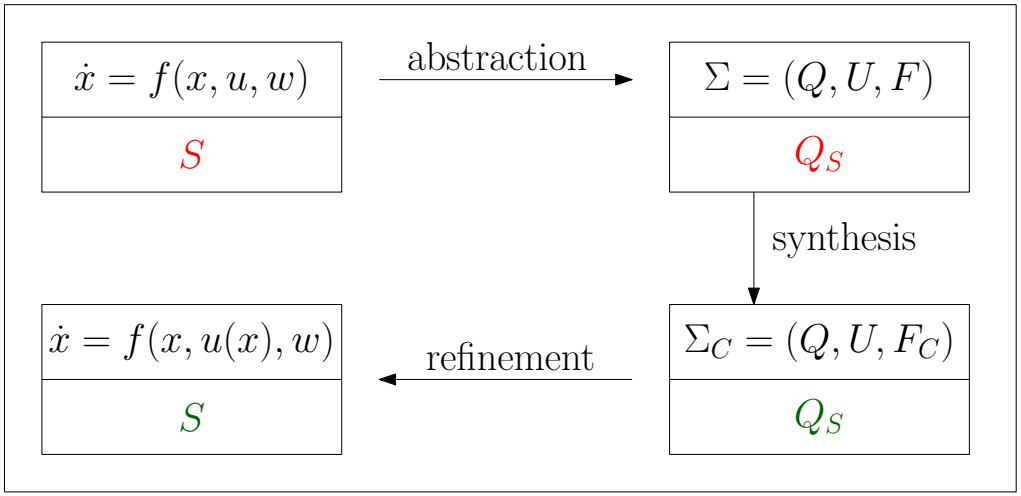
Abstraction-based control synthesis approach.

Problem Statement

Continous-time control system:

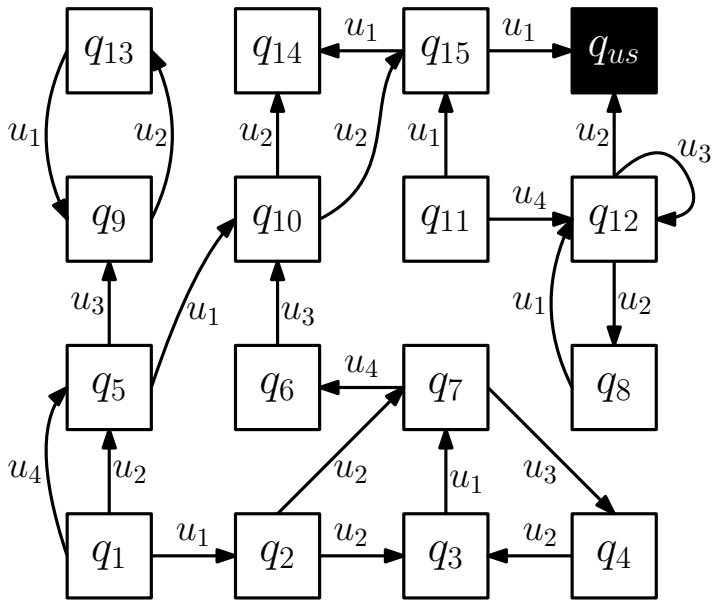
$$\dot{x} = f(x, u, w), x \in \mathbb{R}^n, u \in U \subset \mathbb{R}^p, w \in W \subset \mathbb{R}^m.$$

Safety specification: design a controller $u(t, x)$ maintaining all trajectories of the closed-loop system within a safe set S.



Abstraction-based control synthesis approach.

Finite Transition System. Safety Specification



A finite transition system (FTS) is a tuple $\Sigma = (Q, U, F)$, consisting of

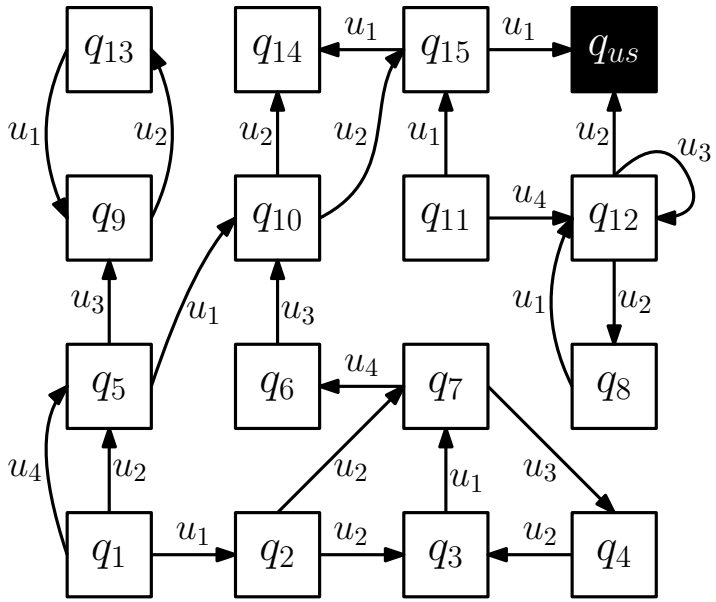
- a finite set of states Q.
  $Q = \{q_{us}, q_1, q_2 \ldots, q_{15}\}$
- a finite set of inputs U.
  $U = \{u_1, u_2, u_3, u_4\}$
- transition relation $F \subseteq Q \times U \times Q$.
  $(q, u, q') \in F \Leftrightarrow q' \in F(q, u)$

A state $q \in Q$ is blocking if $F(q, u) = \emptyset$ for any $u \in U$.

A trajectory is a finite or infinite sequence of transitions $q_0 \xrightarrow{u_0} q_1 \xrightarrow{u_1} \ldots$, s.t. $q^i \in Q, u^i \in U$ and $q^{i+1} \in F(q^i, u^i)$ for all $i \geq 0$.

Safety specification: Safe set $Q_S \subset Q \Rightarrow$ Unsafe set $Q \setminus Q_S$.
$Q_S = \{q_1, q_2, \ldots, q_{15}\} \Rightarrow Q \setminus Q_S = \{q_{us}\}$.

FTS. Maximal Safety Controller



A controller is a map $C\colon Q \to 2^U$, such that $C(q) \subseteq \mathrm{En}_F(q)$ for every $q \in Q$, where $\mathrm{En}_F(q) = \{u \in U \mid F(q,u) \neq \emptyset\}$.

A safety controller C is a controller such that

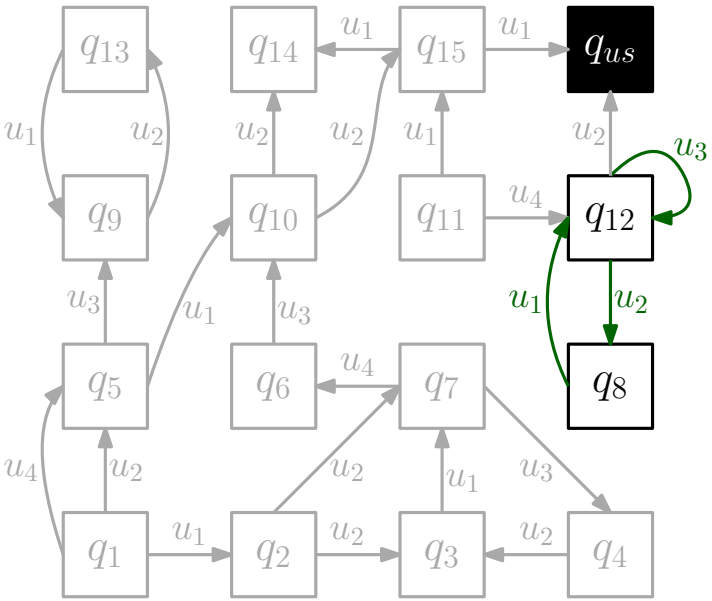
- $\mathrm{Dom}(C) \subseteq Q_S$, where $\mathrm{Dom}(C) = \{q \in Q \mid C(q) \neq \emptyset\}$.
- for all $q \in \mathrm{Dom}(C)$ and for all $u \in C(q) \Rightarrow F(q,u) \subseteq \mathrm{Dom}(C)$.

There is a unique maximal safety controller C^* such that for any safety controller C

- $\mathrm{Dom}(C) \subseteq \mathrm{Dom}(C^*)$.
- for all $q \in \mathrm{Dom}(C)$, $C(q) \subseteq C^*(q)$.

The set of safely controllable states is a set $\mathrm{Cont}(\Sigma, Q_S) = \{q \in Q \mid q \in \mathrm{Dom}(C^*)\}$.

FTS. Maximal Safety Controller



A controller is a map $C\colon Q \to 2^U$, such that $C(q) \subseteq \mathrm{En}_F(q)$ for every $q \in Q$, where $\mathrm{En}_F(q) = \{u \in U \mid F(q,u) \neq \emptyset\}$.

A safety controller C is a controller such that

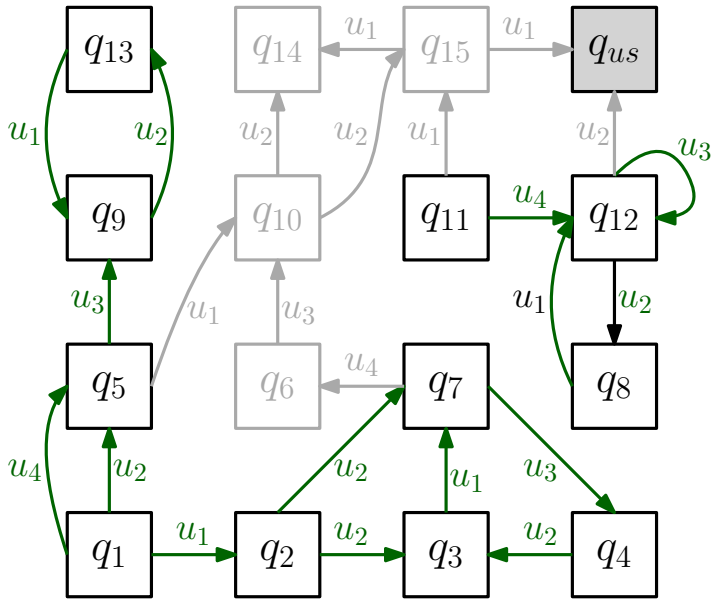
- $\mathrm{Dom}(C) \subseteq Q_S$, where $\mathrm{Dom}(C) = \{q \in Q \mid C(q) \neq \emptyset\}$.
- for all $q \in \mathrm{Dom}(C)$ and for all $u \in C(q) \Rightarrow F(q,u) \subseteq \mathrm{Dom}(C)$.

There is a unique maximal safety controller C^* such that for any safety controller C

- $\mathrm{Dom}(C) \subseteq \mathrm{Dom}(C^*)$.
- for all $q \in \mathrm{Dom}(C)$, $C(q) \subseteq C^*(q)$.

The set of safely controllable states is a set $\mathrm{Cont}(\Sigma, Q_S) = \{q \in Q \mid q \in \mathrm{Dom}(C^*)\}$.

FTS. Maximal Safety Controller



A controller is a map $C\colon Q \to 2^U$, such that $C(q) \subseteq \mathrm{En}_F(q)$ for every $q \in Q$, where $\mathrm{En}_F(q) = \{u \in U \mid F(q,u) \neq \emptyset\}$.

A safety controller C is a controller such that

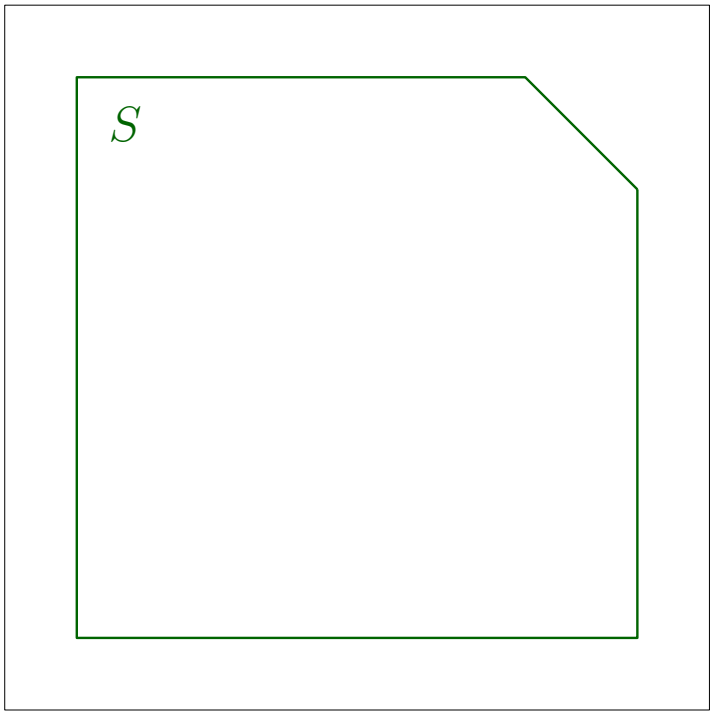
- $\mathrm{Dom}(C) \subseteq Q_S$, where $\mathrm{Dom}(C) = \{q \in Q \mid C(q) \neq \emptyset\}$.
- for all $q \in \mathrm{Dom}(C)$ and for all $u \in C(q) \Rightarrow F(q,u) \subseteq \mathrm{Dom}(C)$.

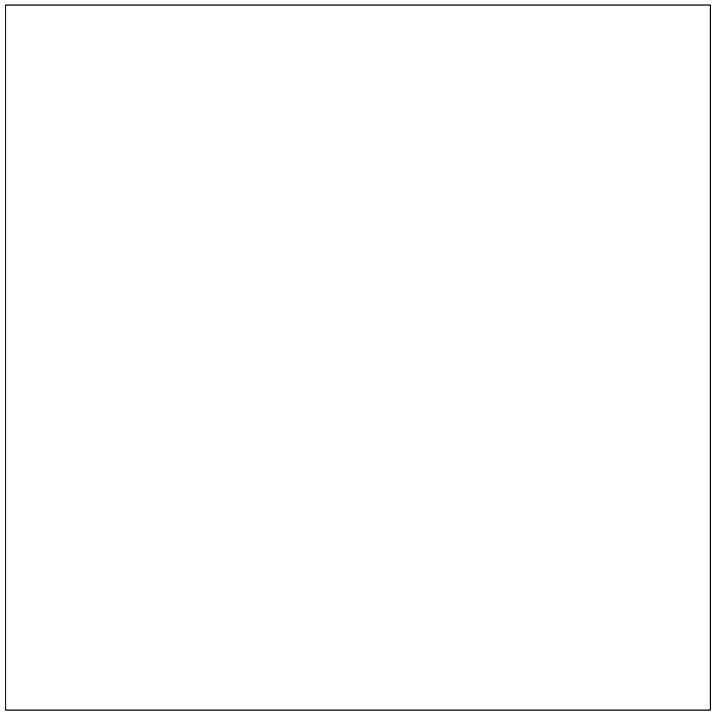
There is a unique maximal safety controller C^* such that for any safety controller C

- $\mathrm{Dom}(C) \subseteq \mathrm{Dom}(C^*)$.
- for all $q \in \mathrm{Dom}(C)$, $C(q) \subseteq C^*(q)$.

The set of safely controllable states is a set $\mathrm{Cont}(\Sigma, Q_S) = \{q \in Q \mid q \in \mathrm{Dom}(C^*)\}$.

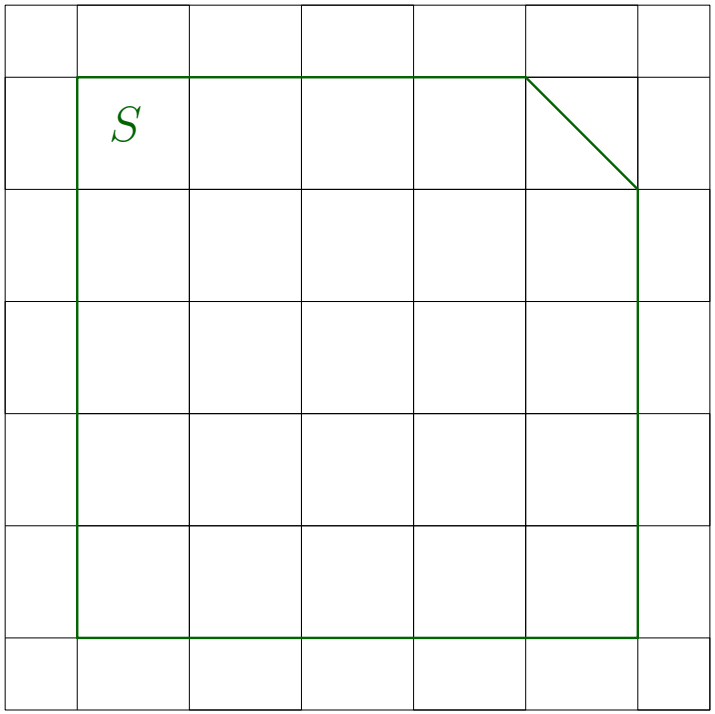
Symbolic Model

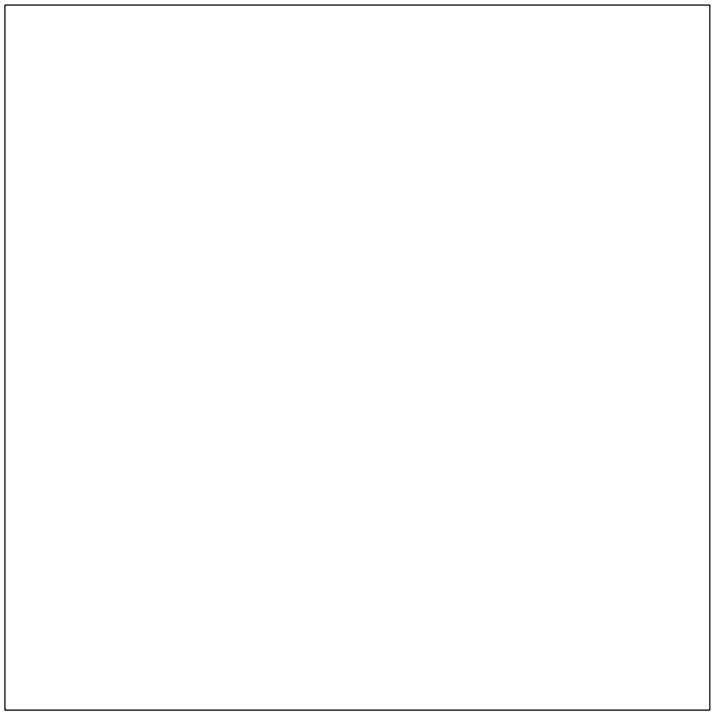




How do we create a finite transition system
which mimics the dynamic of the original plant?

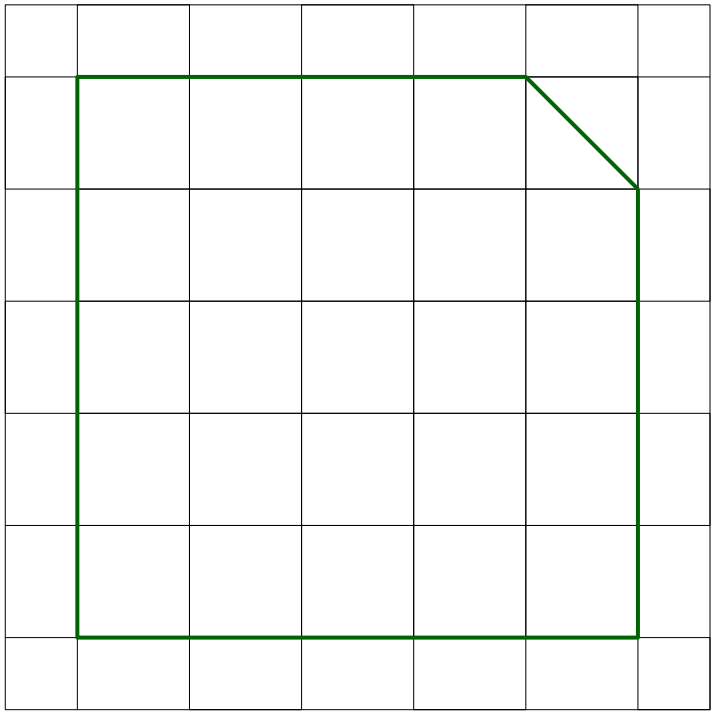
Symbolic Model

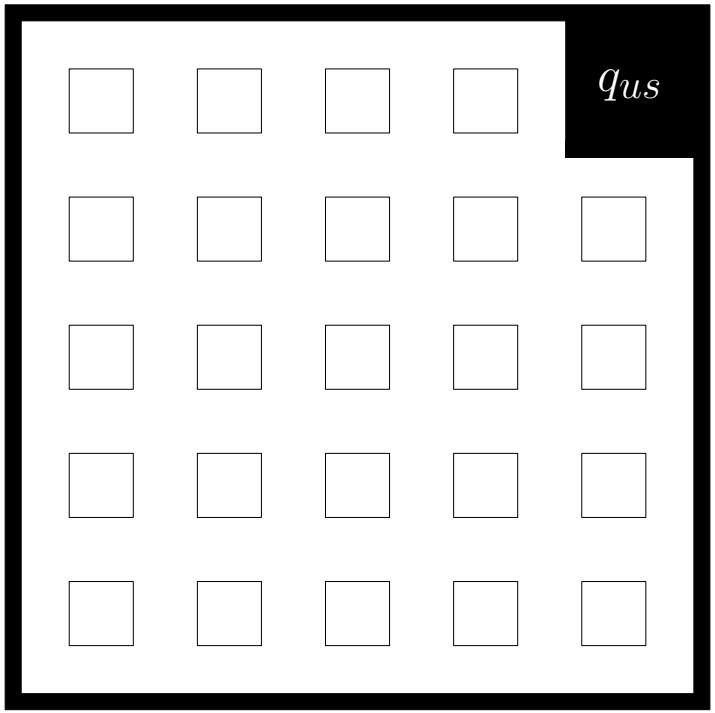




Introduce a partitioning on the state space $\mathbb{R}^n$ and associate every element of this partitioning with an abstract state.

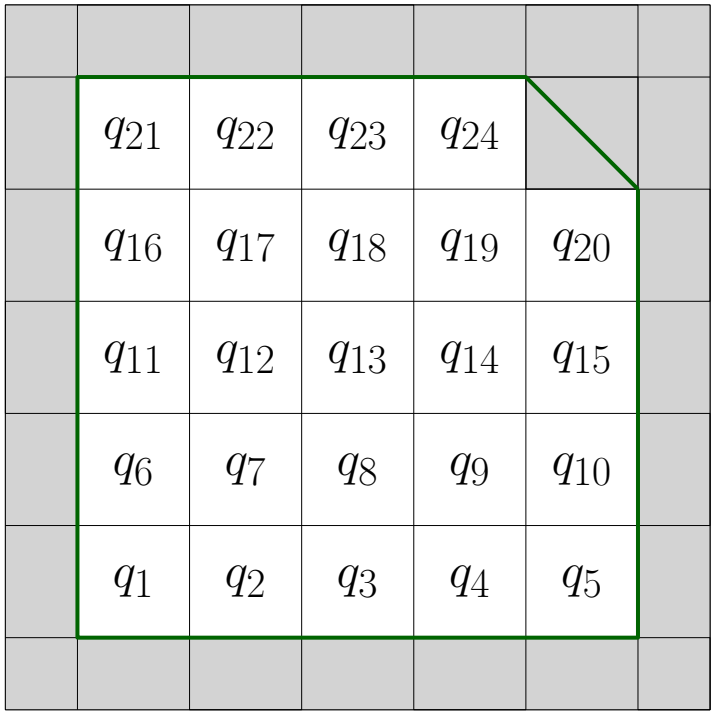
Symbolic Model

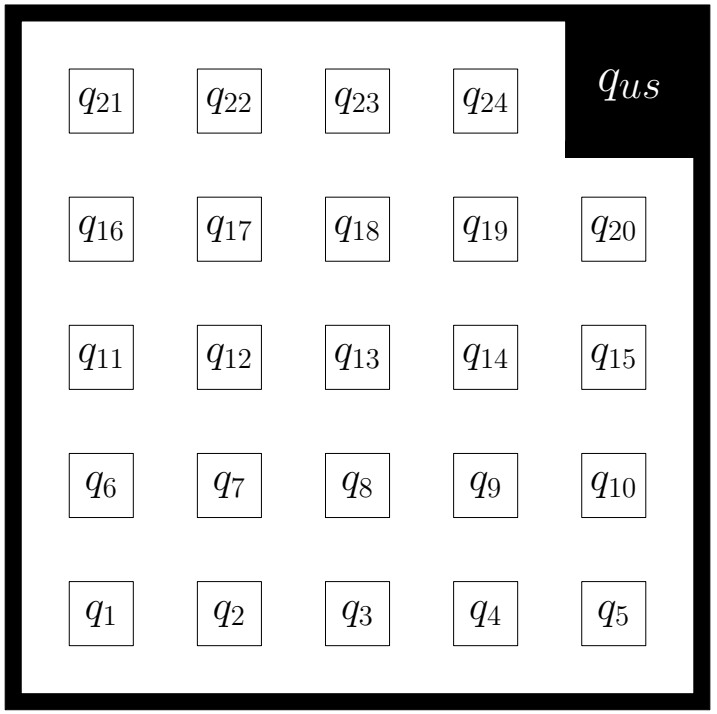




Introduce a partitioning on the state space $\mathbb{R}^n$ and associate every element of this partitioning with an abstract state.

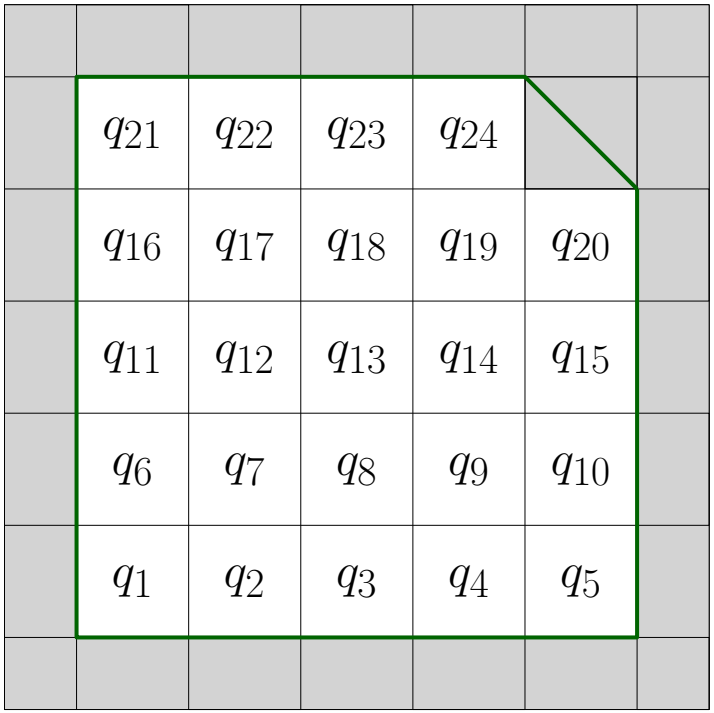
Symbolic Model

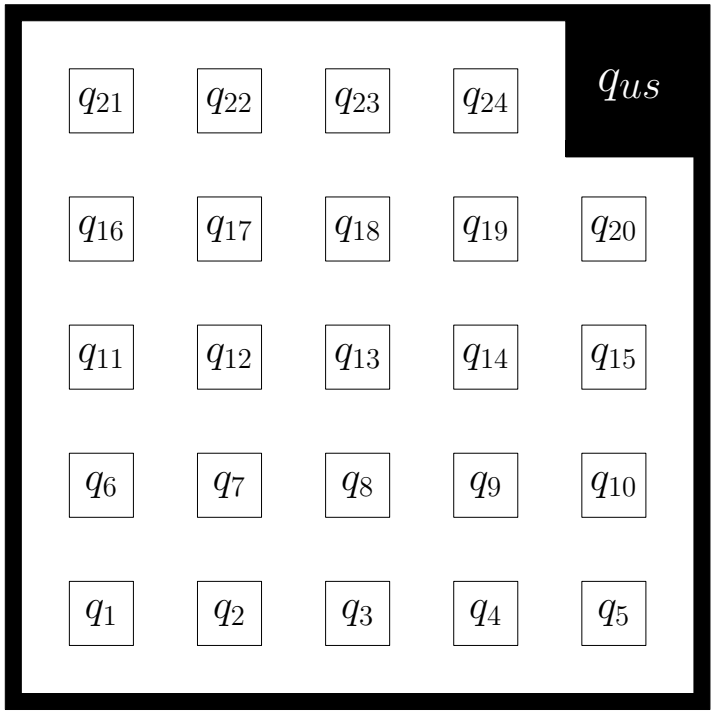




The elements belonging to a safe set S are marked as safe states, while all the others accumulated in the unsafe state q_{us}.

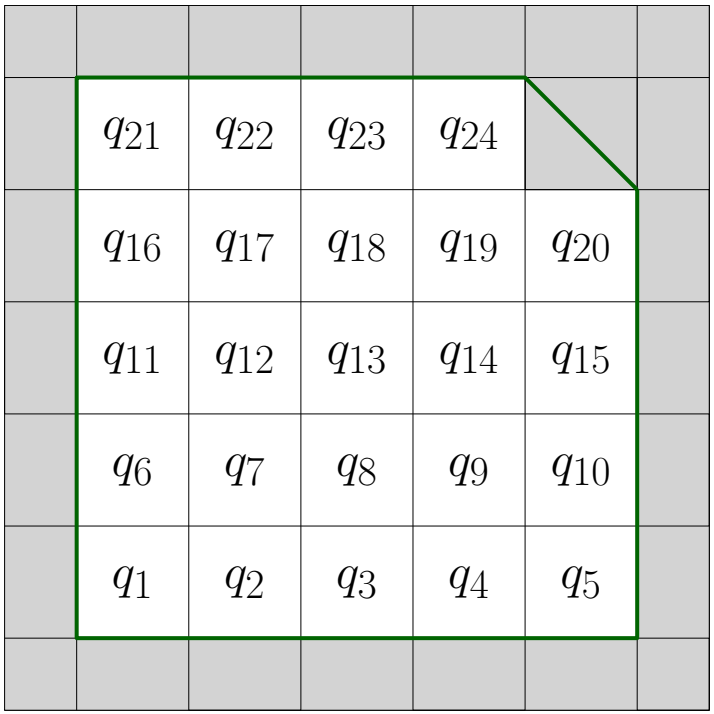
Symbolic Model

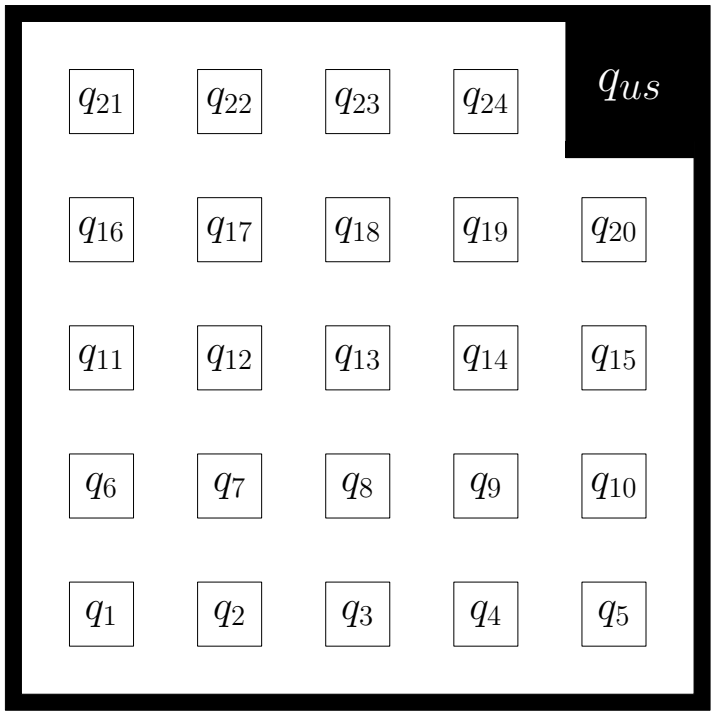




Replace the input set $U \subset \mathbb{R}^p$ by its finite approximation U_μ.
Introduce a time-sampling parameter τ.

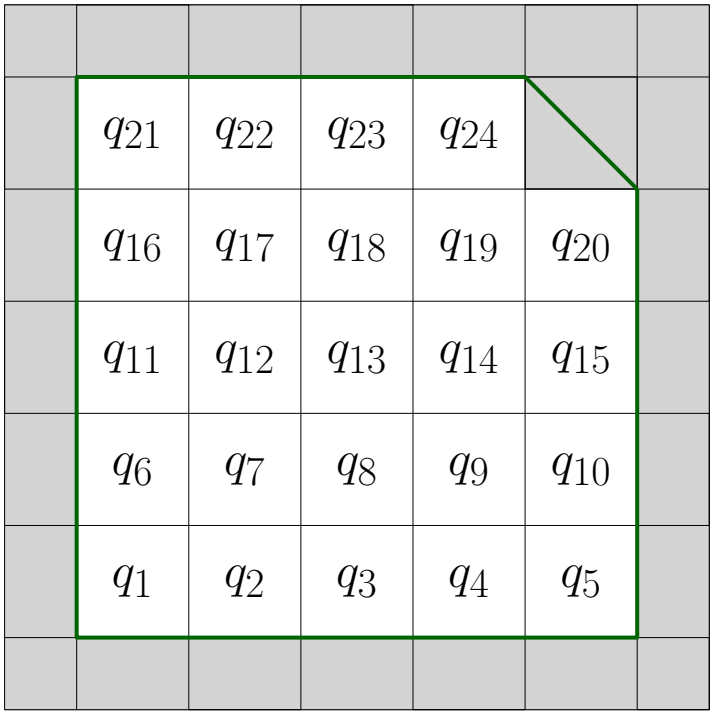
Symbolic Model

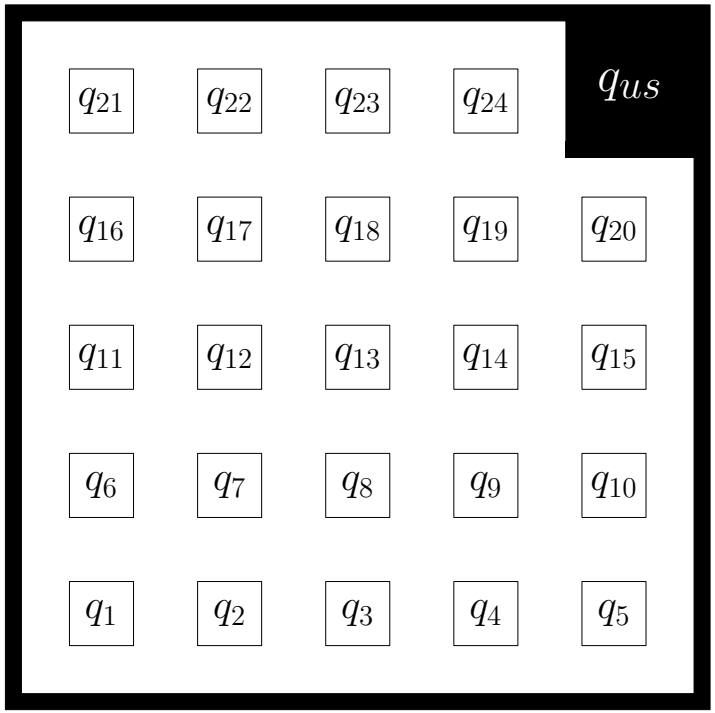




To compute the transition relation we use the notion of the reachable set robust to any admissible disturbance

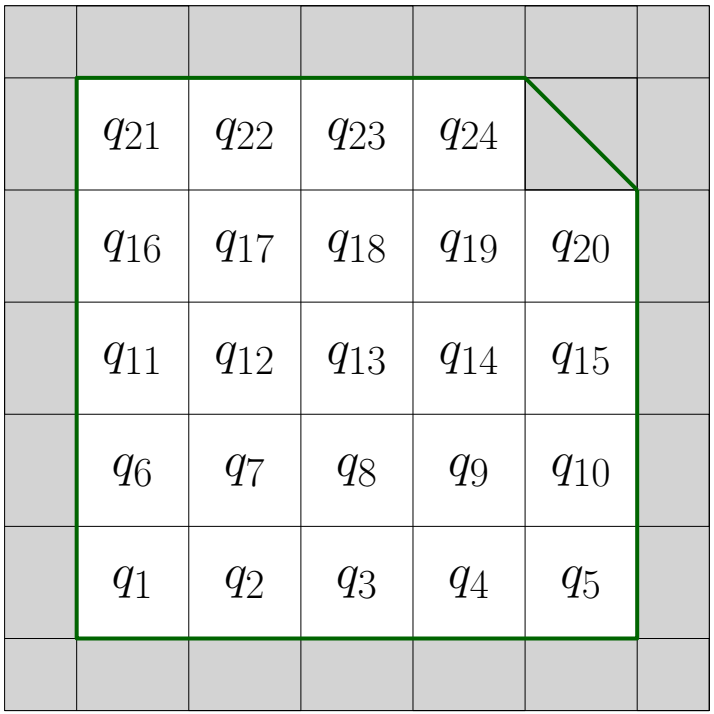
Symbolic Model

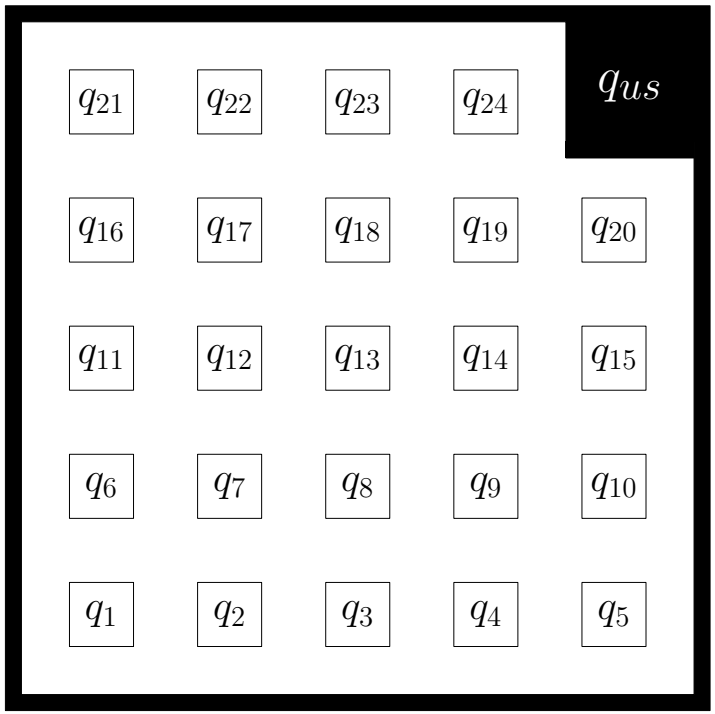




$$\mathrm{R}(t \mid q, u_\mu) = \{x \in \mathbb{R}^n \mid \exists\, x(0) \in q \text{ and } \exists\, w(\cdot) \in \mathcal{L}^\infty([0,t], W) \text{ such that } x_f(t \mid x(0), u_\mu, w(\cdot)) = x\}.$$

Symbolic Model

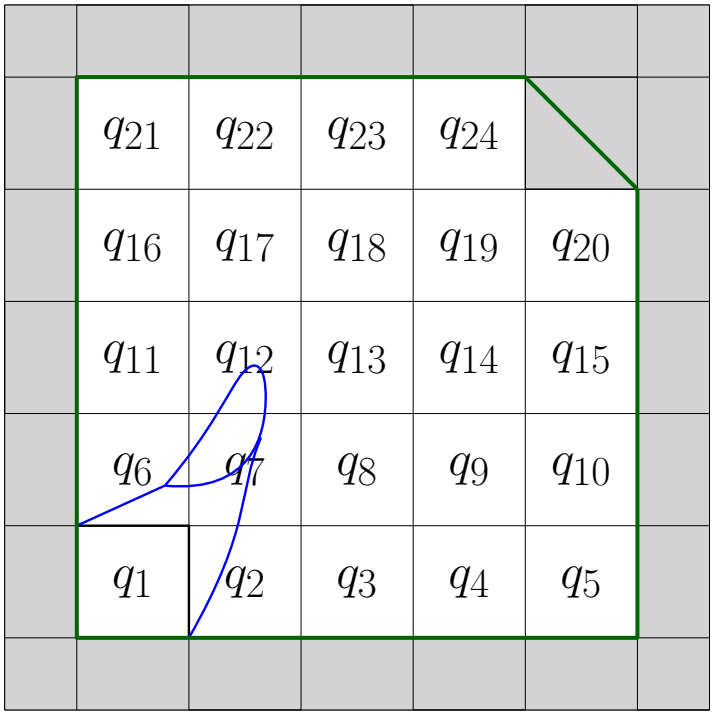


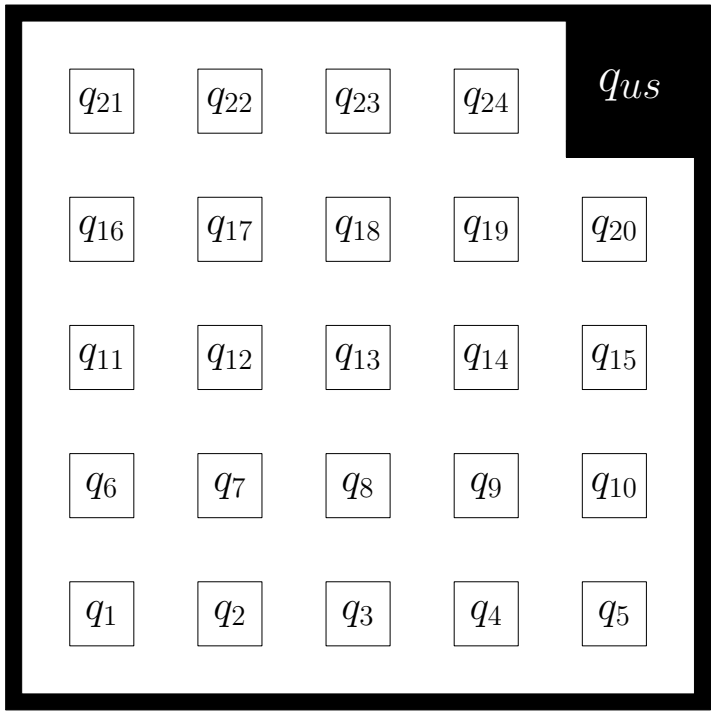


Since the exact computation of reachable set $\mathrm{R}(t \mid q, u_\mu)$ is quite a demanding process its over-approximations[1] $\overline{\mathrm{R}}(t \mid q, u_\mu)$ are used instead.

[1] P.-J. Meyer, A. Devonport and M. Arcak (2021). Interval Reachability Analysis.

Symbolic Model

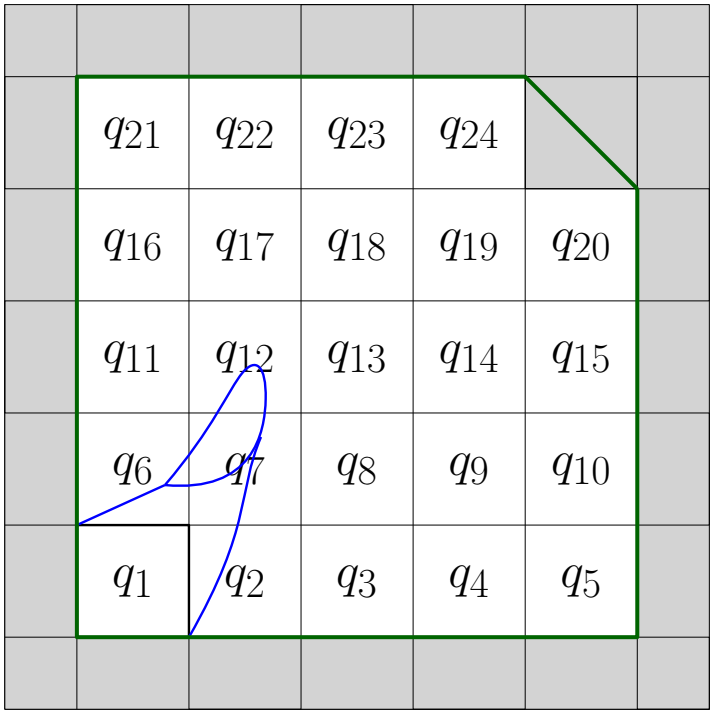


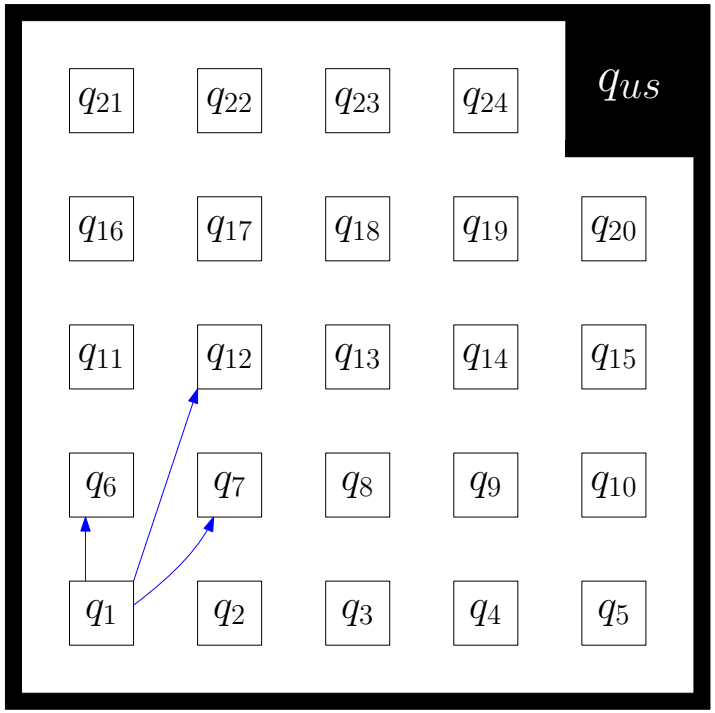


For every state $q \in Q$ and for every input $u \in U_\mu$ the transition

$(q, u_\mu, q') \in F \Leftrightarrow q' \cap \overline{\mathsf{R}}(\tau \mid q, u_\mu) \neq \emptyset$

Symbolic Model

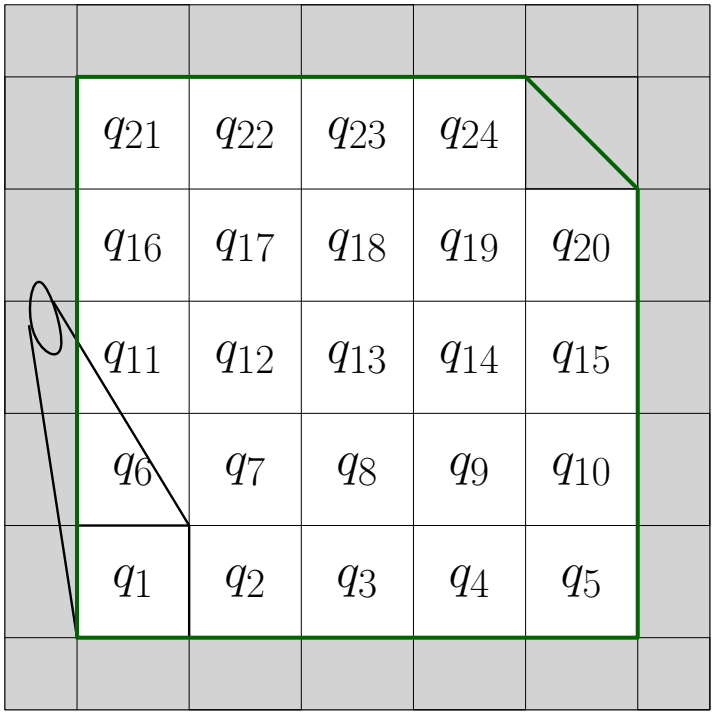


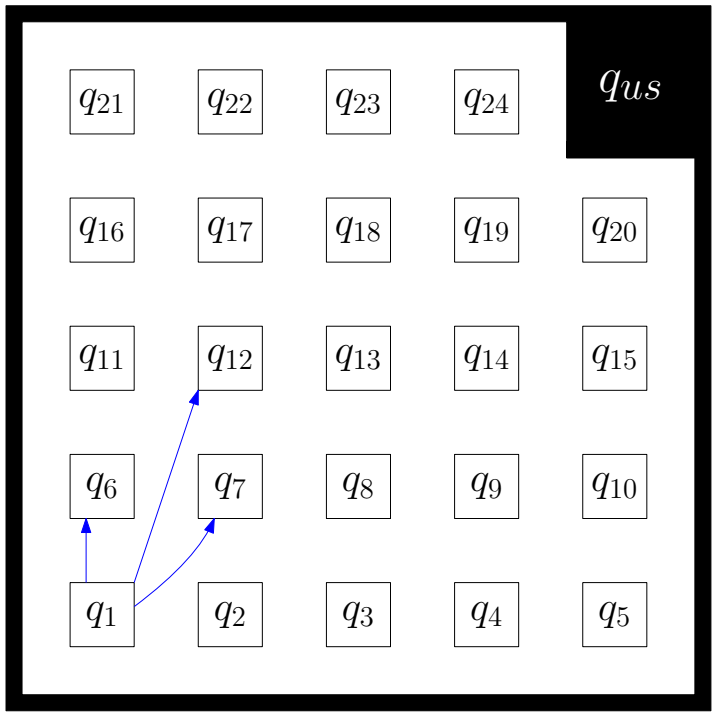


For every state $q \in Q$ and for every input $u \in U_\mu$ the transition

$$(q, u_\mu, q') \in F \Leftrightarrow q' \cap \overline{\mathrm{R}}(\tau \mid q, u_\mu) \neq \emptyset$$

Symbolic Model

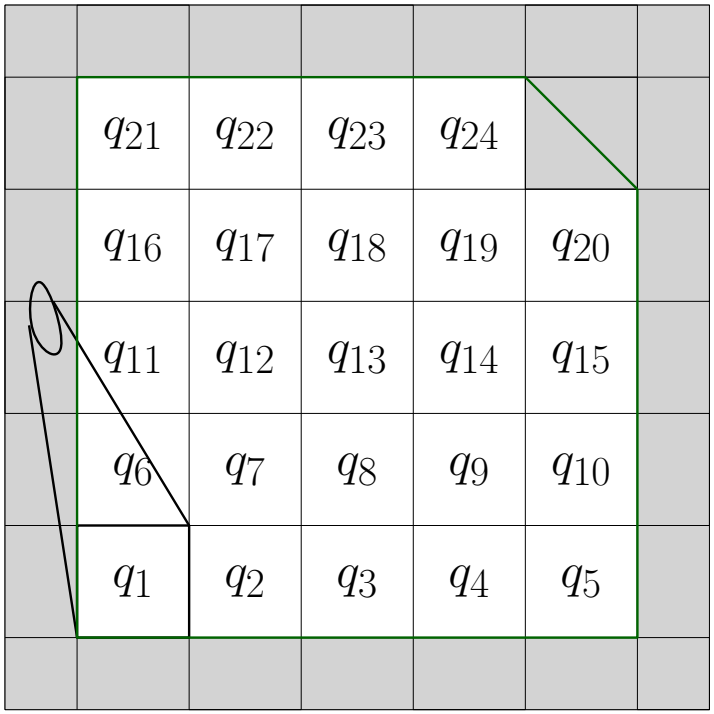


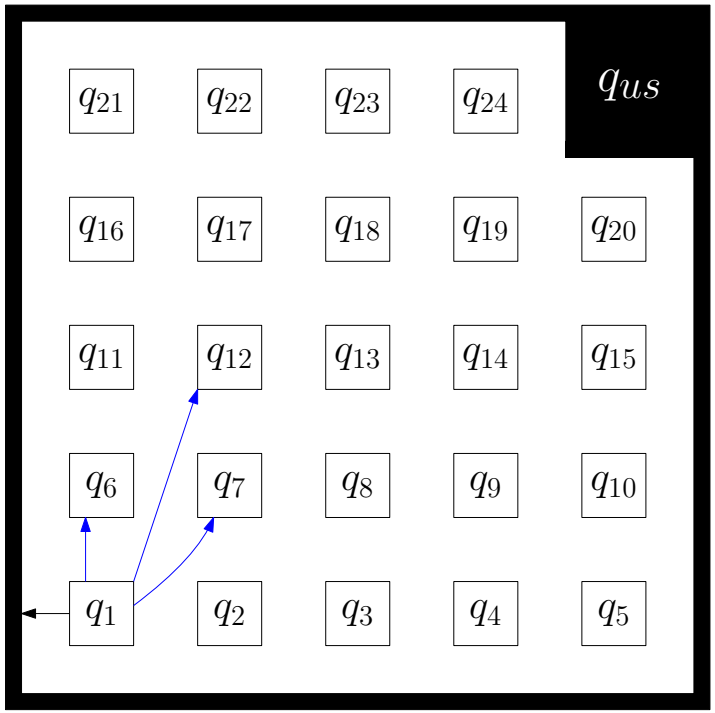


For every state $q \in Q$ and for every input $u \in U_\mu$ the transition

$(q, u_\mu, q') \in F \Leftrightarrow q' \cap \overline{\mathsf{R}}(\tau \mid q, u_\mu) \neq \emptyset$

Symbolic Model

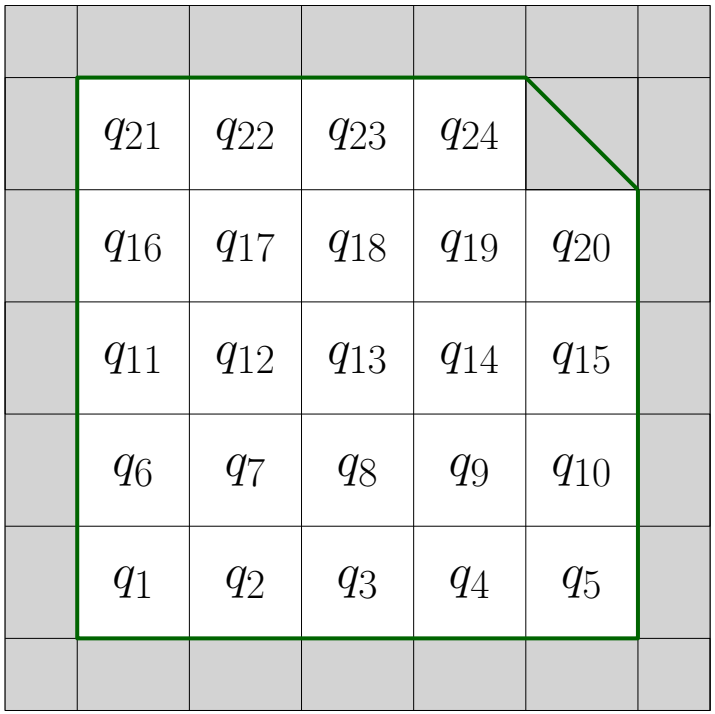


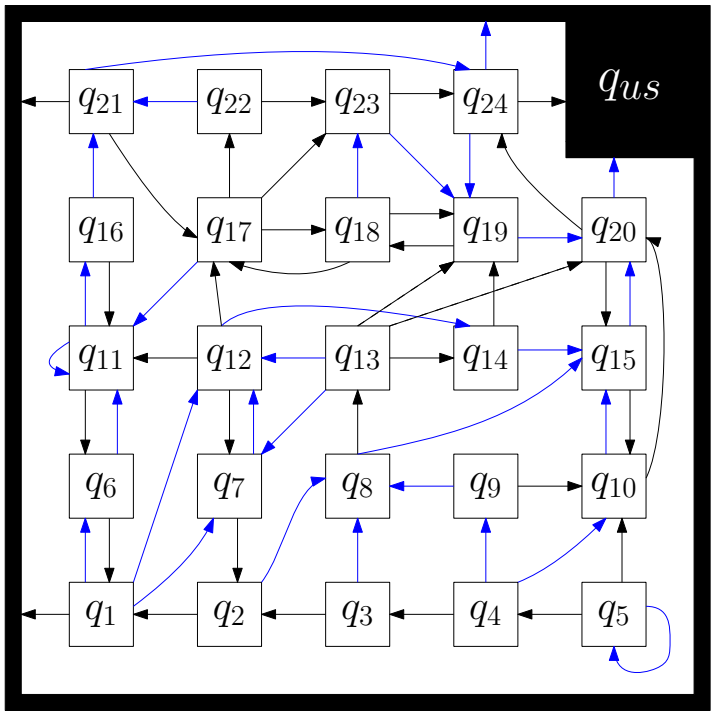


For every state $q \in Q$ and for every input $u \in U_\mu$ the transition

$$(q, u_\mu, q') \in F \Leftrightarrow q' \cap \overline{\mathsf{R}}(\tau \mid q, u_\mu) \neq \emptyset$$

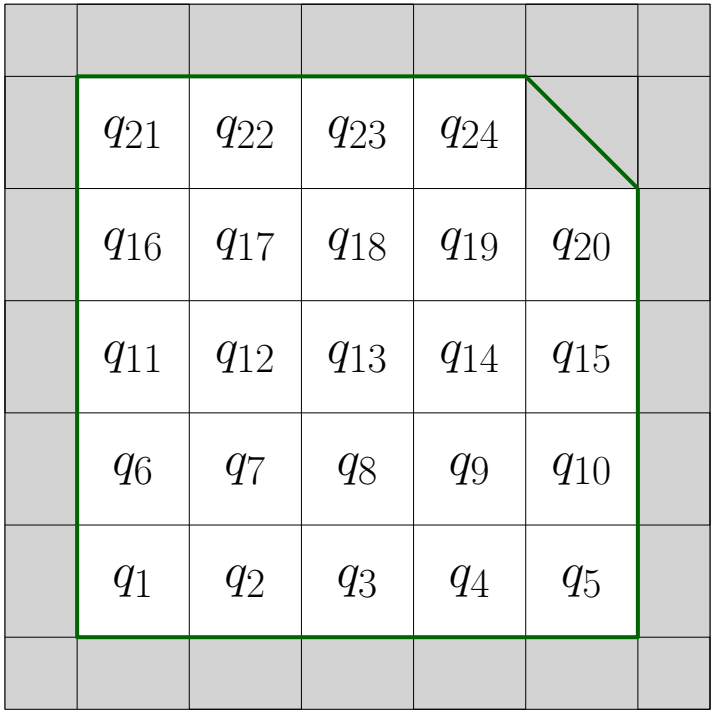
Symbolic Model

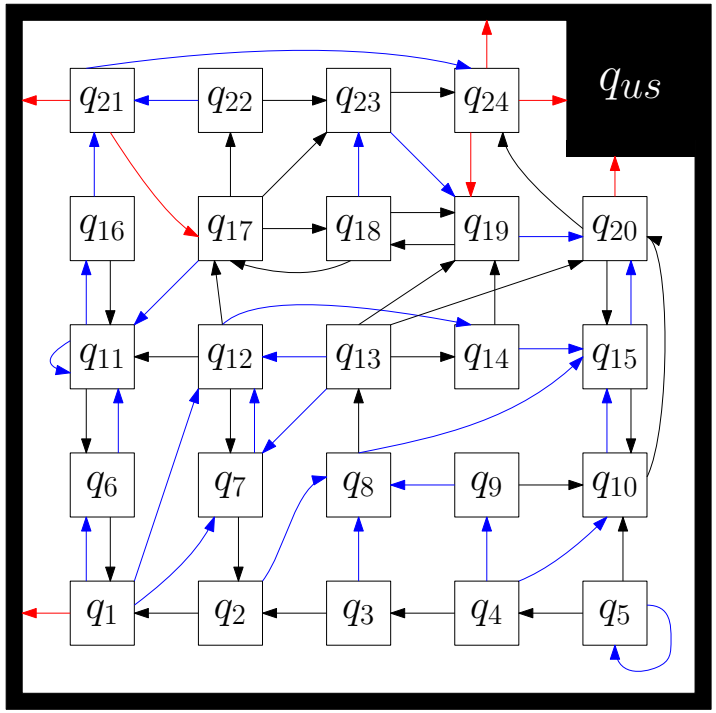




Finally we obtain a finite transition system over-approximating the behavior of the original plant.

Classical Synthesis and Control Refiment

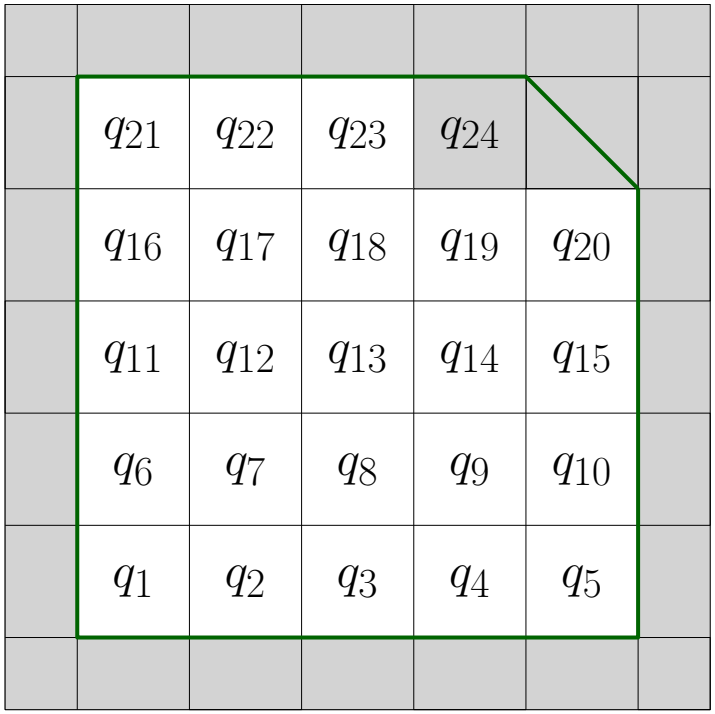


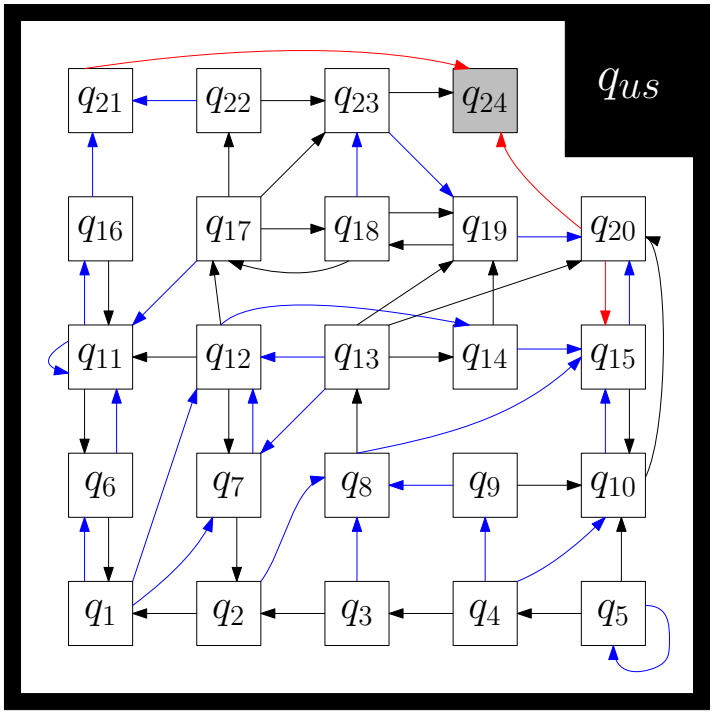


Classical synthesis algorithm is an iterative procedure removing unsafe actions[1] from the finite transition system until the system becomes unchanged.

[1] Tabuada, 2009. Verification and control of hybrid systems: a symbolic approach. Springer.

Classical Synthesis and Control Refiment

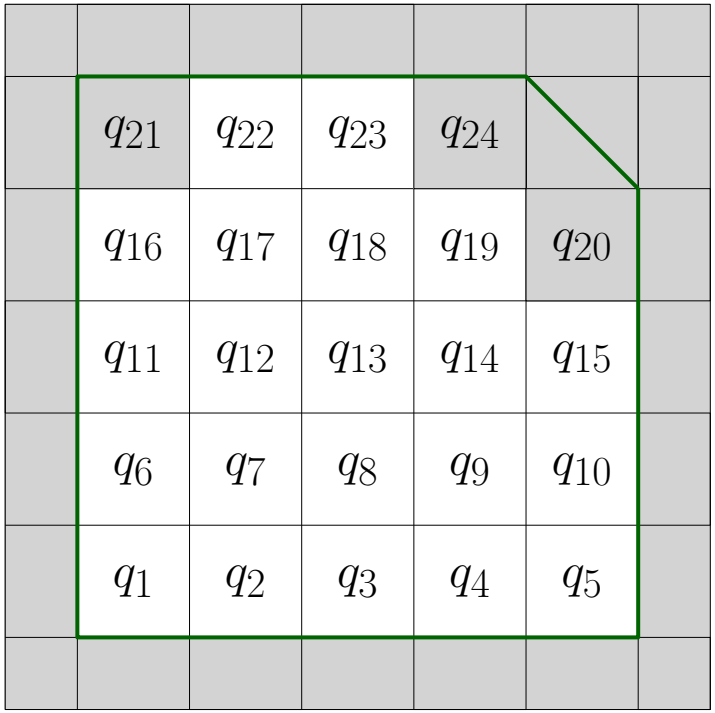


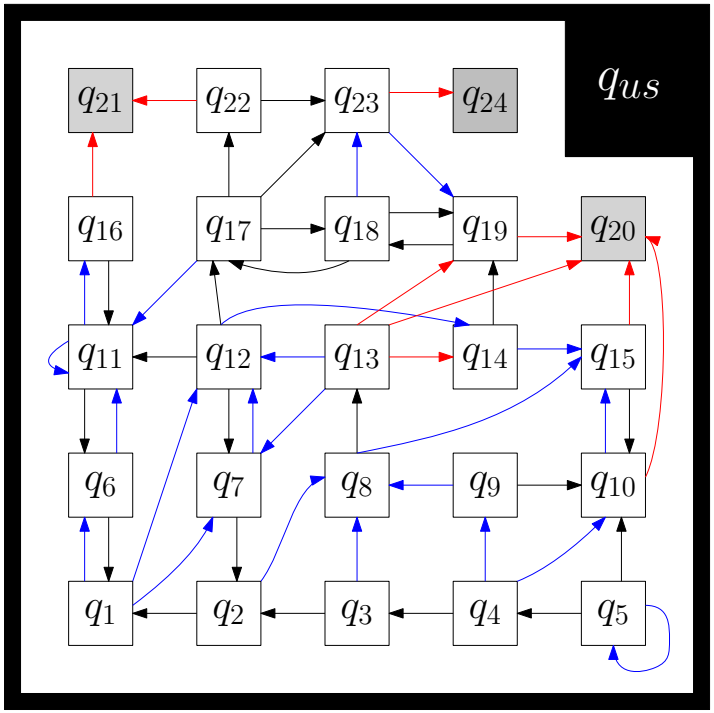


Classical synthesis algorithm is an iterative procedure removing unsafe actions[1] from the finite transition system until the system becomes unchanged.

[1] Tabuada, 2009. Verification and control of hybrid systems: a symbolic approach. Springer.

Classical Synthesis and Control Refiment

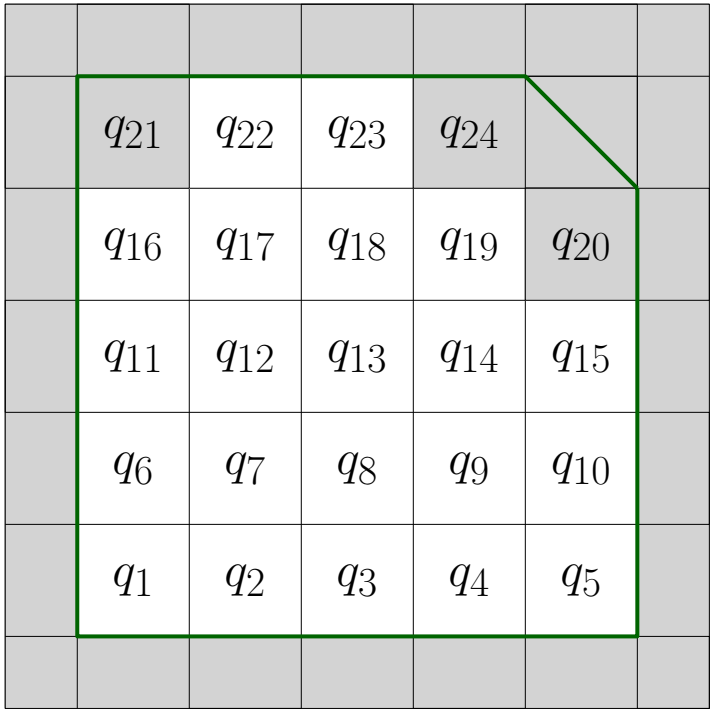


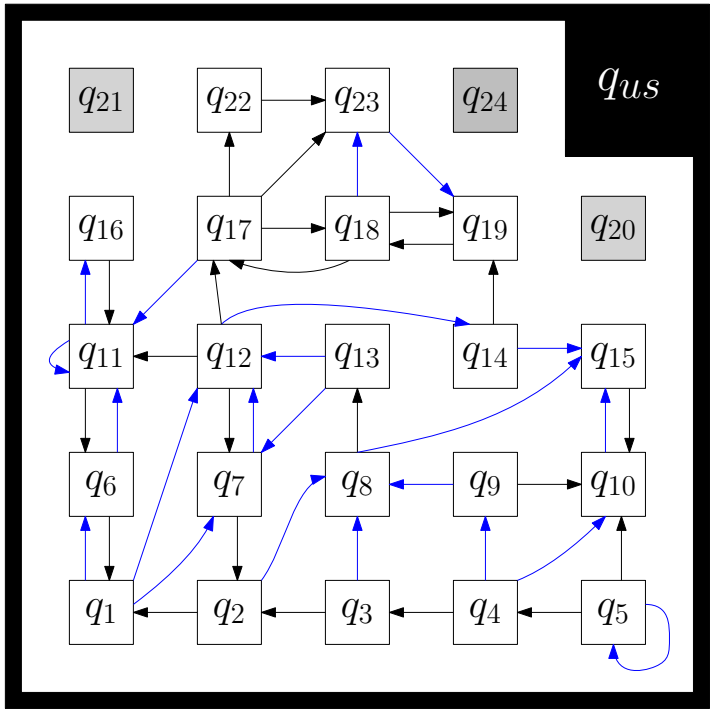


Classical synthesis algorithm is an iterative procedure removing unsafe actions[1] from the finite transition system until the system becomes unchanged.

[1] Tabuada, 2009. Verification and control of hybrid systems: a symbolic approach. Springer.

Classical Synthesis and Control Refiment

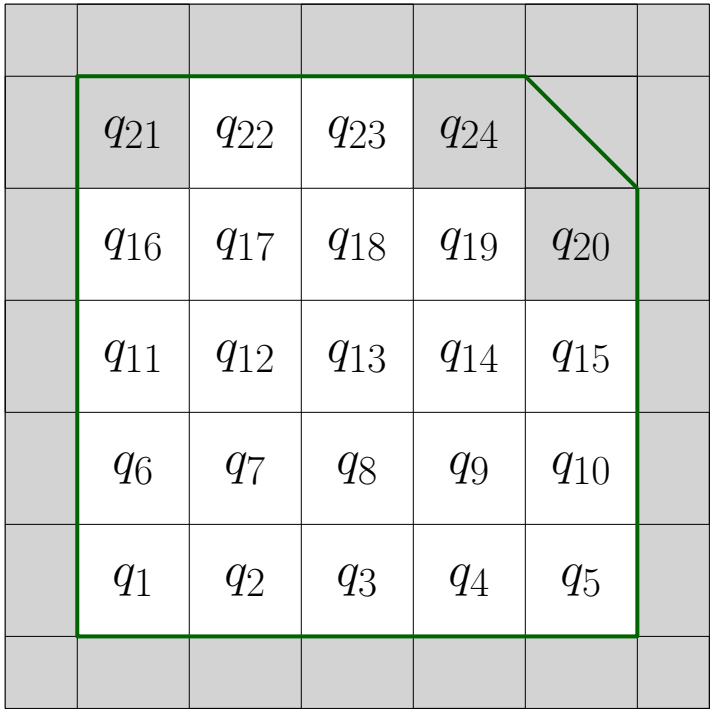


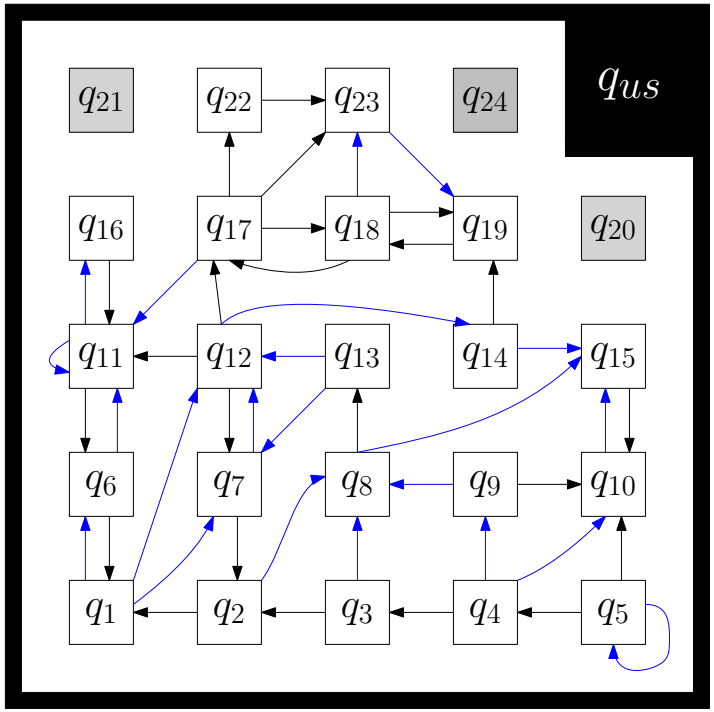


Classical synthesis algorithm is an iterative procedure removing unsafe actions[1] from the finite transition system until the system becomes unchanged.

[1] Tabuada, 2009. Verification and control of hybrid systems: a symbolic approach. Springer.

Classical Synthesis and Control Refiment

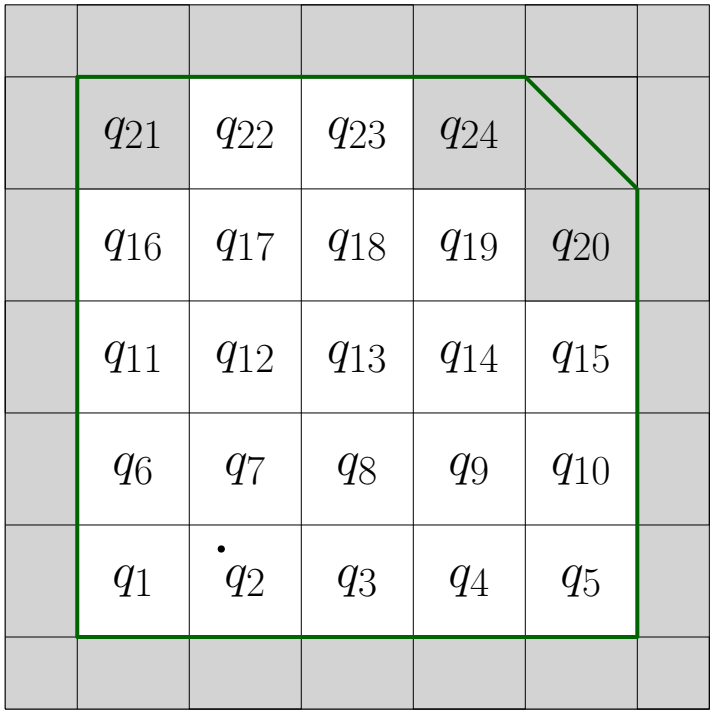


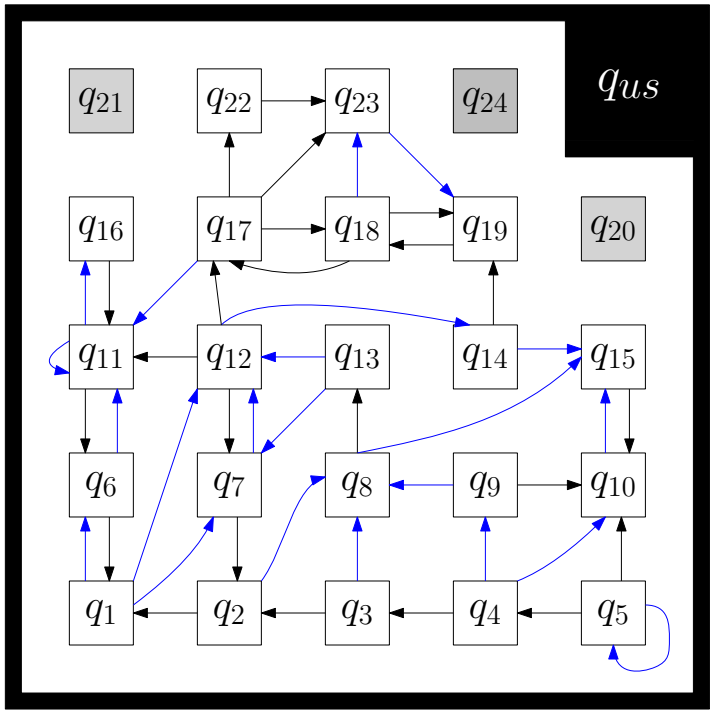


The safely controllable system $\Sigma_{C^*} = (\mathrm{Dom}(C^*), U, F_{C^*})$ is non-blocking and all its trajectories belong to the safe set Q_S.

[1] Tabuada, 2009. Verification and control of hybrid systems: a symbolic approach. Springer.

Classical Synthesis and Control Refiment

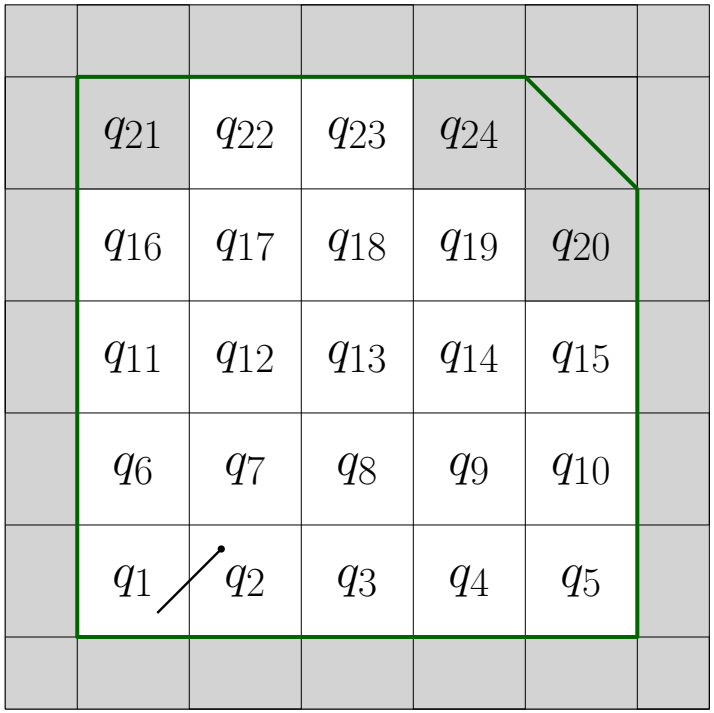


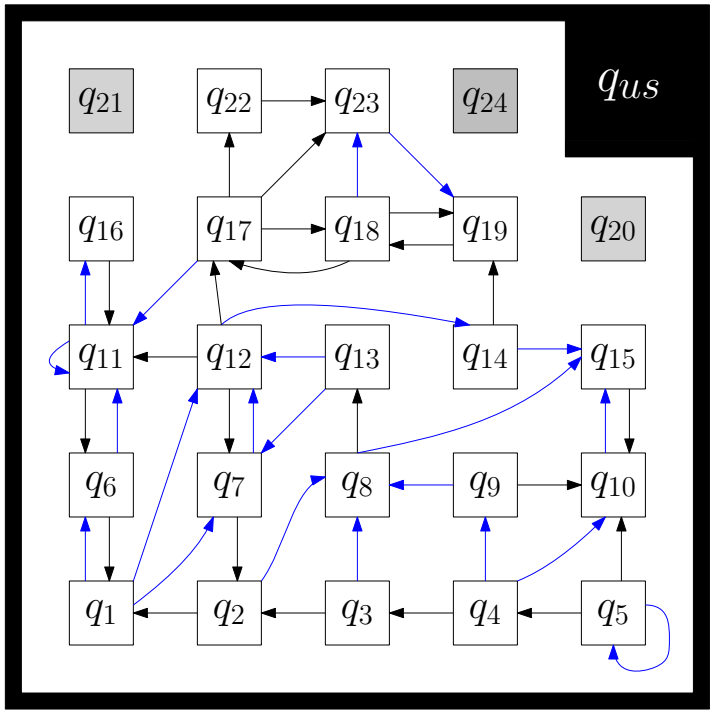


The safety controller for the original system is then implemented as a look-up table.

[1] Tabuada, 2009. Verification and control of hybrid systems: a symbolic approach. Springer.

Classical Synthesis and Control Refiment

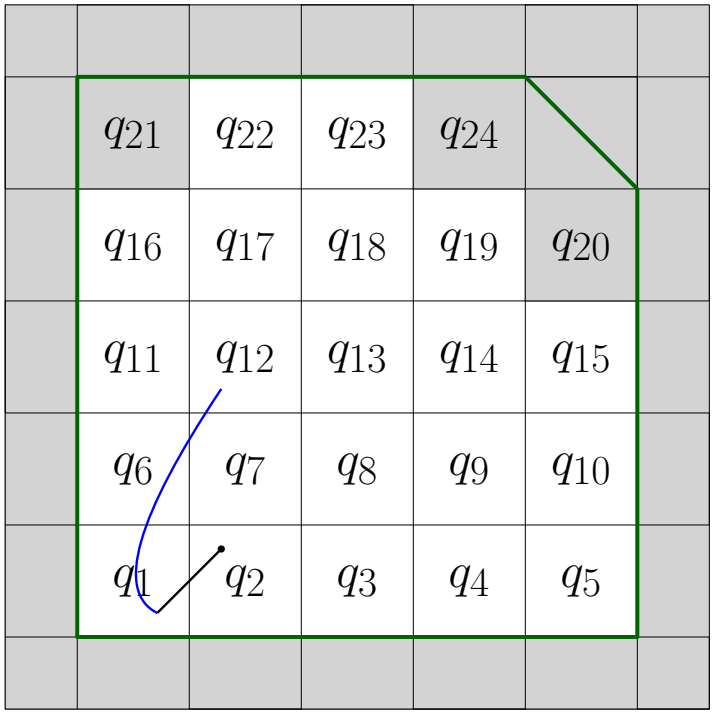


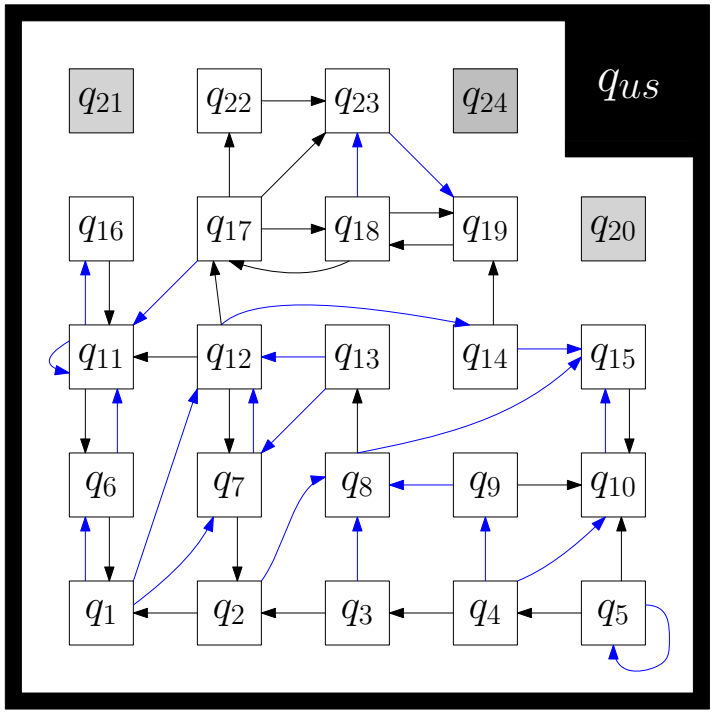


The safety controller for the original system is then implemented as a look-up table.

[1] Tabuada, 2009. Verification and control of hybrid systems: a symbolic approach. Springer.

Classical Synthesis and Control Refiment

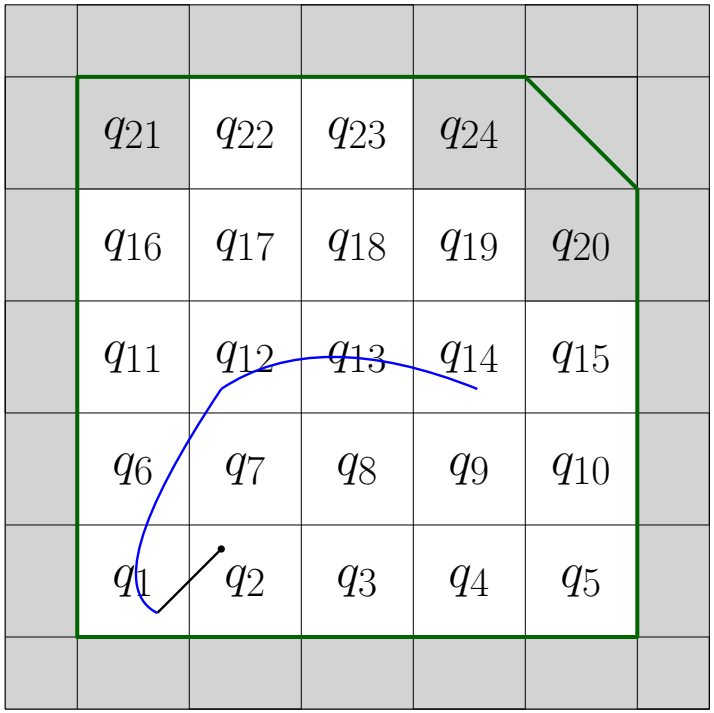


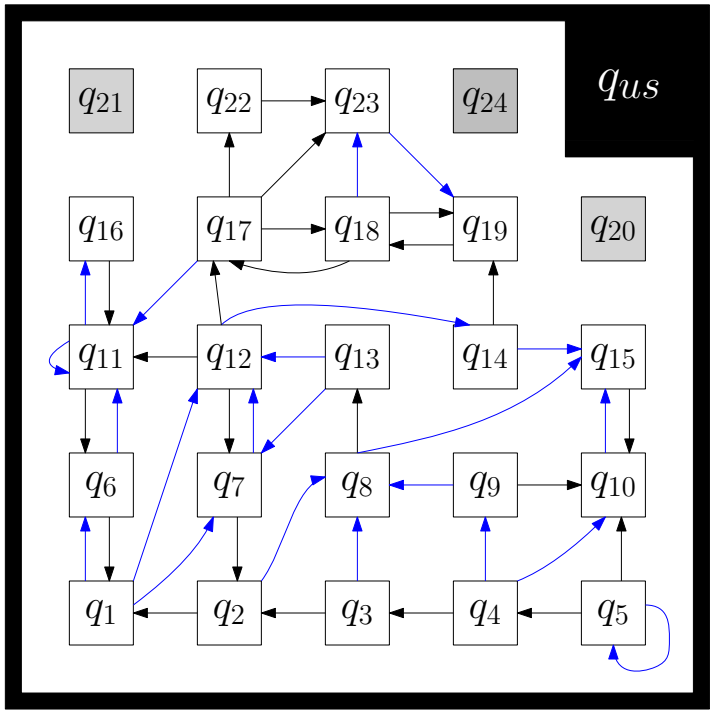


The safety controller for the original system is then implemented as a look-up table.

[1] Tabuada, 2009. Verification and control of hybrid systems: a symbolic approach. Springer.

Classical Synthesis and Control Refiment

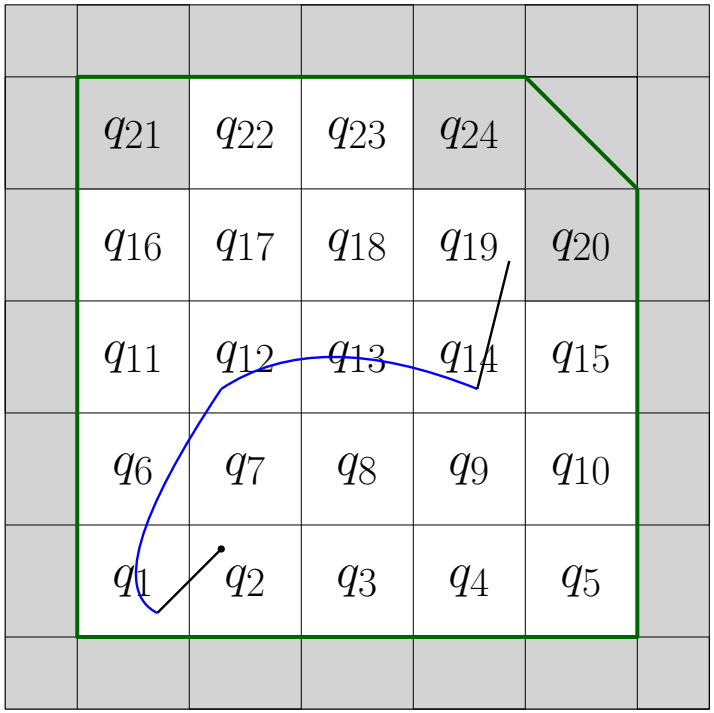


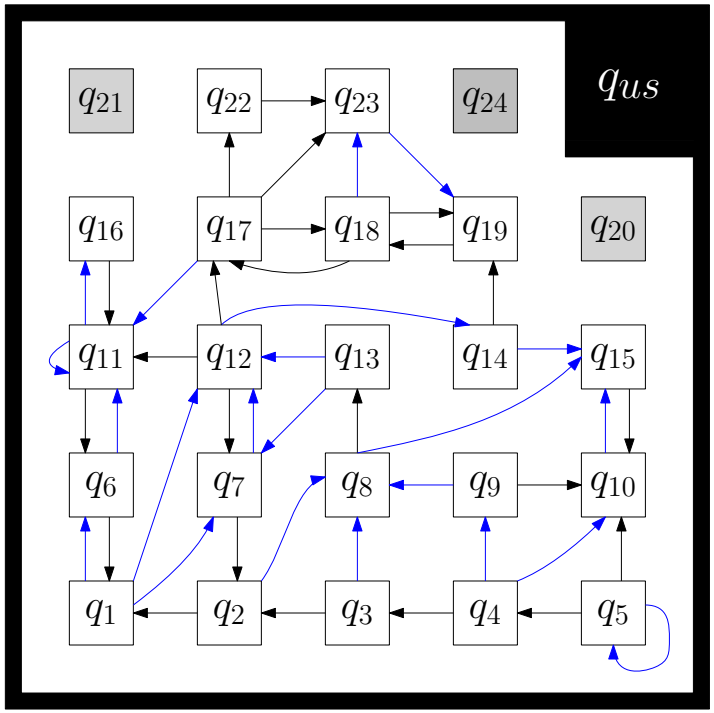


The safety controller for the original system is then implemented as a look-up table.

[1] Tabuada, 2009. Verification and control of hybrid systems: a symbolic approach. Springer.

Classical Synthesis and Control Refiment





The safety controller for the original system is then implemented as a look-up table.

[1] Tabuada, 2009. Verification and control of hybrid systems: a symbolic approach. Springer.

Drawbacks of Classical Synthesis Procedure

Classical Synthesis Approach

- The symbolic model is required to be precomputed before the synthesis.
- Brute-force exploration of the finite transition system.

Drawbacks of Classical Synthesis Procedure

Classical Synthesis Approach

- The symbolic model is required to be precomputed before the synthesis.
- Brute-force exploration of the finite transition system.

Lazy Synthesis Approaches

- The abstraction is computed on-the-fly, during the synthesis procedure.
- Only essential for synthesis part of abstraction is explored.

A. Girard, G. Gössler, and S. Moueli, (2016). Safety controller synthesis for incrementally stable switched systems using Multi-scale symbolic models. IEEE Transactions on Automatic Control.

O. Hussien and P. Tabuada. (2018). Lazy controller synthesis using three-valued abstractions for safety and reachability specifications. CDC.

K. Hsu, R. Majumdar, K. Mallik, and A.K. Schmuck. (2019). Lazy abstraction-based controller synthesis. Automated Technology for Verification and Analysis.

Lazy Synthesis Approaches

Lazy synthesis approach for multi-scale symbolic models.

- E. Ivanova and A. Girard (2020). Lazy safety controller synthesis with multi-scale adaptive-sampling abstractions of nonlinear systems. IFAC WC.

Lazy exploration restricted to boundary states.

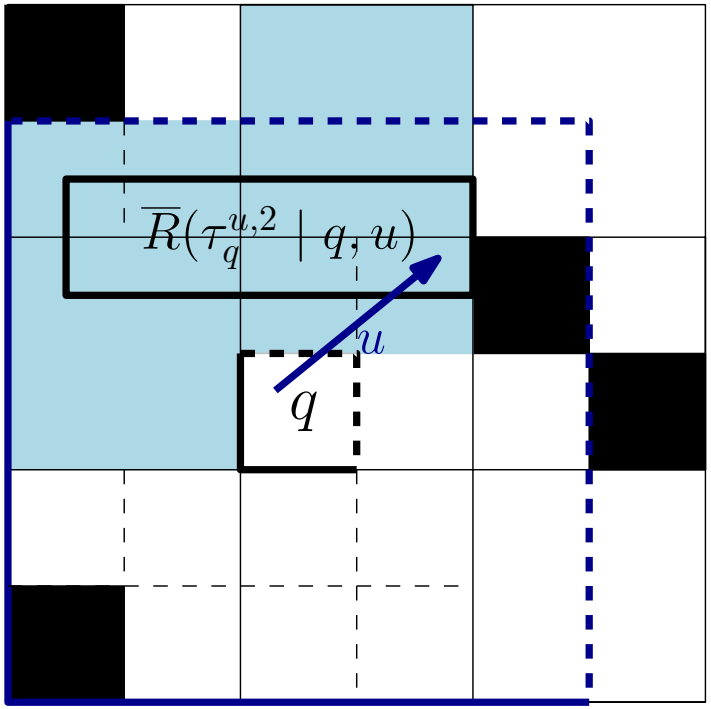
- E. Ivanova and A. Girard (2021). Lazy Symbolic Controller for Continuous-Time Systems Based on Safe Set Boundary Exploration, IFAC ADHS.

Lazy synthesis approach for monotone transition systems.

- E. Ivanova, A. Saoud, and A. Girard (2021). Lazy Controller Synthesis for Monotone Transition Systems and Directed Safety Specifications, Automatica.
- A. Saoud , E. Ivanova and A. Girard (2019). Efficient Synthesis for Monotone Transition Systems and Directed Safety Specifications. IEEE CDC.

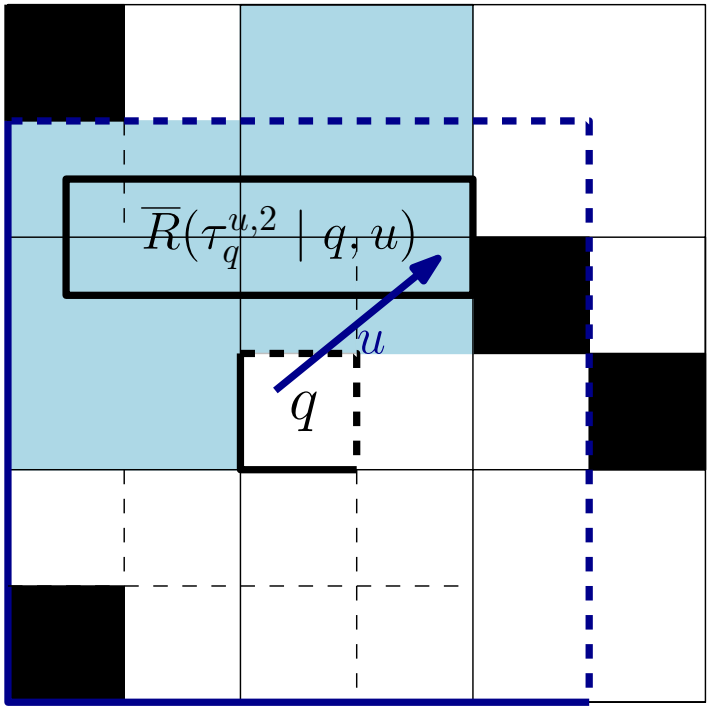
Lazy synthesis approach for multi-scale symbolic models.

Lazy Synthesis Approach for Multi-scale Symbolic Models.



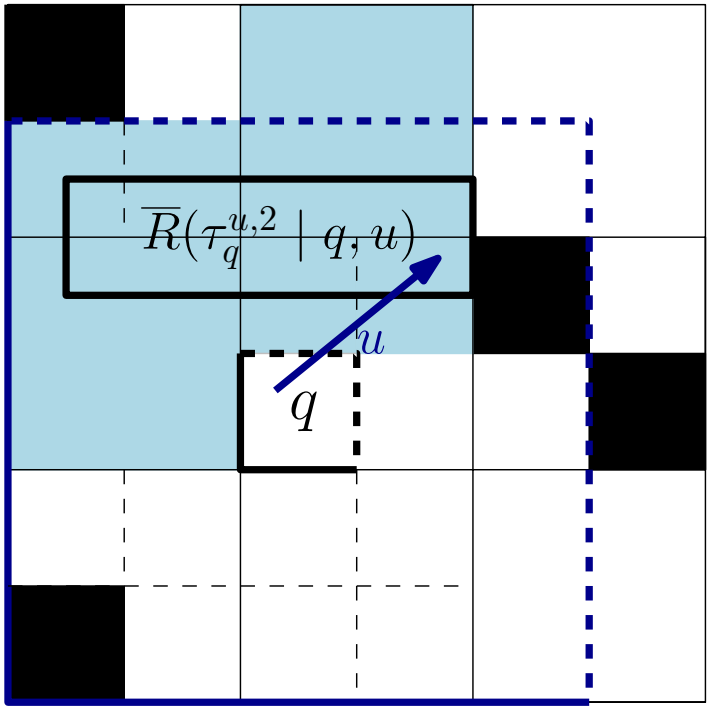
- Incremental forward exploration of the symbolic dynamics, allowing us to restrict the controller synthesis computations to reachable states only

Lazy Synthesis Approach for Multi-scale Symbolic Models.



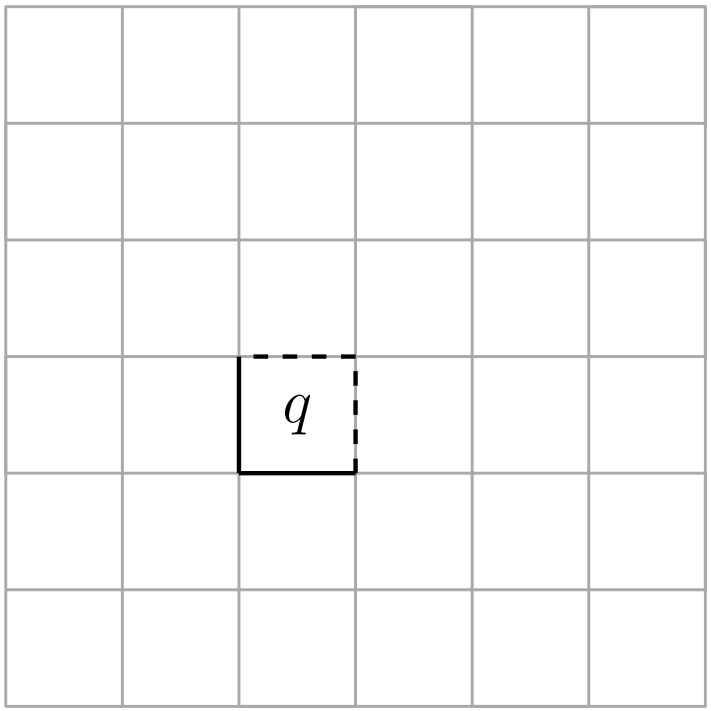
- Incremental forward exploration of the symbolic dynamics, allowing us to restrict the controller synthesis computations to reachable states only
- Adaptive grid: start with a coarse grid and locally refine it in case of need.

Lazy Synthesis Approach for Multi-scale Symbolic Models.



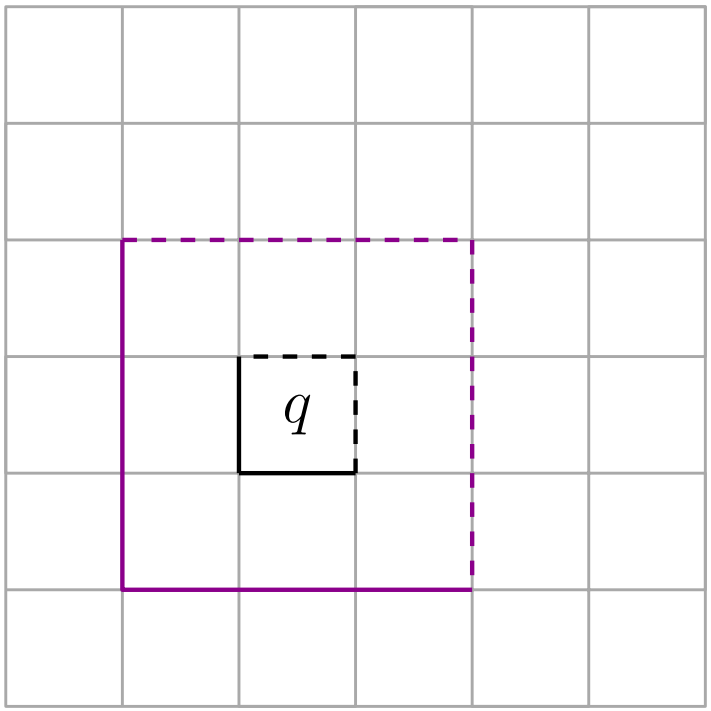
- Incremental forward exploration of the symbolic dynamics, allowing us to restrict the controller synthesis computations to reachable states only
- Adaptive grid: start with a coarse grid and locally refine it in case of need.
- Prioritize inputs with longer duration.

Lazy Synthesis Approach for Multi-scale Symbolic Models.



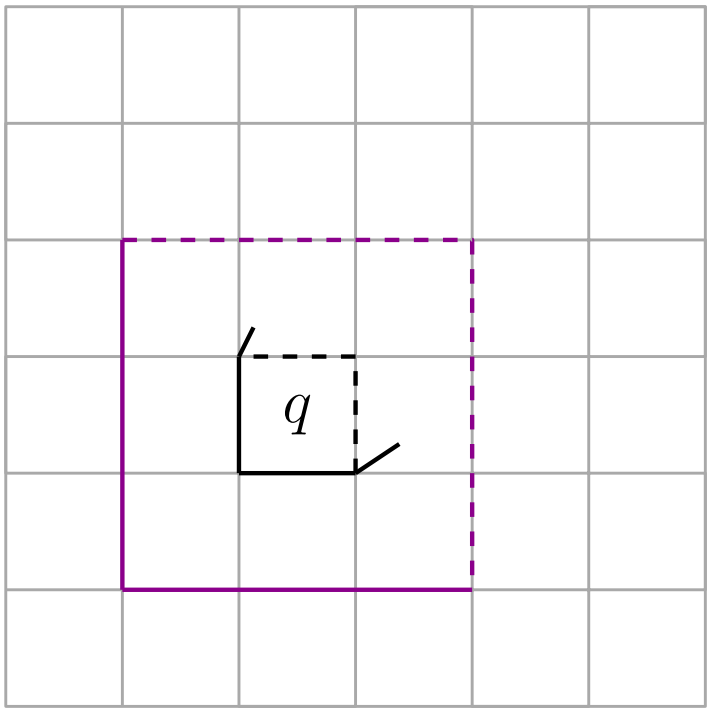
- Incremental forward exploration of the symbolic dynamics, allowing us to restrict the controller synthesis computations to reachable states only
- Adaptive grid: start with a coarse grid and locally refine it in case of need.
- Prioritize inputs with longer duration.
- Transition duration is constrained by intervals that must contain the reachable set.

Lazy Synthesis Approach for Multi-scale Symbolic Models.



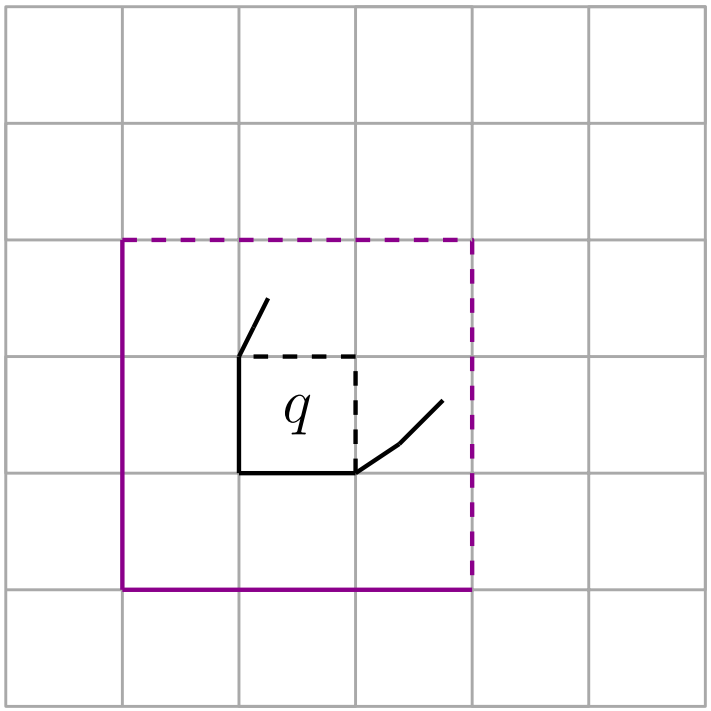
- Incremental forward exploration of the symbolic dynamics, allowing us to restrict the controller synthesis computations to reachable states only
- Adaptive grid: start with a coarse grid and locally refine it in case of need.
- Prioritize inputs with longer duration.
- Transition duration is constrained by intervals that must contain the reachable set.

Lazy Synthesis Approach for Multi-scale Symbolic Models.



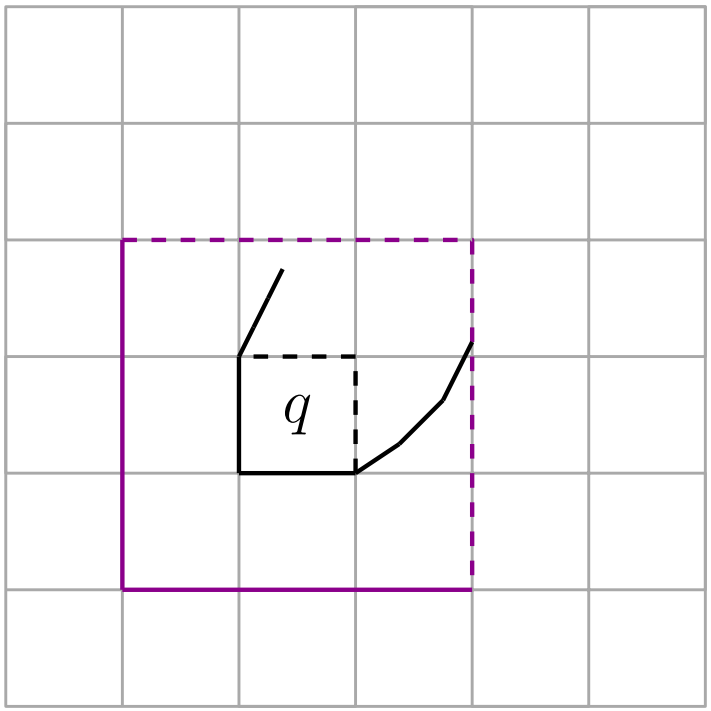
- Incremental forward exploration of the symbolic dynamics, allowing us to restrict the controller synthesis computations to reachable states only
- Adaptive grid: start with a coarse grid and locally refine it in case of need.
- Prioritize inputs with longer duration.
- Transition duration is constrained by intervals that must contain the reachable set.

Lazy Synthesis Approach for Multi-scale Symbolic Models.



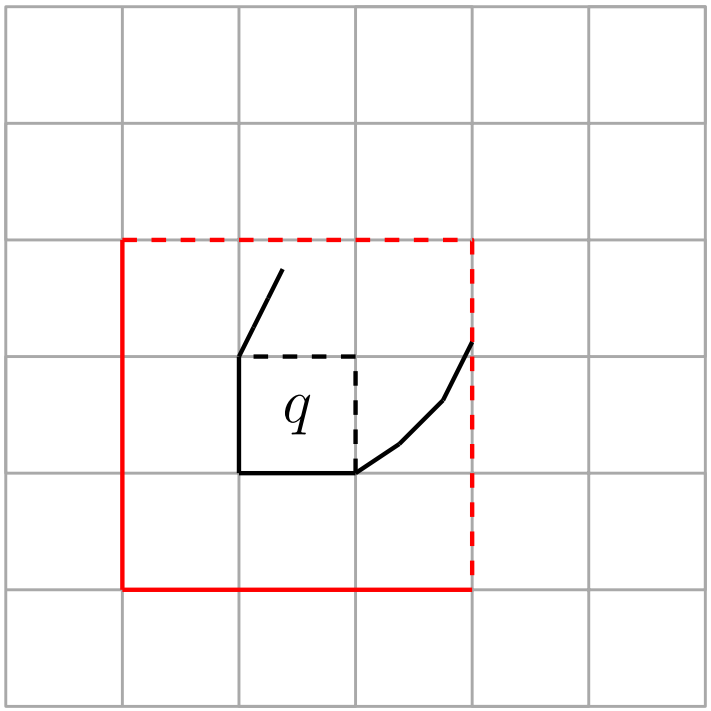
- Incremental forward exploration of the symbolic dynamics, allowing us to restrict the controller synthesis computations to reachable states only
- Adaptive grid: start with a coarse grid and locally refine it in case of need.
- Prioritize inputs with longer duration.
- Transition duration is constrained by intervals that must contain the reachable set.

Lazy Synthesis Approach for Multi-scale Symbolic Models.



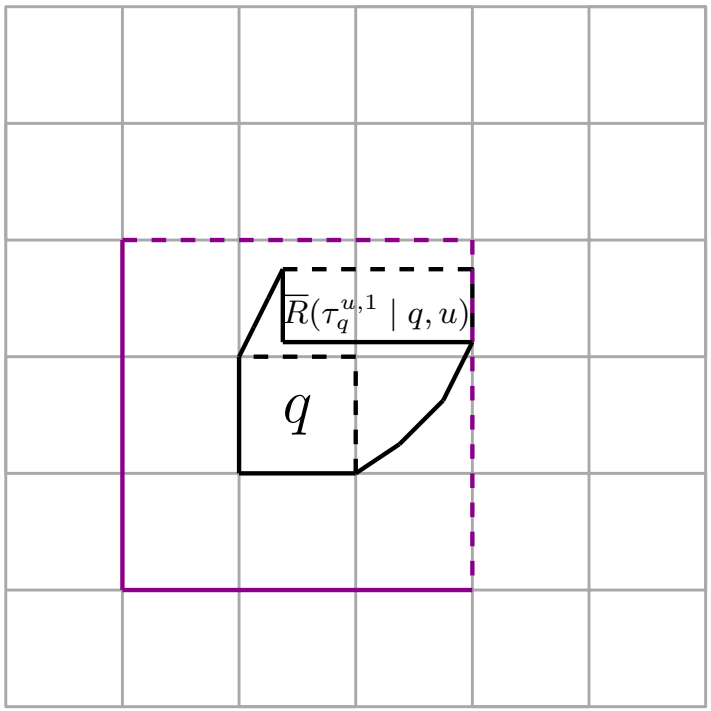
- Incremental forward exploration of the symbolic dynamics, allowing us to restrict the controller synthesis computations to reachable states only
- Adaptive grid: start with a coarse grid and locally refine it in case of need.
- Prioritize inputs with longer duration.
- Transition duration is constrained by intervals that must contain the reachable set.

Lazy Synthesis Approach for Multi-scale Symbolic Models.



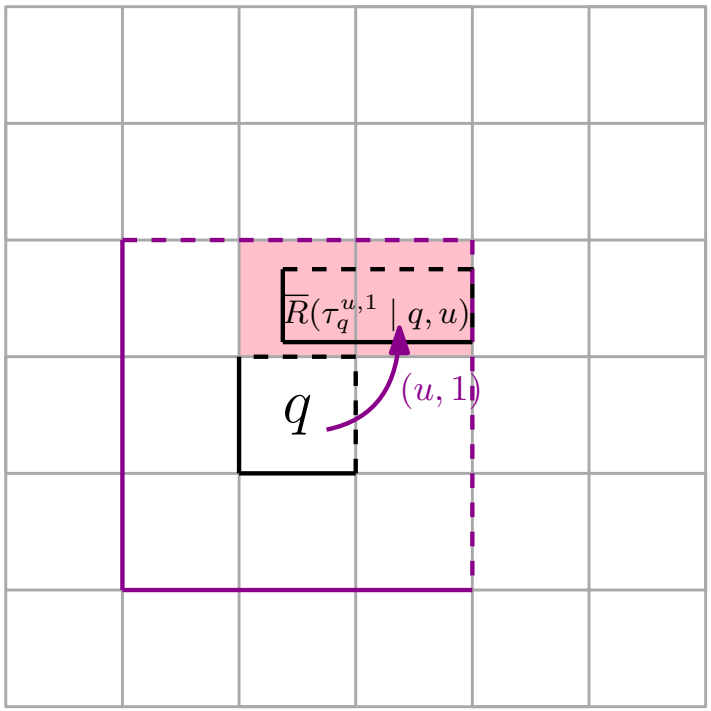
- Incremental forward exploration of the symbolic dynamics, allowing us to restrict the controller synthesis computations to reachable states only
- Adaptive grid: start with a coarse grid and locally refine it in case of need.
- Prioritize inputs with longer duration.
- Transition duration is constrained by intervals that must contain the reachable set.

Lazy Synthesis Approach for Multi-scale Symbolic Models.



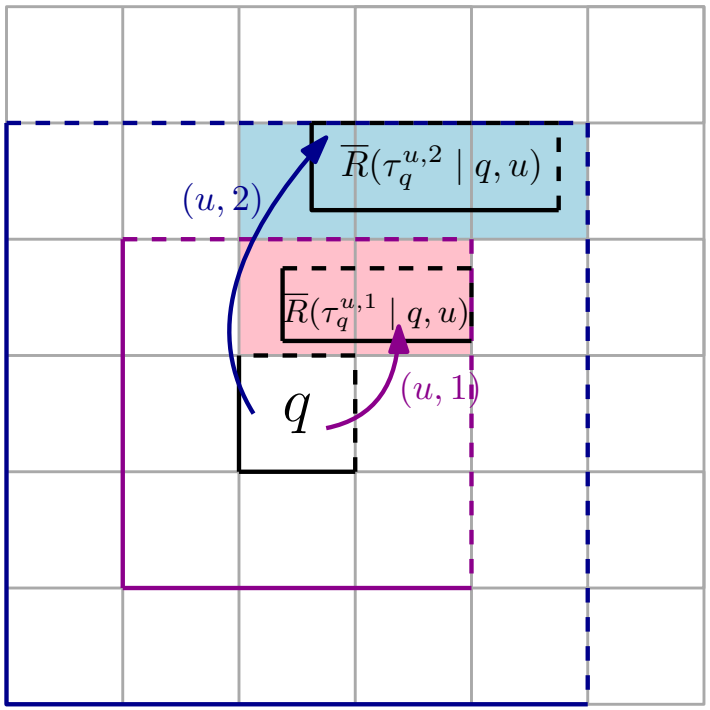
- Incremental forward exploration of the symbolic dynamics, allowing us to restrict the controller synthesis computations to reachable states only
- Adaptive grid: start with a coarse grid and locally refine it in case of need.
- Prioritize inputs with longer duration.
- Transition duration is constrained by intervals that must contain the reachable set.

Lazy Synthesis Approach for Multi-scale Symbolic Models.



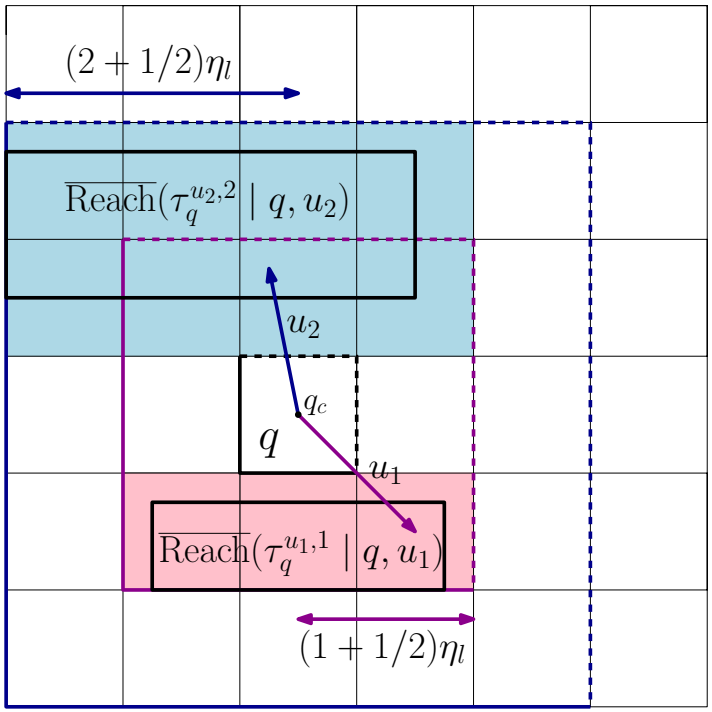
- Incremental forward exploration of the symbolic dynamics, allowing us to restrict the controller synthesis computations to reachable states only
- Adaptive grid: start with a coarse grid and locally refine it in case of need.
- Prioritize inputs with longer duration.
- Transition duration is constrained by intervals that must contain the reachable set.

Lazy Synthesis Approach for Multi-scale Symbolic Models.



- Incremental forward exploration of the symbolic dynamics, allowing us to restrict the controller synthesis computations to reachable states only
- Adaptive grid: start with a coarse grid and locally refine it in case of need.
- Prioritize inputs with longer duration.
- Transition duration is constrained by intervals that must contain the reachable set.

Lazy Synthesis Approach for Multi-scale Symbolic Models.



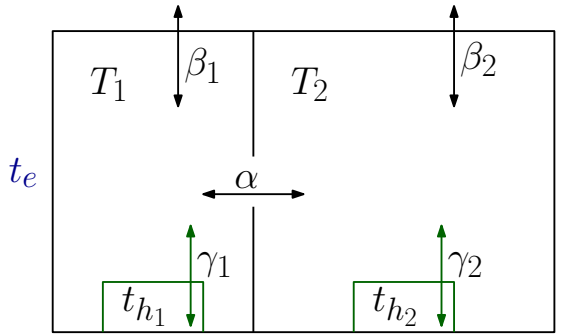
- Incremental forward exploration of the symbolic dynamics, allowing us to restrict the controller synthesis computations to reachable states only
- Adaptive grid: start with a coarse grid and locally refine it in case of need.
- Prioritize inputs with longer duration.
- Transition duration is constrained by intervals that must contain the reachable set.

Temperature Regulation in Smart Buildings

System dynamics

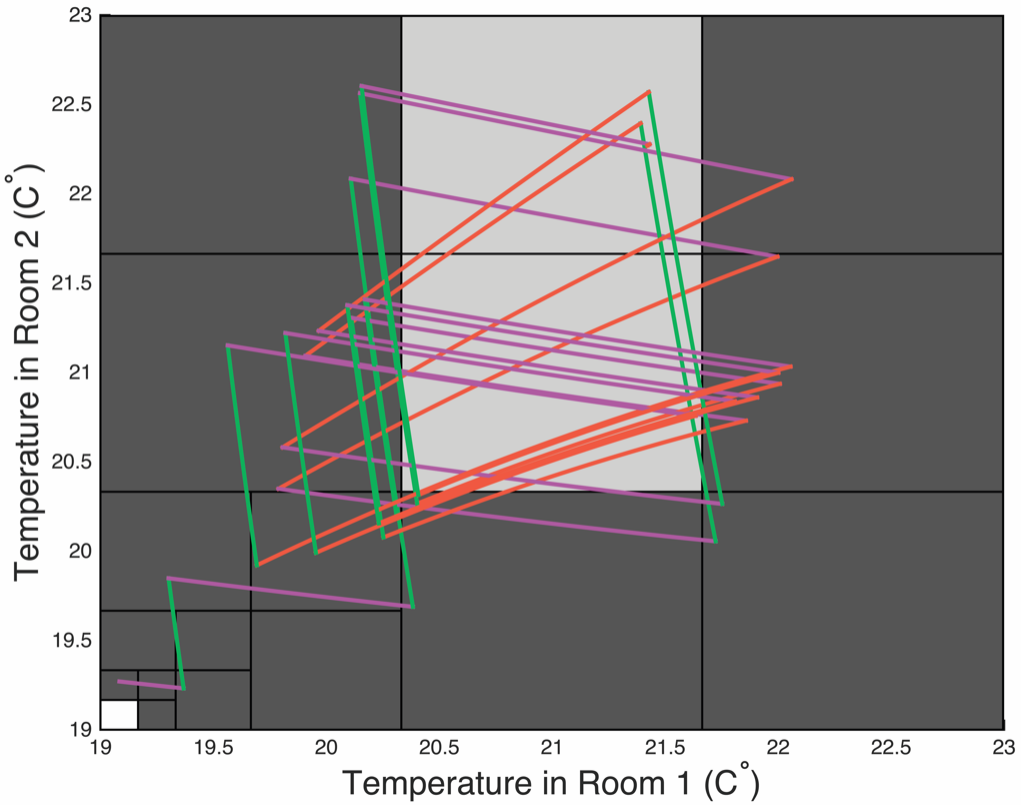
$$\dot{T}_1 = \alpha(T_2 - T_1) + \beta_1(t_e - T_1) + \gamma_1(t_{h_1} - T_1)u_1$$

$$\dot{T}_2 = \alpha(T_1 - T_2) + \beta_2(t_e - T_2) + \gamma_2(t_{h_2} - T_2)u_2$$



- Disturbance: $t_e \in [-10, 10]\ C^\circ$.
- Control: $u = (u_1, u_2) \in \{(0,1), (1,0), (0,0)\}$.
- Specification: $T_1 \in [19, 23]\ C^\circ$ and $T_2 \in [19, 23]\ C^\circ$.

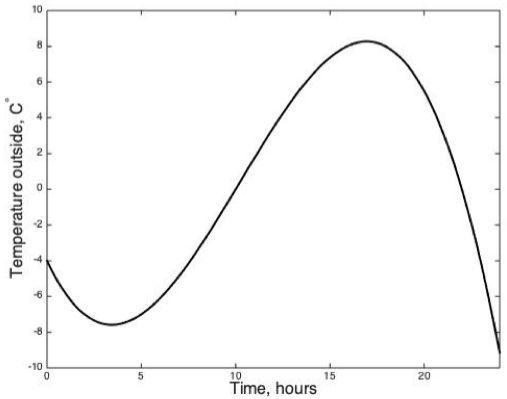
Temperature Regulation in Smart Buildings. Simulations Results



only 1st room heater works

only 2nd room heater works

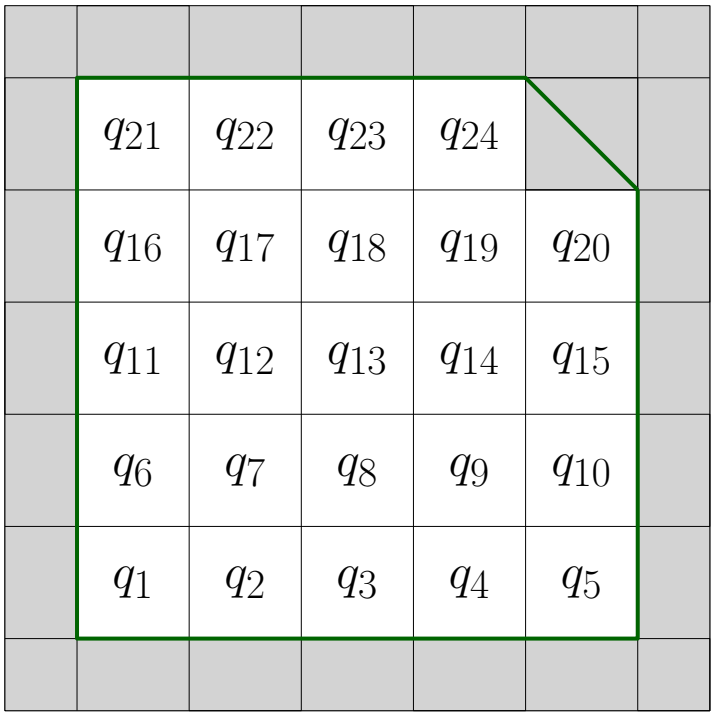
both heaters are turn off

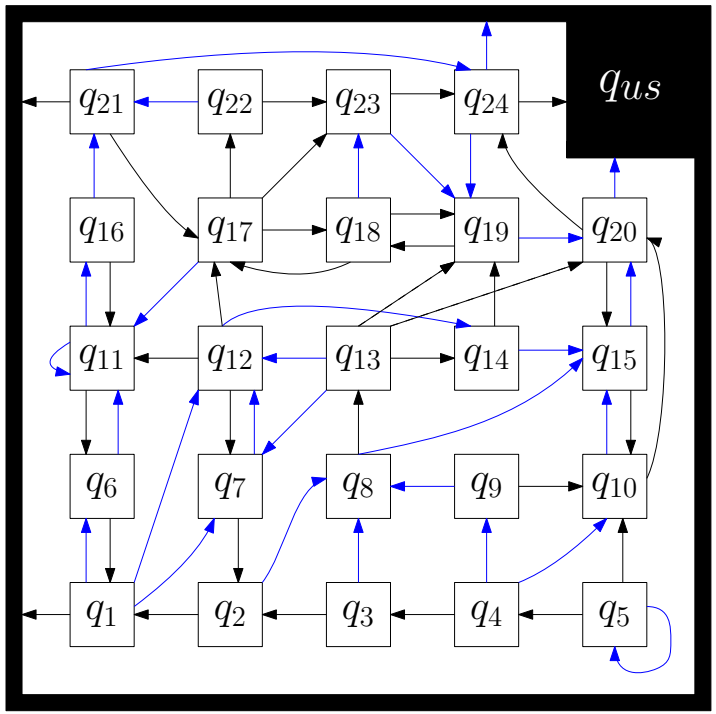


Grid	Number of states	Time	Cont. Ratio
Adaptive grid	18	7 s	98%
Coarsest grid	9	5 s	89%
Finest grid	625	50 s	98%

Lazy exploration restricted to boundary states.

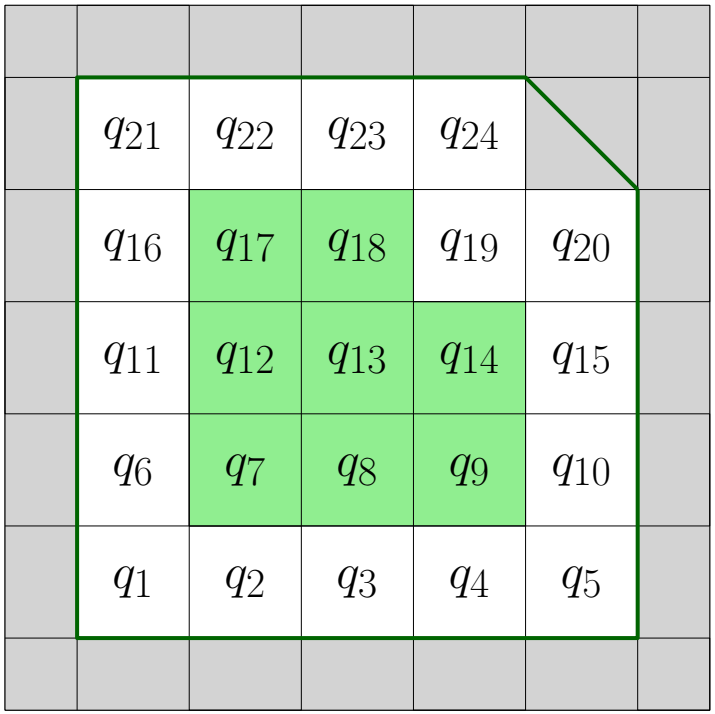
Lazy Exploration Restricted to Boundary States.

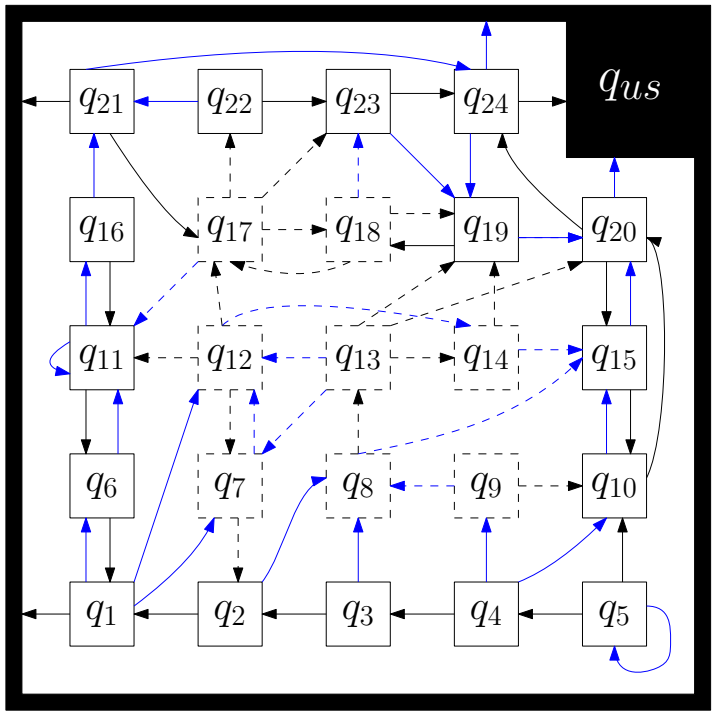




Let us iteratively explore states on the boundary of controllable domain while avoiding exploration of internal states.

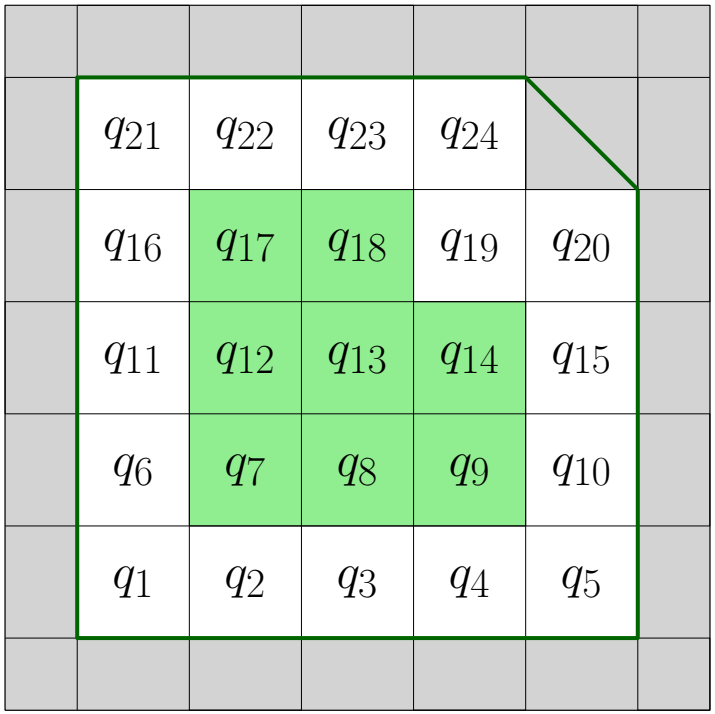
Lazy Exploration Restricted to Boundary States.

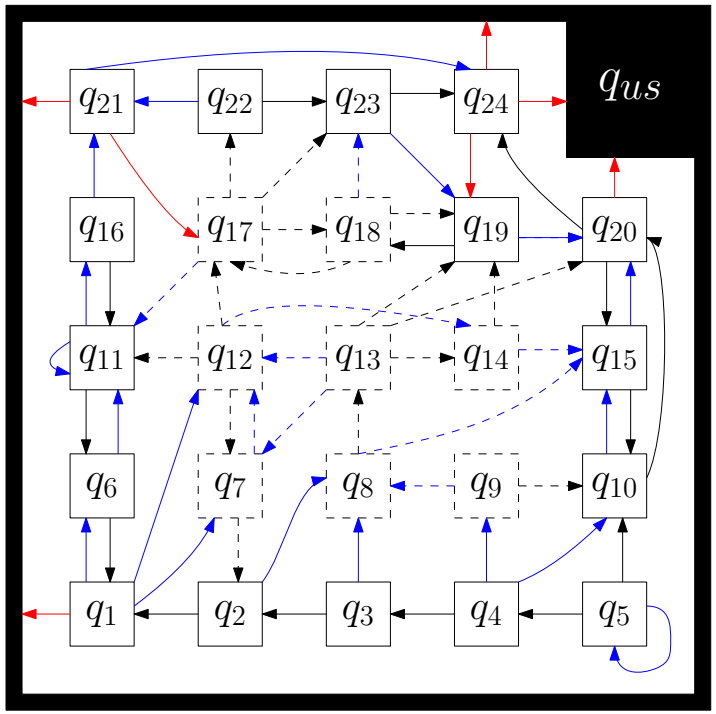




Let us iteratively explore states on the boundary of controllable domain while avoiding exploration of internal states.

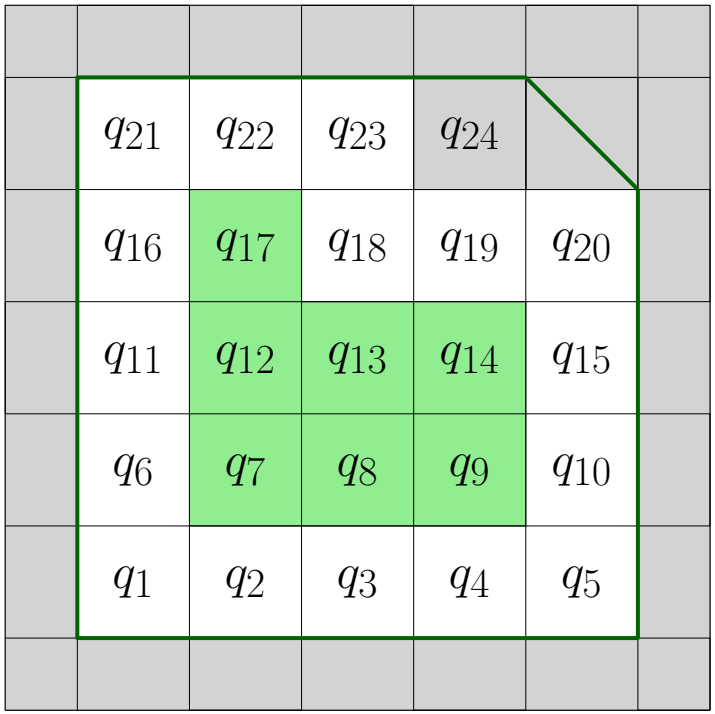
Lazy Exploration Restricted to Boundary States.

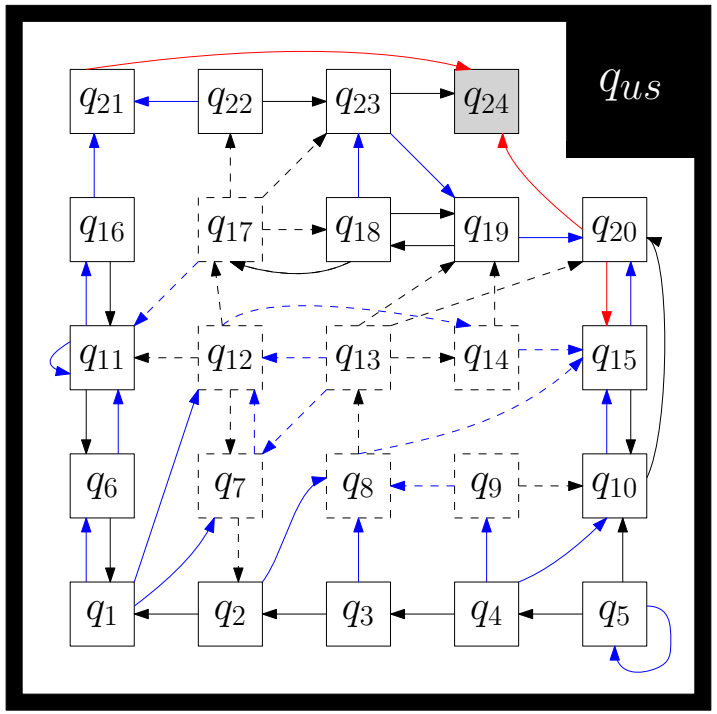




Let us iteratively explore states on the boundary of controllable domain while avoiding exploration of internal states.

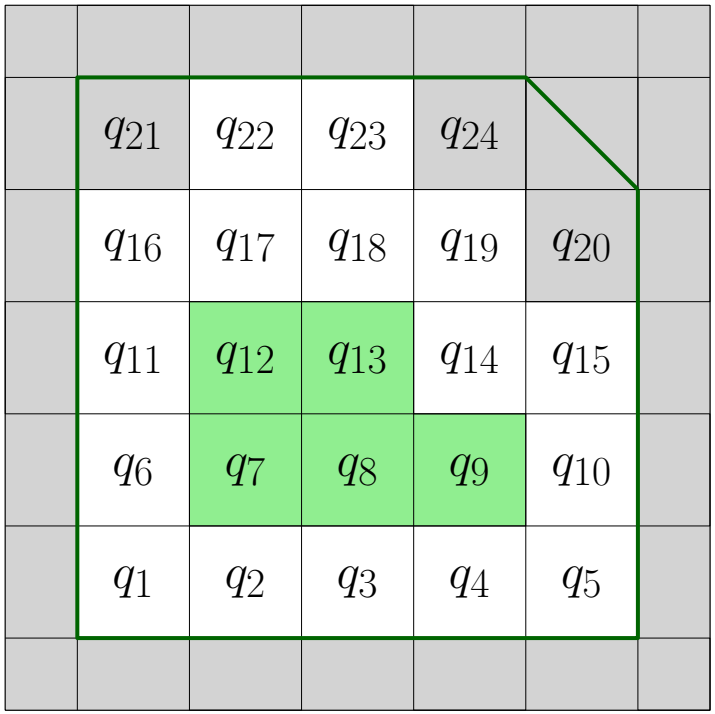
Lazy Exploration Restricted to Boundary States.

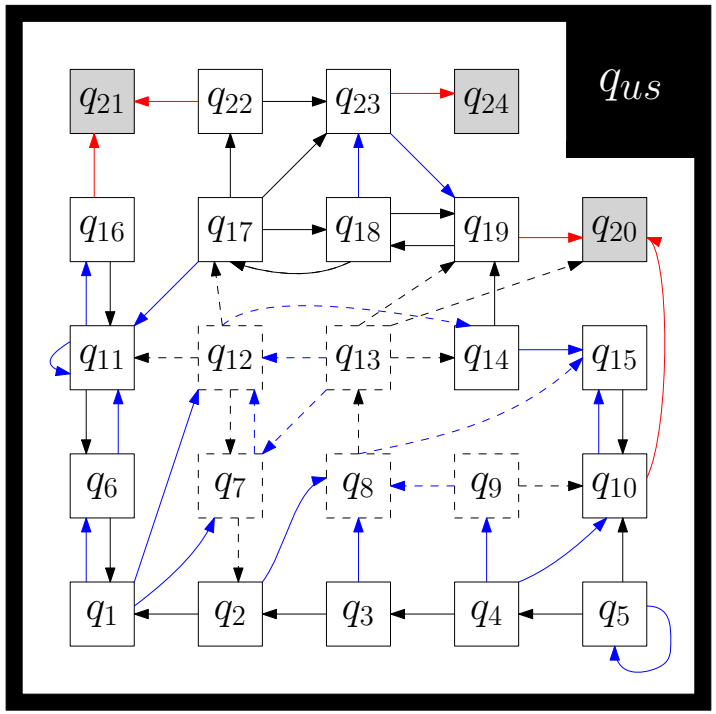




Let us iteratively explore states on the boundary of controllable domain while avoiding exploration of internal states.

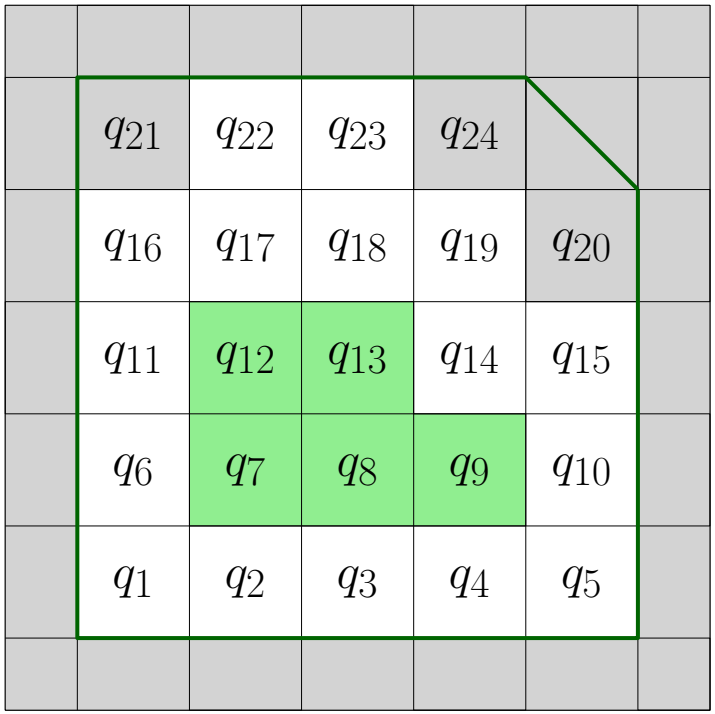
Lazy Exploration Restricted to Boundary States.

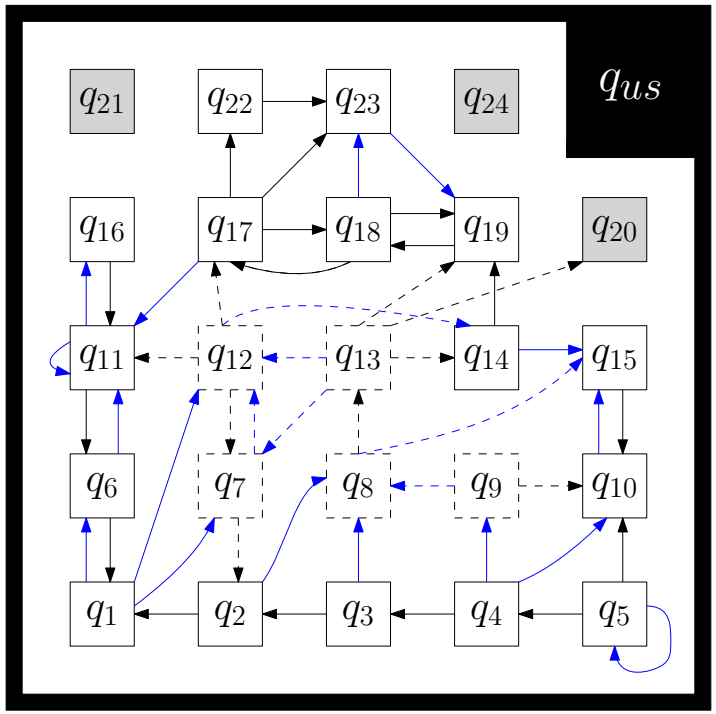




Let us iteratively explore states on the boundary of controllable domain while avoiding exploration of internal states.

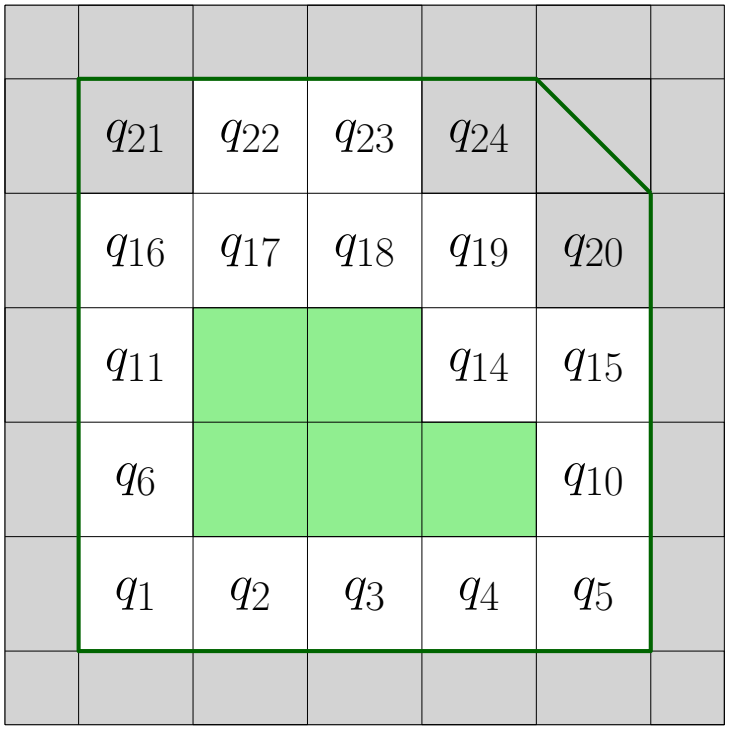
Lazy Exploration Restricted to Boundary States.

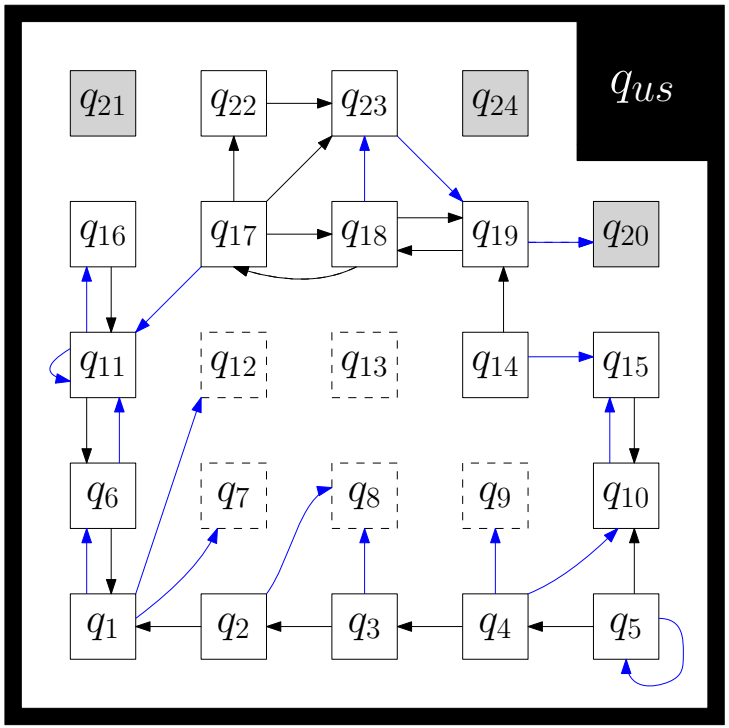




Let us iteratively explore states on the boundary of controllable domain while avoiding exploration of internal states.

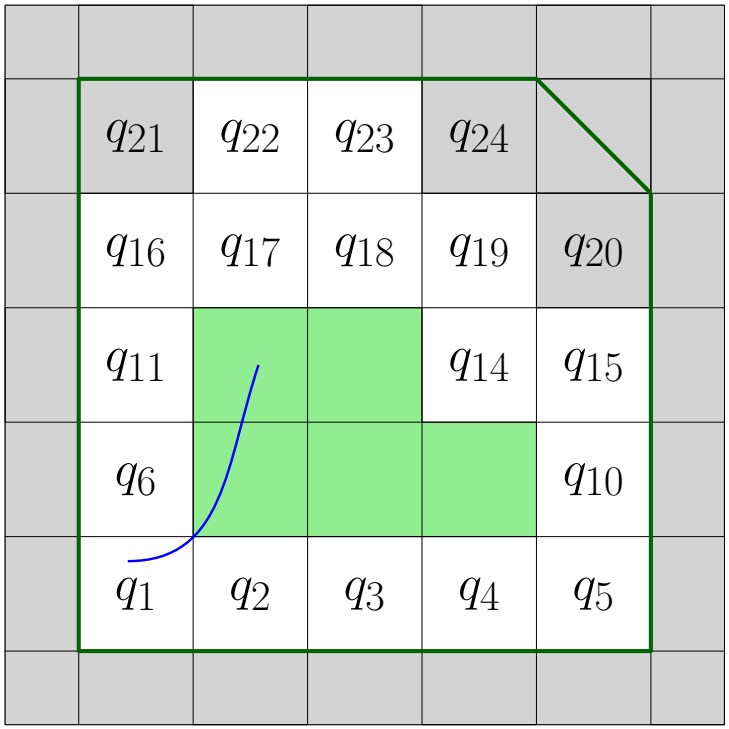
Lazy Exploration Restricted to Boundary States. Control Refinement

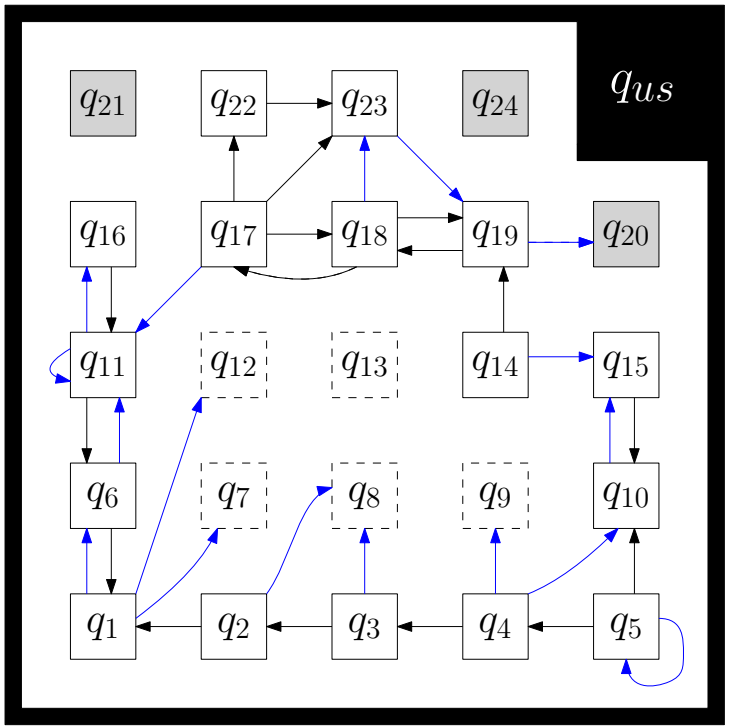




With a novel refinement scheme the abstract controller pushes the system from a boundary state back to the interior, while for inner states any input from U is valid.

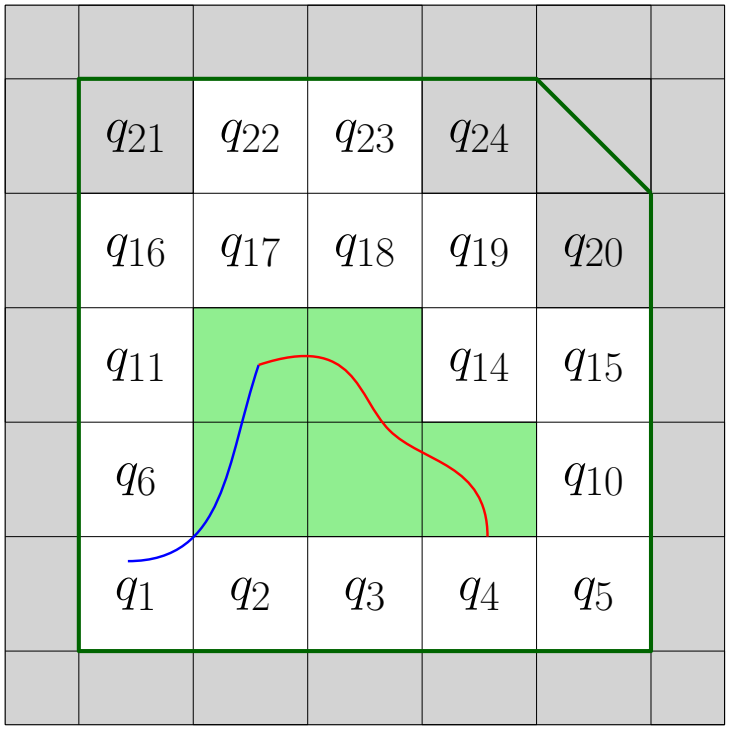
Lazy Exploration Restricted to Boundary States. Control Refinement

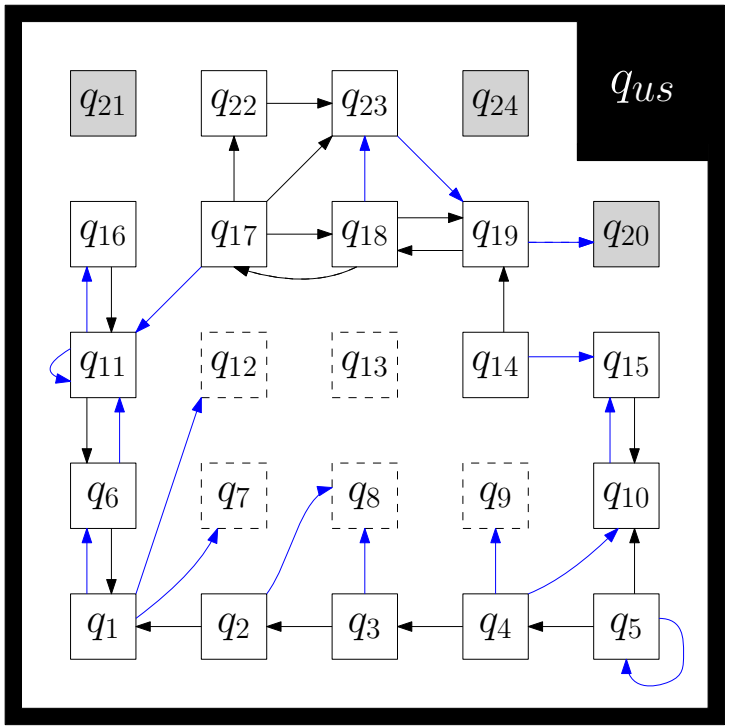




With a novel refinement scheme the abstract controller pushes the system from a boundary state back to the interior, while for inner states any input from U is valid.

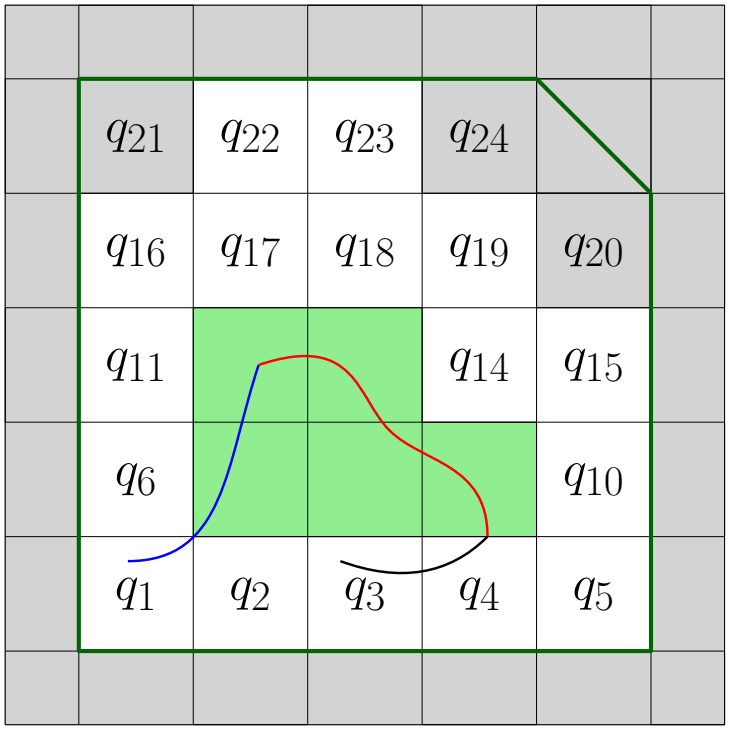
Lazy Exploration Restricted to Boundary States. Control Refinement

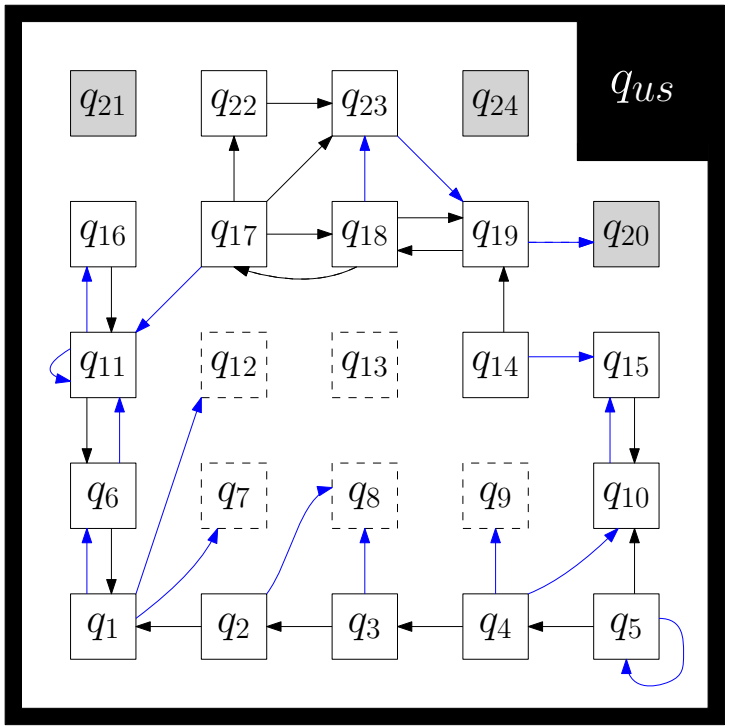




With a novel refinement scheme the abstract controller pushes the system from a boundary state back to the interior, while for inner states any input from U is valid.

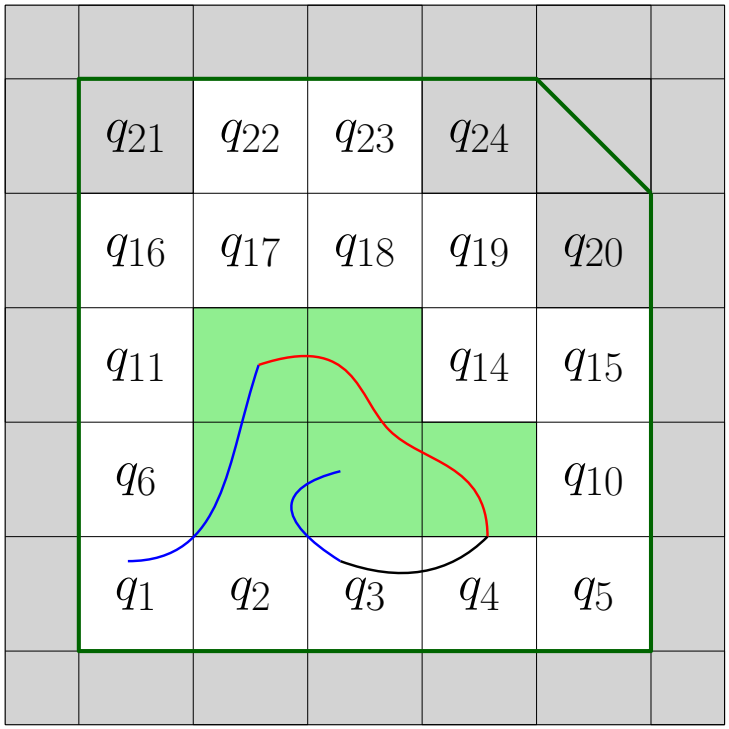
Lazy Exploration Restricted to Boundary States. Control Refinement

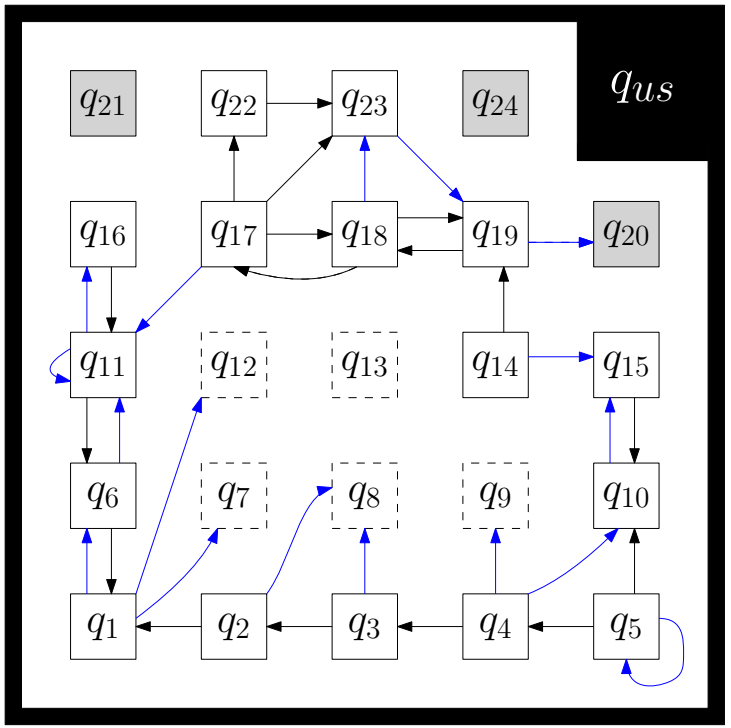




With a novel refinement scheme the abstract controller pushes the system from a boundary state back to the interior, while for inner states any input from U is valid.

Lazy Exploration Restricted to Boundary States. Control Refinement





With a novel refinement scheme the abstract controller pushes the system from a boundary state back to the interior, while for inner states any input from U is valid.

Adaptive Cruise Control

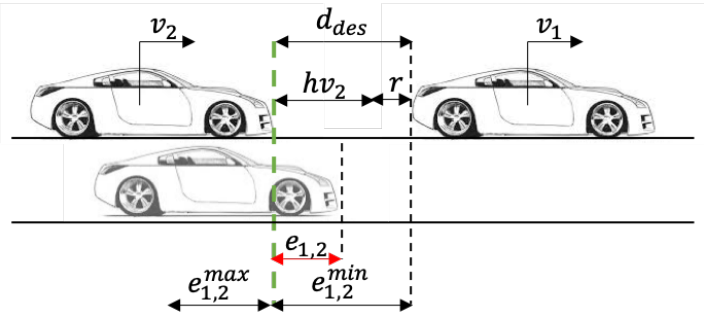
Two cars move along a straight road. The lead car acts as a disturbance, the following car is under our control.

Each car is modeled as a point mass

$$m_i \dot{v}_i = F_i - (a + bv_i + cv_i^2),\ i = 1, 2.$$

Both cars respect the speed limitations:

$$v_i \in [0, v_{max}],\ i = 1, 2.$$



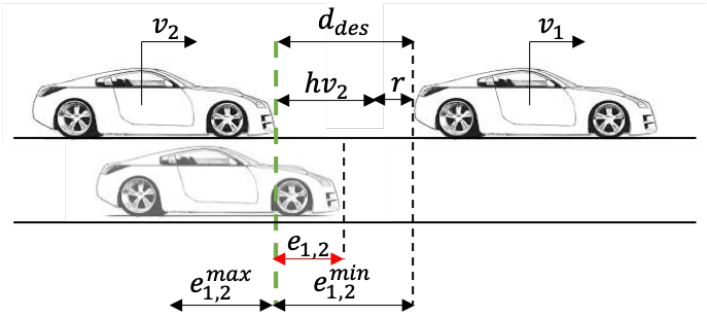
- Disturbance: $F_1 \in [-0.3m_1 g, 0.2m_1 g]$.
- Control: $F_2 \in [-0.3m_2 g, 0.2m_2 g]$.

Adaptive Cruise Control

Two cars move along a straight road. The lead car acts as a disturbance, the following car is under our control.

Each car is modeled as a point mass

$$m_i \dot{v}_i = F_i - (a + bv_i + cv_i^2),\ i = 1, 2.$$



Both cars respect the speed limitations:

$$v_i \in [0, v_{max}],\ i = 1, 2.$$

The deviation from the desired distance d_{des}:

$$\dot{e}_{12} = v_1 - v_2 + h\dot{v}_2$$

- Disturbance: $F_1 \in [-0.3m_1 g, 0.2m_1 g]$.
- Control: $F_2 \in [-0.3m_2 g, 0.2m_2 g]$.
- Specification: $d_{des} = hv_2 + r$, $h > 0$, $e_{1,2} \in [-hv_2 - r, e_{1,2}^{max}]$

Adaptive Cruise Control

Let us use as abstract inputs feedback stabilizing control laws, instead of constant inputs.

Each car is modeled as a point mass

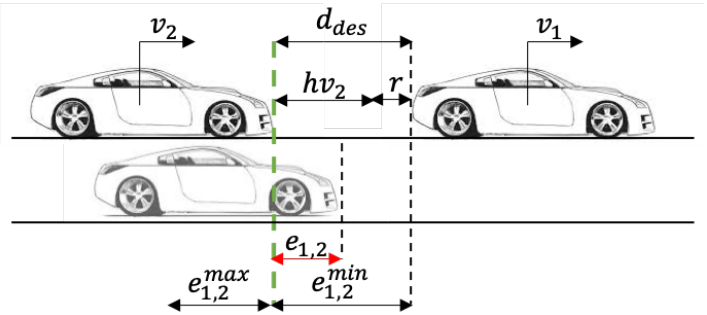
$$m_i \dot{v}_i = F_i - (a + bv_i + cv_i^2),\ i = 1, 2.$$

Both cars respect the speed limitations:

$$v_i \in [0, v_{max}],\ i = 1, 2.$$

The deviation from the desired distance d_{des}:

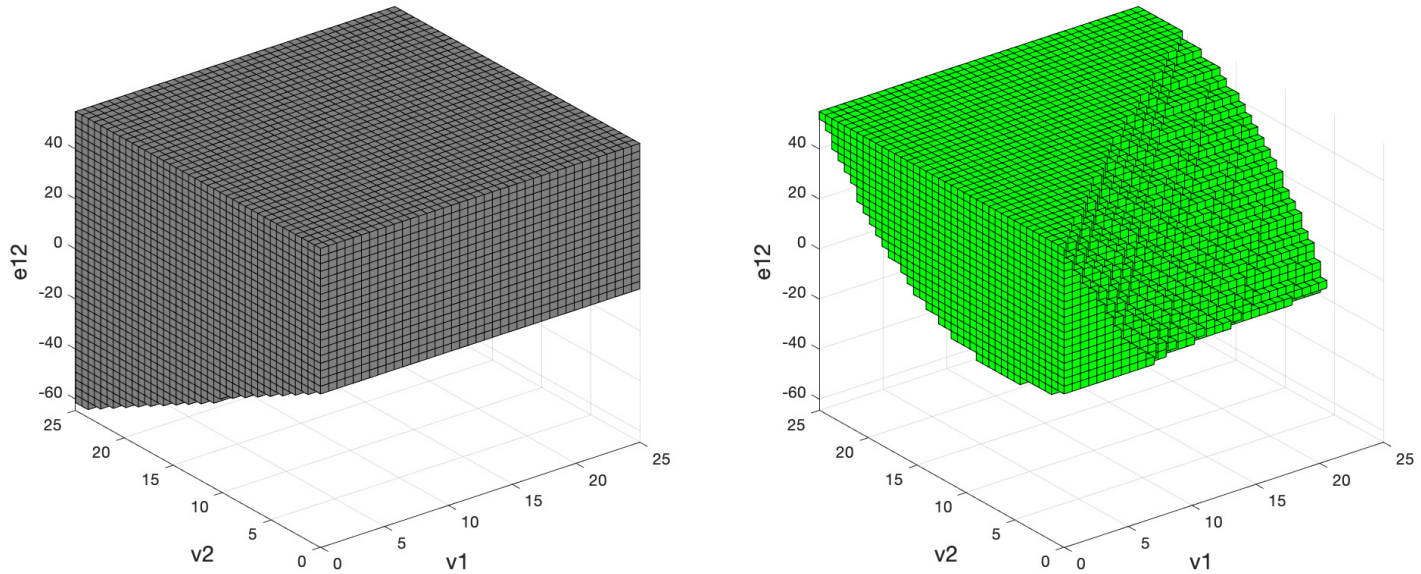
$$\dot{e}_{12} = v_1 - v_2 + h\dot{v}_2$$



- Disturbance: $F_1 \in [-0.3m_1 g, 0.2m_1 g]$.
- Control: $F_2 \in [-0.3m_2 g, 0.2m_2 g]$.
- Specification: $d_{des} = hv_2 + r$, $h > 0$, $e_{1,2} \in [-hv_2 - r, e_{1,2}^{max}]$

Adaptive Cruise Control. Simulation Results

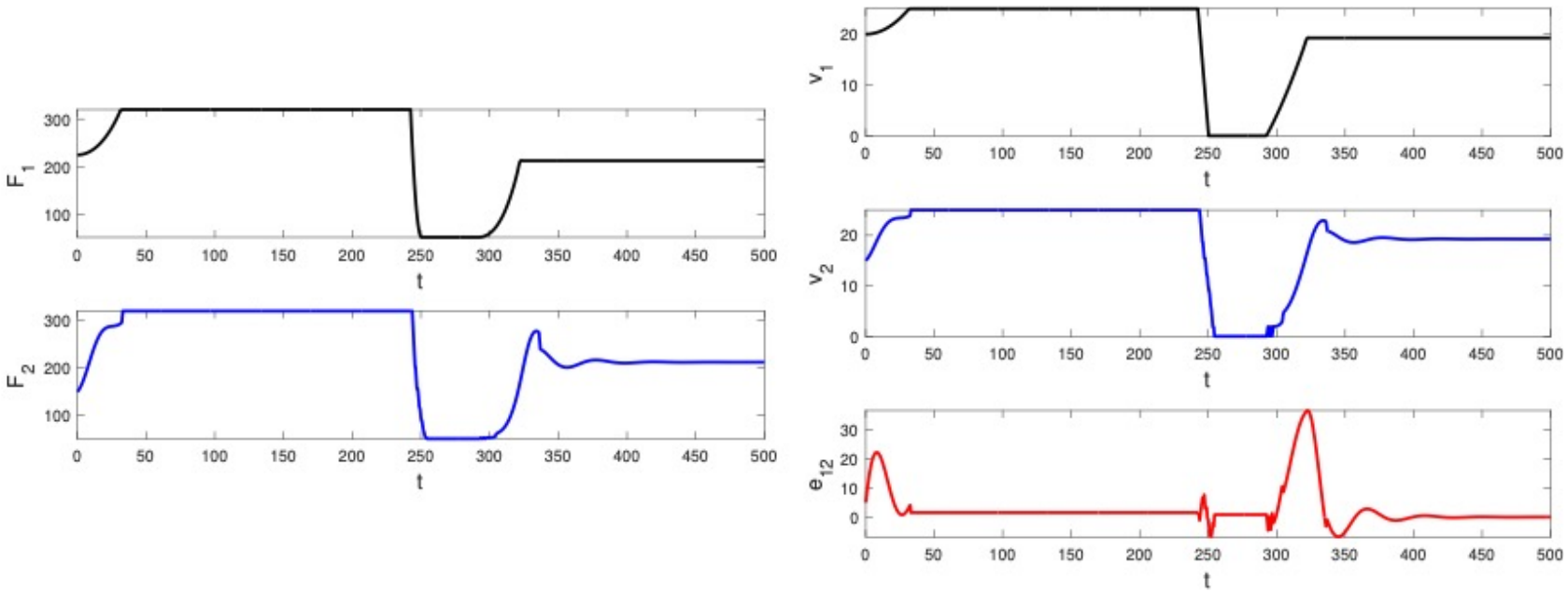
We have implemented classical synthesis approach and lazy synthesis approach for a multi-scale time-sampling abstraction.



In both cases, the controllable domains (right figure) for a given safe set (left figure) coincide. However, lazy approach is 2.58 times faster than the classical one since it explores 19574 less states.

Adaptive Cruise Control. Simulation results

We also simulated a closed-loop trajectory for a given disturbance realisation F_1.

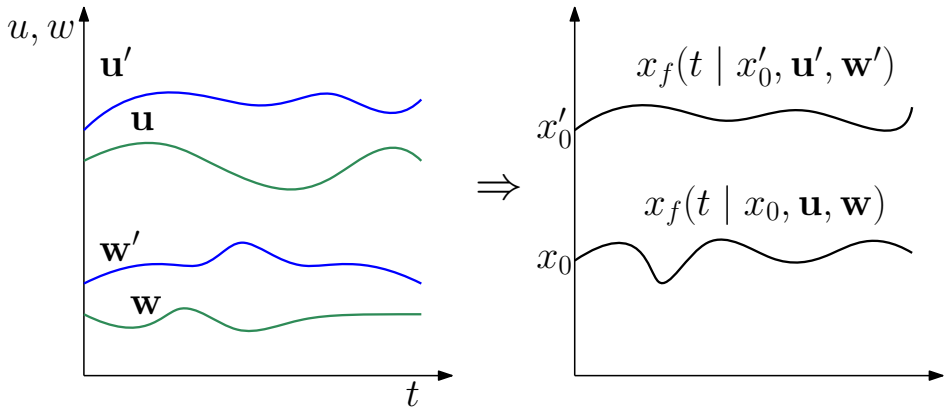


The closed-loop trajectory satisfies the safety restriction and shows a nice behavior in terms of stability.

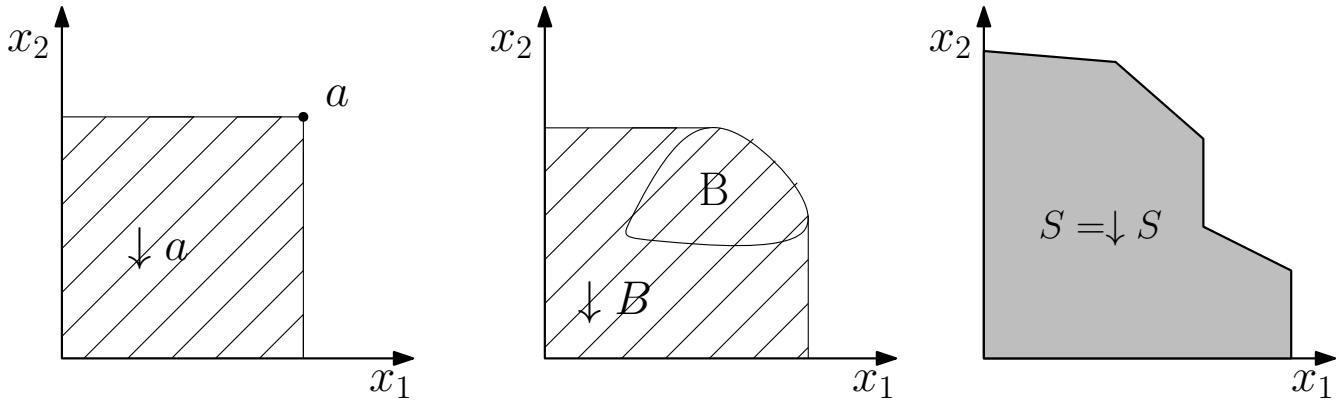
Lazy synthesis approach for monotone transition systems.

Monotone Systems and Lower-closed Safety Specification

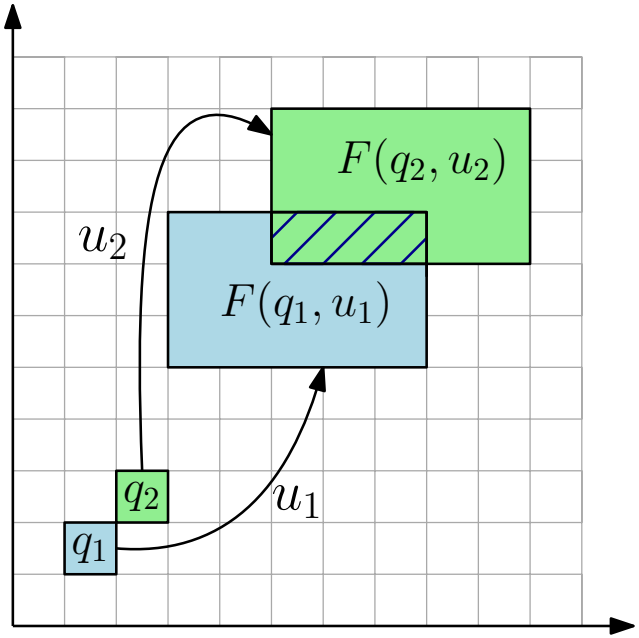
Monotone dynamical system $\dot{x} = f(x, u, w)$

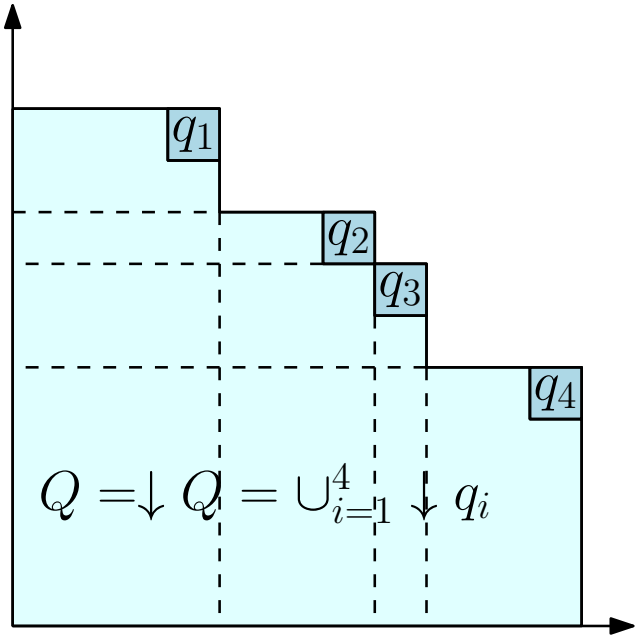


Lower-closed safety specification



Monotone Systems and Lower-closed Safety Specification





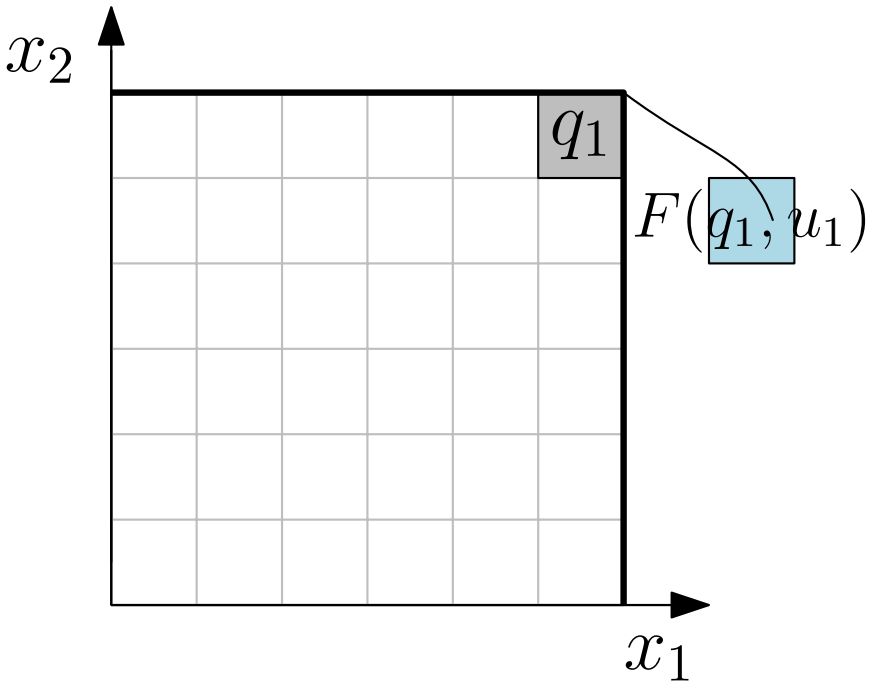
If a transition system is monotone then for all states $q_1 \preceq_Q q_2$, inputs $u_1 \preceq_U u_2$ we have $F(q_1, u_1) \subseteq \downarrow F(q_2, u_2)$.

A finite lower closed set can be represented by its basis[1].

[1] A. Finkel and P. Schnoebelen (2001). Well-structured transition systems every-where! Theoretical Computer Science.

Lazy Computation of Maximal Safety Controller

Let a set of inputs $U = [u_{min}, u_{max}]$, a disturbance set $W = [w_{min}, w_{max}]$, and there is a total order on input space $u_{min} = u_1 < u_2 < \ldots < u_N = u_{max}$.

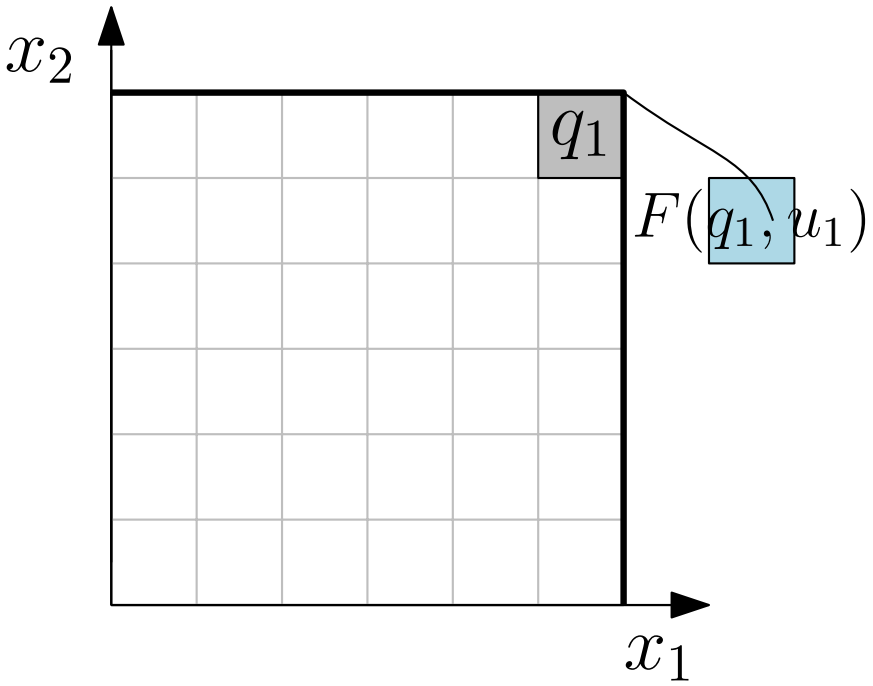


- Non-deterministic transition system is replaceable by deterministic one[1].

[1] E. S. Kim, M. Arcak, and S. A. Seshia (2016). Directed Specifications and Assumption Mining for Monotone Dynamical Systems. HSCC.

Lazy Computation of Maximal Safety Controller

Let a set of inputs $U = [u_{min}, u_{max}]$, a disturbance set $W = [w_{min}, w_{max}]$, and there is a total order on input space $u_{min} = u_1 < u_2 < \ldots < u_N = u_{max}$.

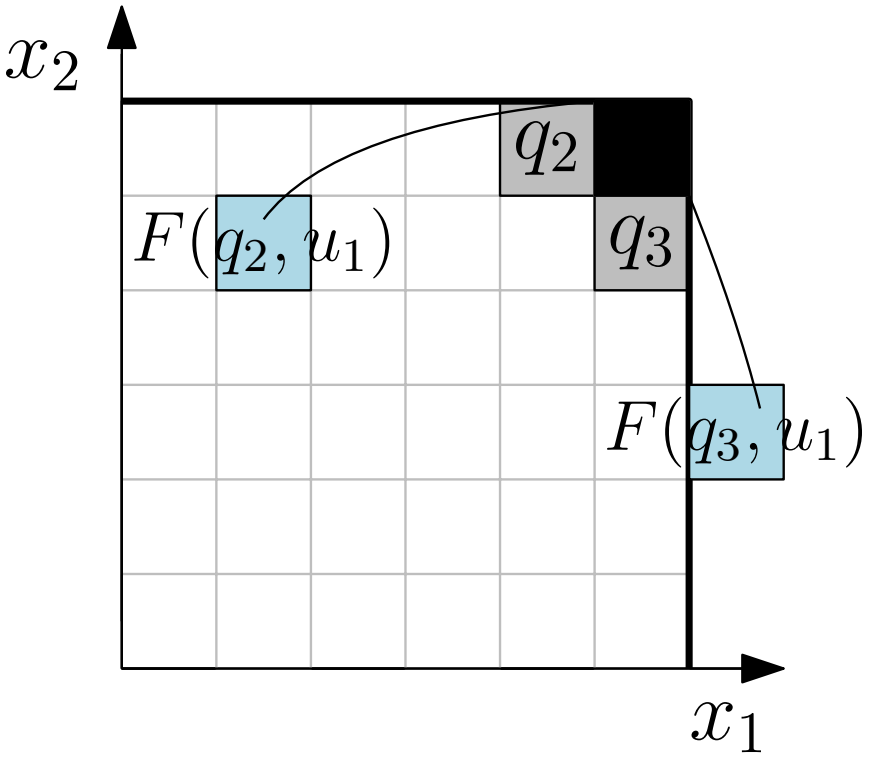


- Non-deterministic transition system is replaceable by deterministic one[1].
- Control domain can be computed by the iterative exploration of the basis, while using only the lower priority inputs.

[1] E. S. Kim, M. Arcak, and S. A. Seshia (2016). Directed Specifications and Assumption Mining for Monotone Dynamical Systems. HSCC.

Lazy Computation of Maximal Safety Controller

Let a set of inputs $U = [u_{min}, u_{max}]$, a disturbance set $W = [w_{min}, w_{max}]$, and there is a total order on input space $u_{min} = u_1 < u_2 < \ldots < u_N = u_{max}$.

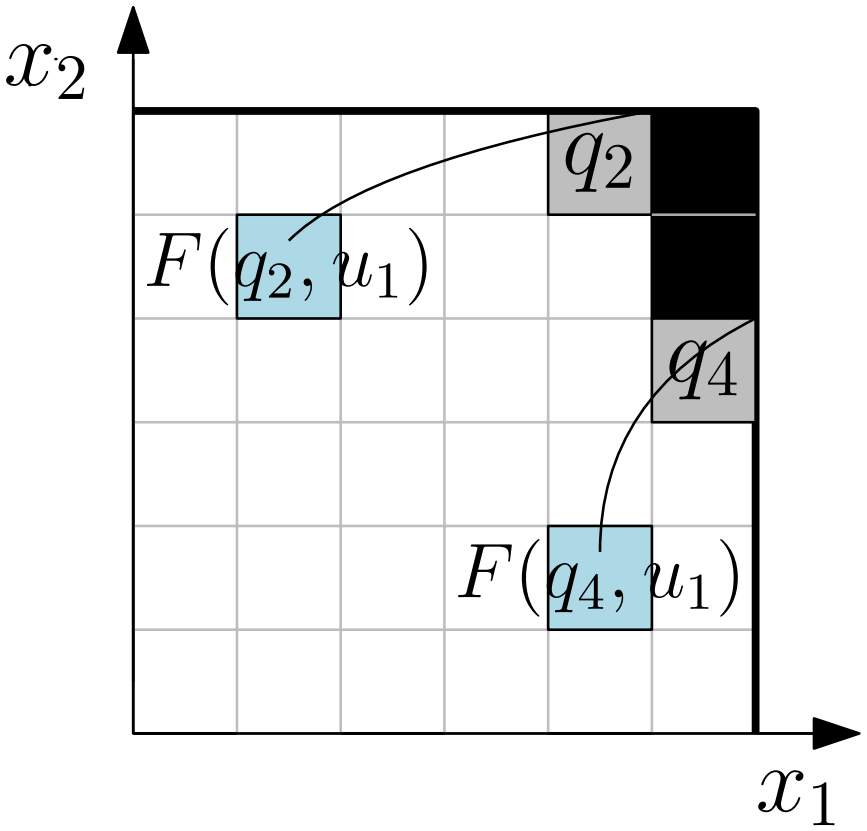


- Non-deterministic transition system is replaceable by deterministic one[1].
- Control domain can be computed by the iterative exploration of the basis, while using only the lower priority inputs.

[1] E. S. Kim, M. Arcak, and S. A. Seshia (2016). Directed Specifications and Assumption Mining for Monotone Dynamical Systems. HSCC.

Lazy Computation of Maximal Safety Controller

Let a set of inputs $U = [u_{min}, u_{max}]$, a disturbance set $W = [w_{min}, w_{max}]$, and there is a total order on input space $u_{min} = u_1 < u_2 < \ldots < u_N = u_{max}$.

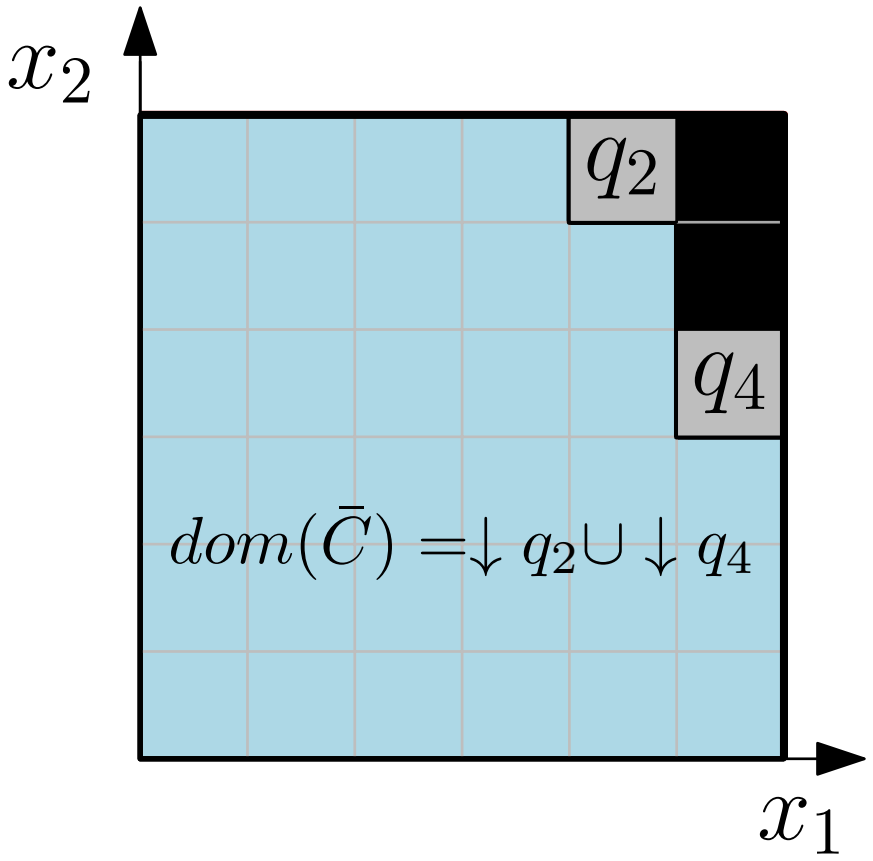


- Non-deterministic transition system is replaceable by deterministic one[1].
- Control domain can be computed by the iterative exploration of the basis, while using only the lower priority inputs.

[1] E. S. Kim, M. Arcak, and S. A. Seshia (2016). Directed Specifications and Assumption Mining for Monotone Dynamical Systems. HSCC.

Lazy Computation of Maximal Safety Controller

Let a set of inputs $U = [u_{min}, u_{max}]$, a disturbance set $W = [w_{min}, w_{max}]$, and there is a total order on input space $u_{min} = u_1 < u_2 < \ldots < u_N = u_{max}$.

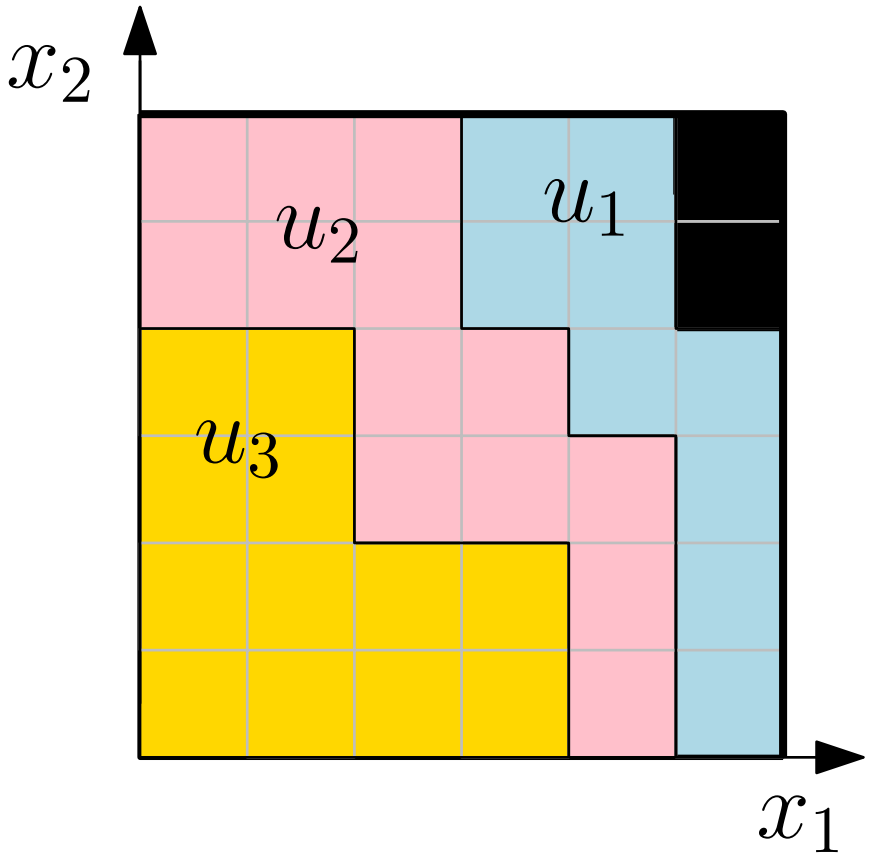


- Non-deterministic transition system is replaceable by deterministic one[1].
- Control domain can be computed by the iterative exploration of the basis, while using only the lower priority inputs.
- We can repeat the similar procedure to find a maximal safety controller.

[1] E. S. Kim, M. Arcak, and S. A. Seshia (2016). Directed Specifications and Assumption Mining for Monotone Dynamical Systems. HSCC.

Lazy Computation of Maximal Safety Controller

Let a set of inputs $U = [u_{min}, u_{max}]$, a disturbance set $W = [w_{min}, w_{max}]$, and there is a total order on input space $u_{min} = u_1 < u_2 < \ldots < u_N = u_{max}$.



- Non-deterministic transition system is replaceable by deterministic one[1].
- Control domain can be computed by the iterative exploration of the basis, while using only the lower priority inputs.
- We can repeat the similar procedure to find a maximal safety controller.

[1] E. S. Kim, M. Arcak, and S. A. Seshia (2016). Directed Specifications and Assumption Mining for Monotone Dynamical Systems. HSCC.

Illustrative example. Adaptive cruise control

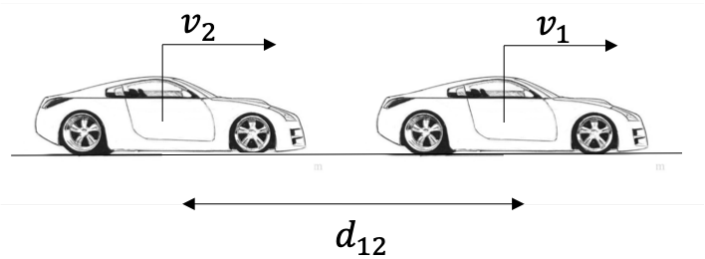
Two cars move along a straight road. The acceleration F_1 of the fist car is a disturbance, the acceleration F_2 of the second is a controlled parameter.

Each car is modeled as a point mass

$$m_i \dot{v}_i = F_i - (a + bv_i + cv_i^2),\ i = 1, 2.$$

Both cars respect the speed limitations:

$$v_i \in [0, v_{max}],\ i = 1, 2.$$



- Disturbance: $F_1 \in [-0.3m_1 g, 0.2m_1 g]$.
- Control: $F_2 \in [-0.3m_2 g, 0.2m_2 g]$.

Illustrative example. Adaptive cruise control

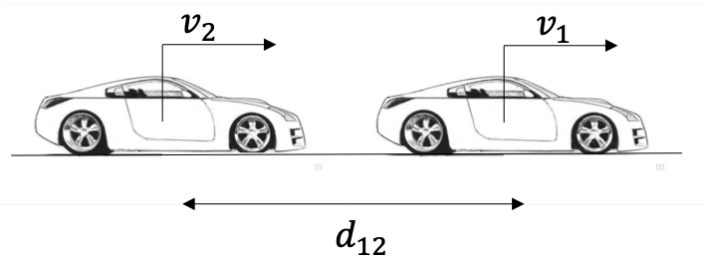
Two cars move along a straight road. The acceleration F_1 of the fist car is a disturbance, the acceleration F_2 of the second is a controlled parameter.

Each car is modeled as a point mass

$$m_i \dot{v}_i = F_i - (a + bv_i + cv_i^2),\ i = 1, 2.$$

Both cars respect the speed limitations:

$$v_i \in [0, v_{max}],\ i = 1, 2.$$



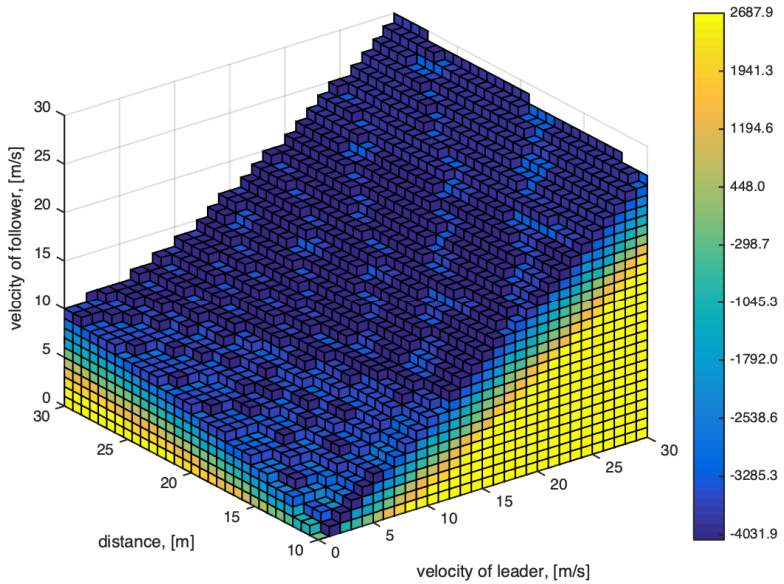
The distance between the cars:

$$\dot{d}_{12} = v_1 - v_2$$

- Disturbance: $F_1 \in [-0.3m_1 g, 0.2m_1 g]$.
- Control: $F_2 \in [-0.3m_2 g, 0.2m_2 g]$.
- Specification: $d_{12} >= 10m$.

Adaptive Cruise Control. Simulation Results

Maximal safety controller for a lower-closed safety specification.



n_x	T^s_{lm}	T^s_{cl} / T^s_{lm}	T^s_{3v} / T^s_{lm}
(31,31,31)	18.16 s	23.19	14.92
(63,63,63)	118.67 s	30.85	16.25
n_u	T^s_{lm}	T^s_{cl} / T^s_{lm}	T^s_{3v} / T^s_{lm}
20	20.56 s	41.08	25.96
40	25.26 s	66.67	43.25

T^s_{cl} : E. S. Kim, M. Arcak, and S. A. Seshia (2016). Directed Specifications and Assumption Mining for Monotone Dynamical Systems. HSCC.

T^s_{3v} : O. Hussien, and P. Tabuada (2018). Lazy controller synthesis using three-valued abstractions for safety and reachability specifications. CDC.

Lazy Synthesis Approaches

Lazy synthesis approach for multi-scale symbolic models.

- E. Ivanova and A. Girard (2020). Lazy safety controller synthesis with multi-scale adaptive-sampling abstractions of nonlinear systems. IFAC WC.

Lazy exploration restricted to boundary states.

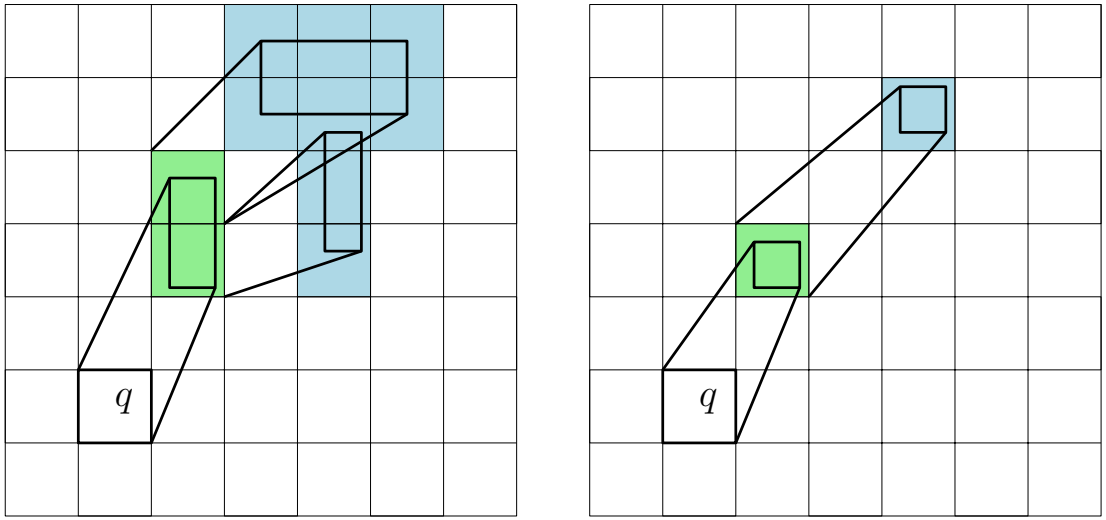
- E. Ivanova and A. Girard (2021). Lazy Symbolic Controller for Continuous-Time Systems Based on Safe Set Boundary Exploration, IFAC ADHS.

Lazy synthesis approach for monotone transition systems.

- E. Ivanova, A. Saoud, and A. Girard (2021). Lazy Controller Synthesis for Monotone Transition Systems and Directed Safety Specifications, Automatica.
- A. Saoud , E. Ivanova and A. Girard (2019). Efficient Synthesis for Monotone Transition Systems and Directed Safety Specifications. IEEE CDC.

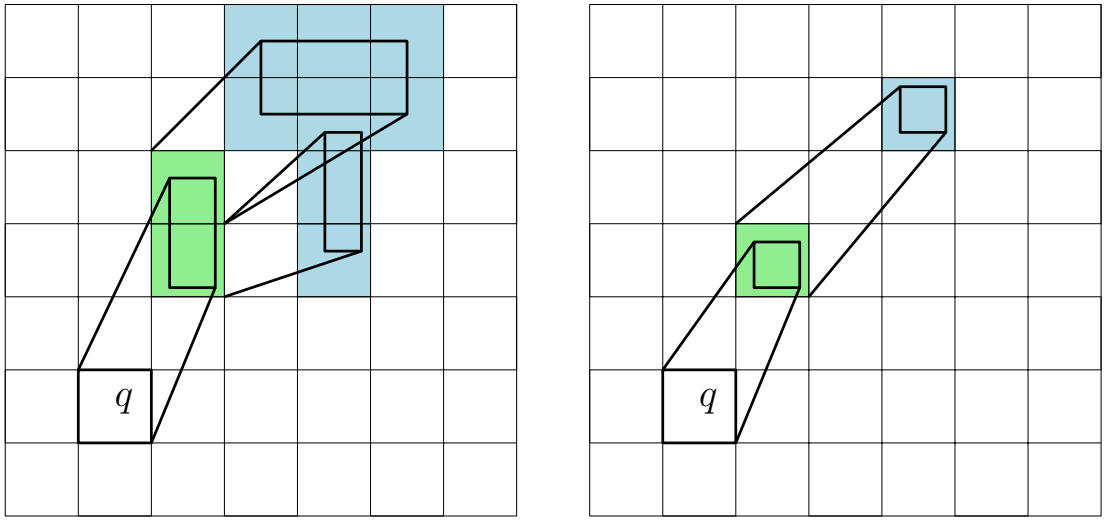
What next?

An important remark...



- Synthesis with abstraction based approaches requires efficient over-approximations of reachable sets
  - L. Jaulin et al. (2001). Applied Interval Analysis. Springer-Verlag London.
  - P.-J. Meyer et al. (2021). Interval Reachability Analysis. Springer Briefs in Control, Automation and Robotics.

An important remark...



- Synthesis with abstraction based approaches requires efficient over-approximations of reachable sets
  - L. Jaulin et al. (2001). Applied Interval Analysis. Springer-Verlag London.
  - P.-J. Meyer et al. (2021). Interval Reachability Analysis. Springer Briefs in Control, Automation and Robotics.
- To address more complex specifications there is a demand to construct deterministic abstractions.
  - P. Tabuada (2008). An approximate simulation approach to symbolic control. IEEE Transactionson Automatic Control.
  - V. Sinyakov and A. Girard (2021). Abstraction of Continuous-time Systems Based on Feedback Controllers and Mixed Monotonicity.

Mixed Monotone Functions

A function $f\colon \mathbb{R}^n \to \mathbb{R}^m$ is said to be mixed-monotone, if there exists $g\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^m$ satisfying the following:

1. $g(x,x) = f(x)$;
2. for all $x, \hat{x} \in \mathbb{R}^n$ such that $x \preceq \hat{x}$ the following holds $g(x,y) \preceq g(\hat{x},y)$ for all $y \in \mathbb{R}^n$;
3. for all $y, \hat{y} \in \mathbb{R}^n$ such that $y \preceq \hat{y}$ the following holds $g(x,y) \succeq g(x,\hat{y})$ for all $x \in \mathbb{R}^n$.

A function g satisfying the above conditions is called a decomposition function of f.

Decomposition function g is called tight if for all $\underline{x}, \overline{x} \in \mathbb{R}^n$ s.t. $\underline{x} \preceq \overline{x}$, $[g(\underline{x},\overline{x}), g(\underline{x},\overline{x})]$ is the smallest interval that contains $\{f(x) | x \in [\underline{x},\overline{x}]\}$. That is

$$[g(\underline{x},\overline{x}), g(\underline{x},\overline{x})] = [\inf_{\xi \in [\underline{x},\overline{x}]} f(\xi), \sup_{\xi \in [\underline{x},\overline{x}]} f(\xi)].$$

Tight Decomposition Functions

Let for all $x, y \in \mathbb{R}^n$ and $h\colon \mathbb{R}^n \to \mathbb{R}^n$, define

$$opt_{\xi}^{(x,y)} h(\xi) = \begin{cases} \inf\limits_{\xi \in [x,y]} h(\xi), & \text{if } x \leq y \\ \sup\limits_{\xi \in [y,x]} h(\xi), & \text{if } x > y \end{cases}$$

Theorem[1]. Let $f\colon \mathbb{R}^n \to \mathbb{R}^m$ be such that $opt_{\xi_i}^{(x_i,y_i)} f(\xi_i)$ is well defined, then the following $g\colon \mathbb{R}^{2n} \to \mathbb{R}^m$ defined element-wise by

$$g_j(x,y) = opt_{\xi_1}^{(x_1,y_1)} opt_{\xi_2}^{(x_2,y_2)} \dots opt_{\xi_n}^{(x_n,y_n)} f_j(\xi), \quad j = 1, 2, \dots, m$$

is a tight decomposition function of f.

[1] L. Yang and N. Ozay. (2019). Tight decomposition functions for mixed monotonicity. 58th Conference on Decision and Control (CDC).

Decomposition functions. Examples.

For $f(x_1, x_2) = x_1 + x_2$ the tight decomposition function $g(x, \hat{x}) = x_1 + x_2$.

For $f(x_1, x_2) = x_1 - x_2$ the tight decomposition function $g(x, \hat{x}) = x_1 - \hat{x}_2$.

For $f(x_1, x_2) = x_1 x_2$ the tight decomposition function

$$g(x, \hat{x}) = \begin{cases} x_1 x_2, \text{ if } x_2 \geq 0 \\ \hat{x}_1 x_2, \text{ if } x_2 < 0 \end{cases} \quad \text{or} \quad g(x, \hat{x}) = \begin{cases} x_1 x_2, \text{ if } x_1 \geq 0 \\ x_1 \hat{x}_2, \text{ if } x_1 < 0 \end{cases}$$

.

We also can derive a decomposition function for f from taylor's approximations

$$g_f(x, \hat{x}) = \begin{cases} f(\hat{x}) + \sup\limits_{z \in [\hat{x}, x]} \left(\sum\limits_{i=1}^{n} \frac{(z-\hat{x})^i}{i!} f^{(i)}(\hat{x}) \right) + \frac{(x-\hat{x})^{n+1}}{(n+1)!} \sup\limits_{z \in [\hat{x}, x]} \max(0, f^{(n+1)}(z)) & \text{if } \hat{x} \leq x \\ f(\hat{x}) + \inf\limits_{z \in [x, \hat{x}]} \left(\sum\limits_{i=1}^{n} \frac{(z-\hat{x})^i}{i!} f^{(i)}(\hat{x}) \right) + \frac{(x-\hat{x})^{n+1}}{(n+1)!} \sup\limits_{z \in [\hat{x}, x]} \max(0, f^{(n+1)}(z)) & \text{if } \hat{x} \geq x, \, n \text{ even} \\ f(\hat{x}) + \inf\limits_{z \in [x, \hat{x}]} \left(\sum\limits_{i=1}^{n} \frac{(z-\hat{x})^i}{i!} f^{(i)}(\hat{x}) \right) + \frac{(x-\hat{x})^{n+1}}{(n+1)!} \inf\limits_{z \in [x, \hat{x}]} \min(0, f^{(n+1)}(z)) & \text{if } \hat{x} \geq x, \, n \text{ odd} \end{cases}$$

Reachable Sets Over-approximations based on Mixed Monotonicity.

Theorem [1]. Given system $\dot{x} = f(x)$, where state $x \in X \subset \mathbb{R}^n$ and vector field f is defined on some open set X_e containing set X, assume that f is mixed monotone and is locally Lipschitz on X_e, the system is forward complete, and the domain X is positively invariant under the considered dynamics. Then, the flow map Φ_t is mixed monotone.

Let the initial set $X^0 = [\underline{x}^0, \overline{x}^0]$ is an interval, consider an embedding system system

$$\begin{aligned} \dot{\overline{x}}_i &= g_i(\overline{x}, \underline{x}), \quad \overline{x}_i(0) = \overline{x}_i^0 \\ \dot{\underline{x}}_i &= g_i(\underline{x}, \overline{x}), \quad \underline{x}_i(0) = \underline{x}_i^0 \end{aligned} \tag{1}$$

Then

$$Reach(t, X^0) \subseteq [\underline{\mathbf{x}}(t; [\underline{x}^0, \overline{x}^0]), \overline{\mathbf{x}}(t; [\overline{x}^0, \underline{x}^0]].$$

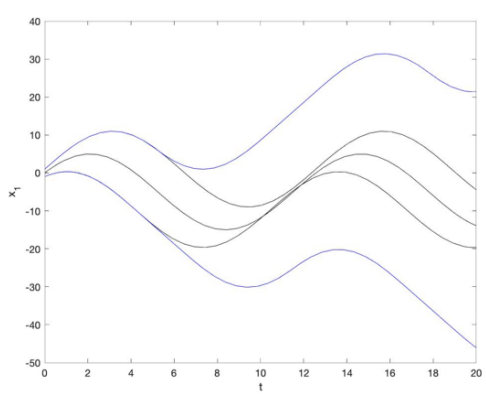
[1] L. Yang, O. Mickelin and N. Ozay. (2019) On Sufficient Conditions for Mixed Monotonicity. IEEE Transactions on Automatic Control.

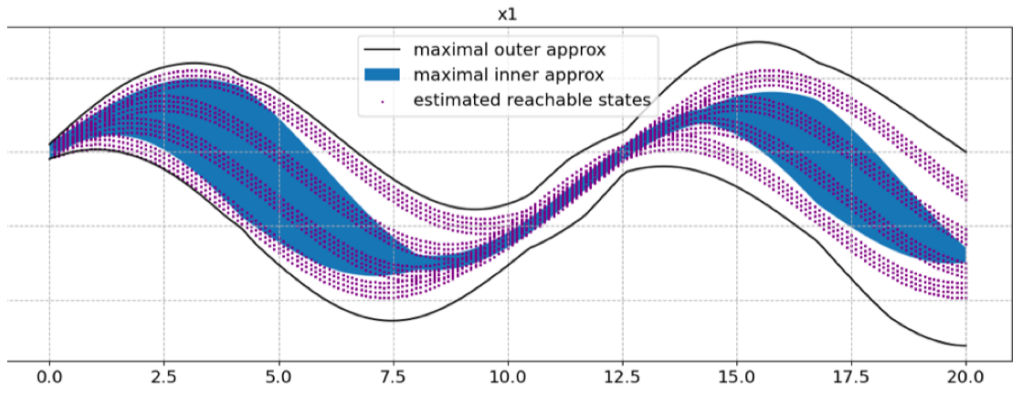
Are they not too much conservative?

Unicycle example

$$\dot{x}_1 = v\cos(x_3)$$
$$\dot{x}_2 = v\sin(x_3)$$
$$\dot{x}_3 = w$$

Here v, w are given functions.

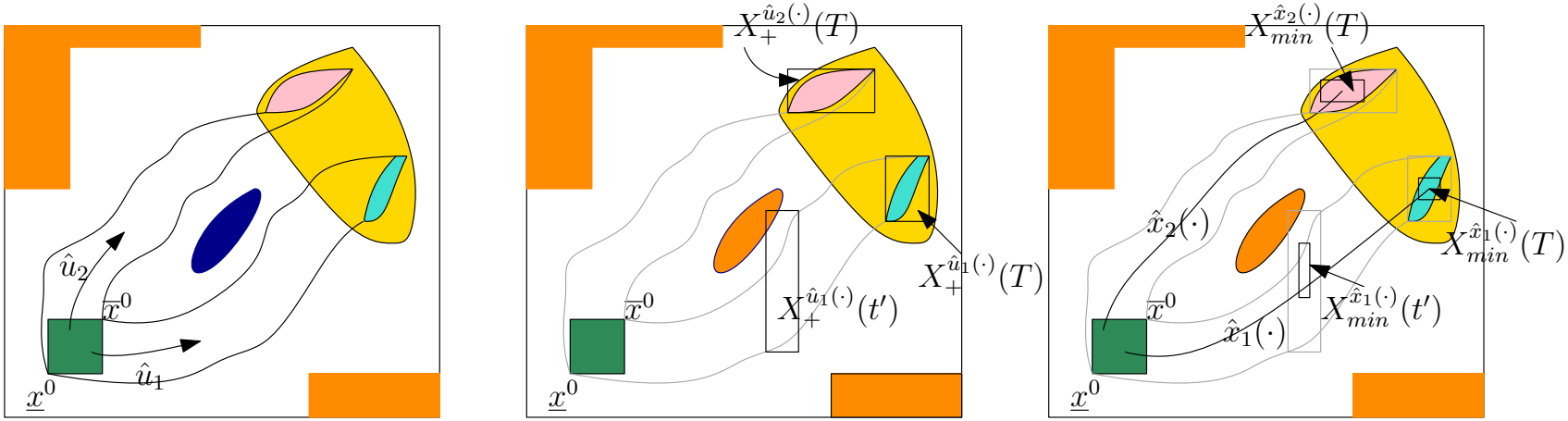




E. Goubault and S. Putot. (2017). Forward Inner-Approximated Reachability of Non-Linear Continuous Systems. In Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control (HSCC '17)

Mixed Monotone Systems: Use Control to Compress the Reachable Set

Under some assumptions, one can use controllable inputs to compress the reachable set including the given trajectory.



[1] V. Sinyakov and A. Girard (2021). Abstraction of Continuous-time Systems Based on Feedback Controllers and Mixed Monotonicity.

Conclusion. Questions?

Abstraction-Based Synthesis Techniques

- abstraction-based methods allow to deal with hybrid dynamic of the systems and complex control objectives, but from a poor scalability.
- lazy synthesis approaches computes the symbolic model on-the-fly, and avoid non-essential for synthesis purpose computations.
- the worst-case complexity of lazy approaches coincide with computational complexity of classical synthesis algorithm, but they are more efficient in practice.

Reachability Analysis

- is a core of abstraction-based synthesis techniques, formal verification, robust control model predictive control and other approaches with safety guarantees.
- we have to develop efficient approaches to compute accurate approximations.

Thank you for your attention!