Formal Methods for the Control of Large-scale Networked Nonlinear Systems with Logic Specifications

Basilica di Santa Maria di Collemaggio, 1287, L'Aquila

Lecture L7b: Efficient algorithms for controller synthesis

Speaker: Alessandro Borri

What's new?

- Computation of controllers presented in lecture L7a may require high computational effort

- Here: efficient algorithms for computational complexity reduction in designing controllers

Tools:

- On-the-fly algorithms studied in computer science

Lecture based on:

[Pola et al., TAC12] Pola, G., Borri, A., Di Benedetto, M.D., Integrated design of symbolic controllers for nonlinear systems, IEEE Transactions on Automatic Control, 57(2):534-539, February 2012

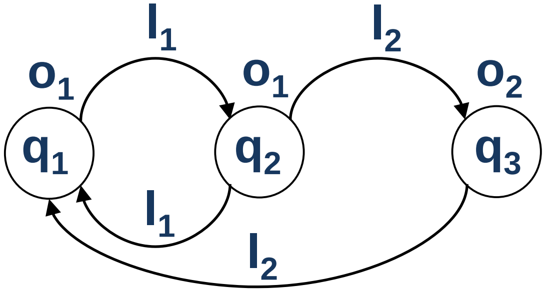
Preliminary definitions

Definition A transition system is a tuple:

$$T = (Q, Q_0, L, \longrightarrow, Q_m, O, H),$$

consisting of:

- a set of states Q
- a set of initial states $Q_0 \subseteq Q$
- a set of control labels L
- a transition relation $\longrightarrow \subseteq Q \times L \times Q$
- a set of marked states $Q_m \subseteq Q$
- an output set O
- an output function $H: Q \rightarrow O$



We will follow standard practice and denote $(q, l, q') \in \longrightarrow$ by $q \xrightarrow{l} q'$

Review: Construction of symbolic models

We consider digital control systems, i.e. control systems where input signals are piecewise constant.

Consider a nonlinear digital control system

$$T(\Sigma) = (X, X_0, \mathcal{U}, \longrightarrow, X_m, O, H),$$

and given some $\tau > 0$, define the transition system

$$T_\tau(\Sigma) = (X, X_0, \mathcal{U}_\tau, \longrightarrow_\tau, X_m, O, H),$$

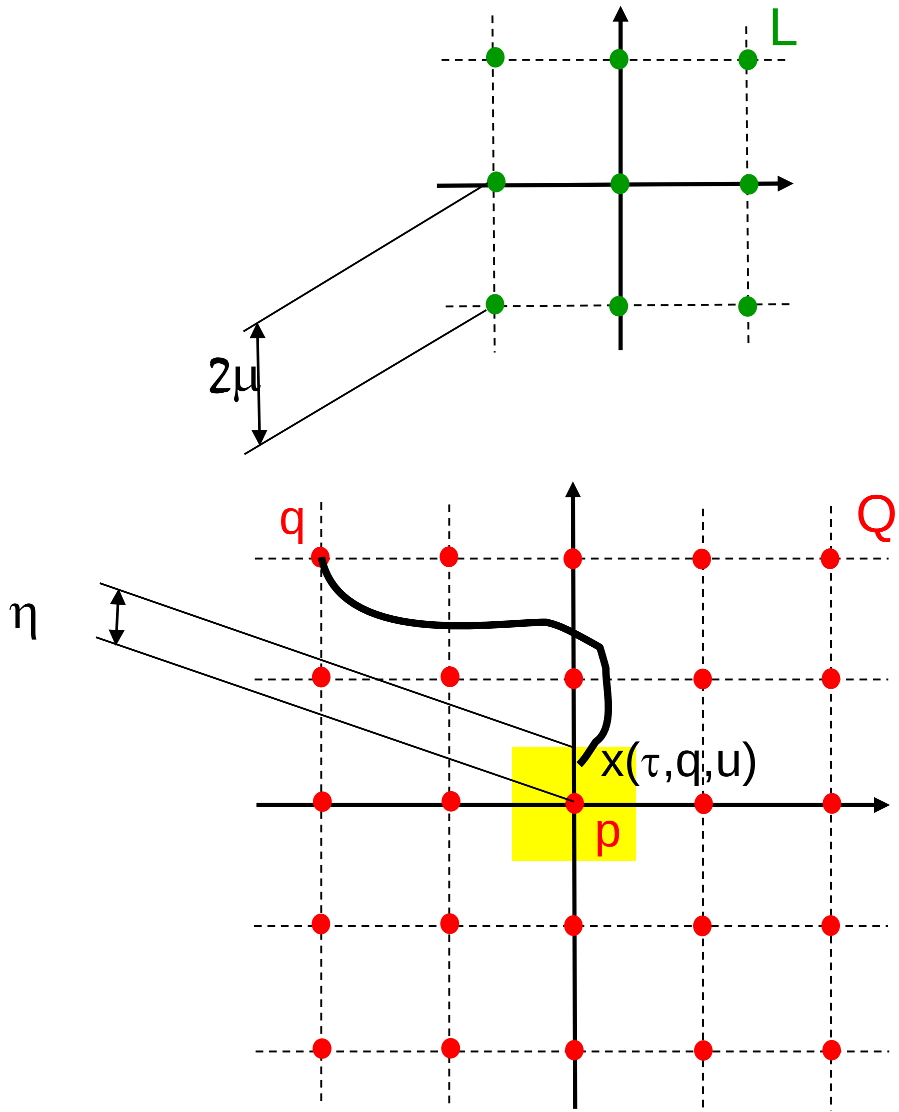
where:

- $\mathcal{U}_\tau$ is the collection of constant input functions $u : [0,\tau] \rightarrow R^m$
- $p \xrightarrow{u}_\tau q$ if $x(\tau,p,u) = q$

Review: Construction of symbolic models

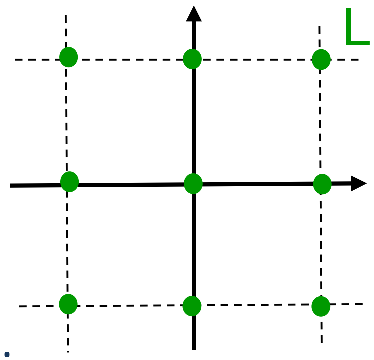
Consider the following parameters:

- $\tau > 0$ sampling time
- $\eta > 0$ state space quantization
- $\mu > 0$ input space quantization



Review: Construction of symbolic models

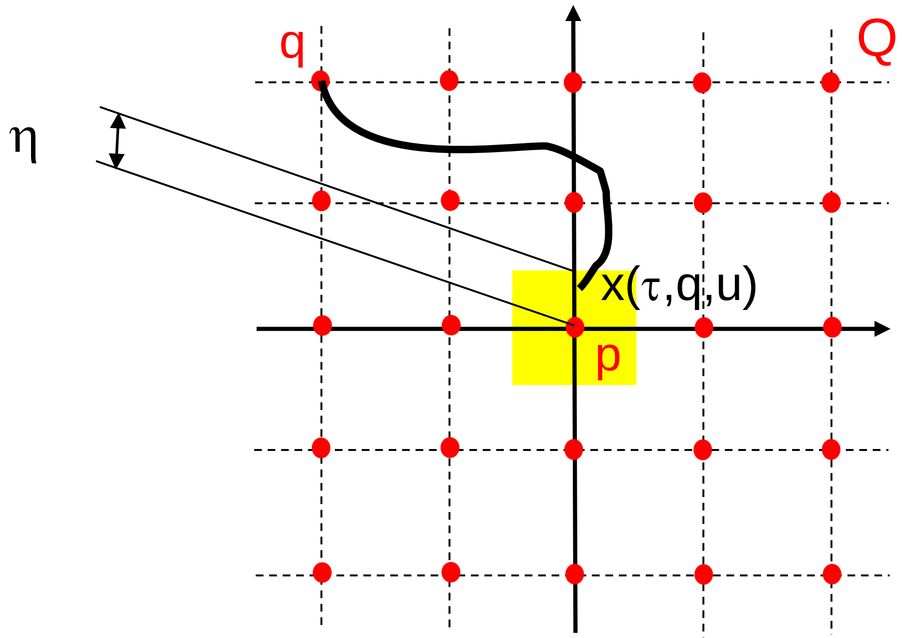
Consider the following parameters:



- $\tau > 0$ sampling time
- $\eta > 0$ state space quantization
- $\mu > 0$ input space quantization

and define $T_{\tau,\eta,\mu}(\Sigma) = (X_{\tau,\eta,\mu}, X_{0,\tau,\eta,\mu}, U_{\tau,\eta,\mu}, \longrightarrow_{\tau,\eta,\mu}, X_{m,\tau,\eta,\mu}, O, H)$, where:

- $X_{\tau,\eta,\mu} = [X]_{2\eta}$
- $X_{0,\tau,\eta,\mu} = X_{\tau,\eta,\mu} \cap X_0$
- $U_{\tau,\eta,\mu} = [U]_{2\mu}$
- $q \xrightarrow{u}_{\tau,\eta,\mu} p$, if $|x(\tau,q,u) - p| \leq \eta$
- $X_{m,\tau,\eta,\mu} = X_{\tau,\eta,\mu} \cap X_m$
- $O = X$
- H is the identity function

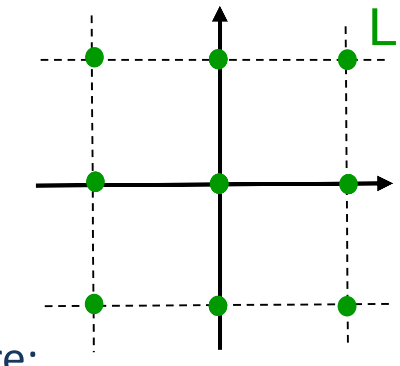


Remark

Transition system $T_{\tau,\eta,\mu}(\Sigma)$ is countable.
If state and input spaces of Σ are bounded then $T_{\tau,\eta,\mu}(\Sigma)$ is symbolic!

Review: Construction of symbolic models

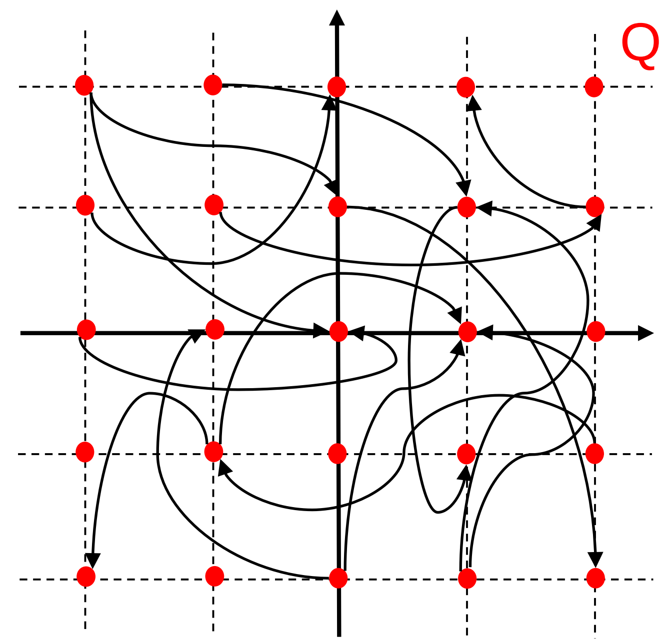
Consider the following parameters:



- $\tau > 0$ sampling time
- $\eta > 0$ state space quantization
- $\mu > 0$ input space quantization

and define $T_{\tau,\eta,\mu}(\Sigma) = (X_{\tau,\eta,\mu}, X_{0,\tau,\eta,\mu}, U_{\tau,\eta,\mu}, \longrightarrow_{\tau,\eta,\mu}, X_{m,\tau,\eta,\mu}, O, H)$, where:

- $X_{\tau,\eta,\mu} = [X]_{2\eta}$
- $X_{0,\tau,\eta,\mu} = X_{\tau,\eta,\mu} \cap X_0$
- $U_{\tau,\eta,\mu} = [U]_{2\mu}$
- $q \xrightarrow{u}_{\tau,\eta,\mu} p$, if $|x(\tau,q,u) - p| \leq \eta$
- $X_{m,\tau,\eta,\mu} = X_{\tau,\eta,\mu} \cap X_m$
- $O = X$
- H is the identity function



Theorem If Σ is δ-ISS, for any desired accuracy $\varepsilon > 0$ and for any $\tau, \eta, \mu > 0$ satisfying

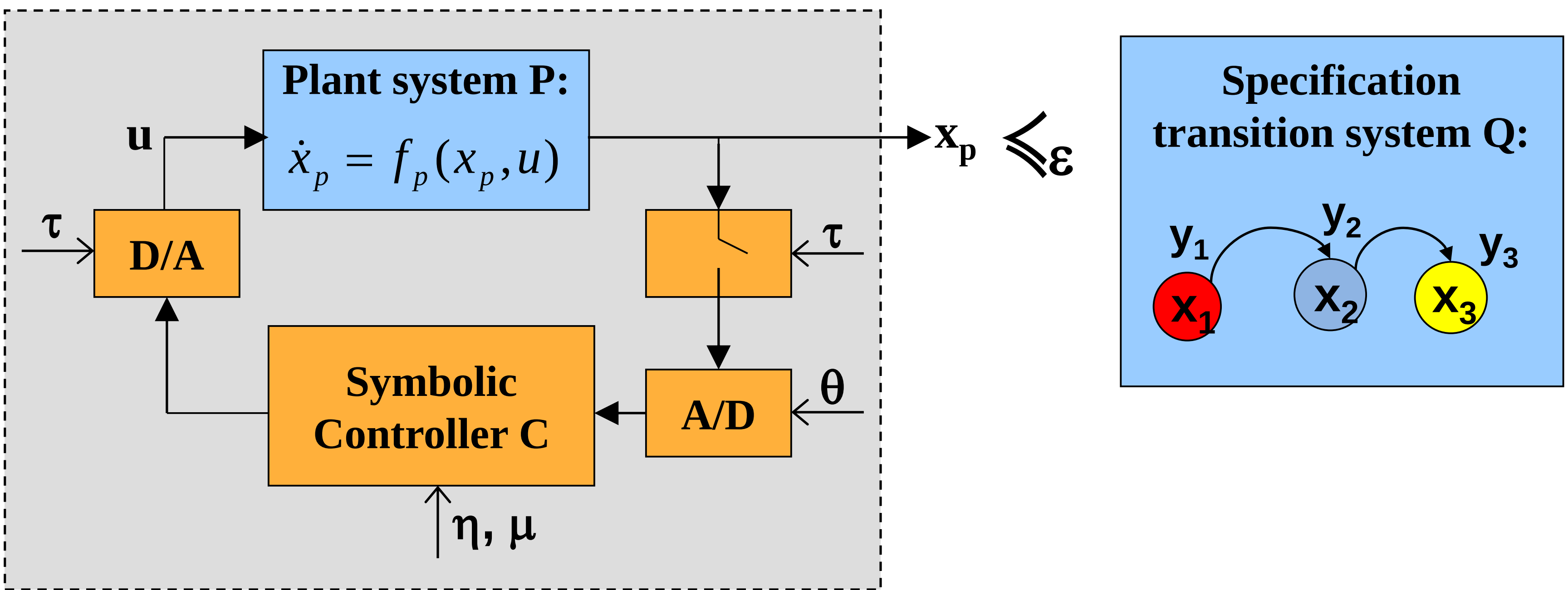
$$\beta(\varepsilon,\tau) + \eta + \gamma(\mu) \leq \varepsilon$$

then $T_\tau(\Sigma)$ and $T_{\tau,\eta,\mu}(\Sigma)$ are ε-bisimilar

Design of symbolic controllers

Problem: Specifications given as deterministic transition systems

Given a plant P, a deterministic specification Q and a desired accuracy $\varepsilon > 0$, find a symbolic controller that implements Q up to the accuracy ε and that is alive when interacting with P.



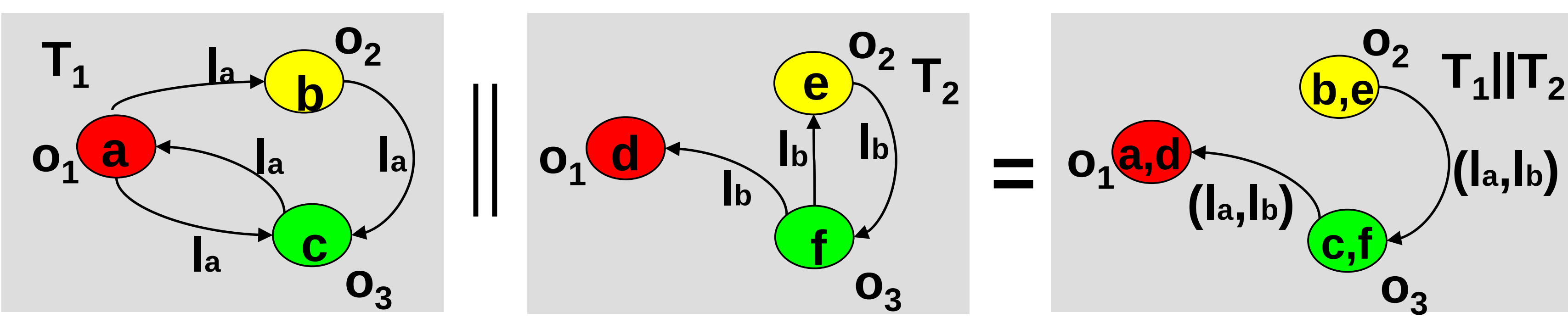
Approximate composition [Tabuada IEEE TAC 08]

Definition Given $T_1 = (Q_1, Q_{01}, L_1, \longrightarrow_1, Q_{m1}, O_1, H_1)$ and $T_2 = (Q_2, Q_{02}, L_2, \longrightarrow_2, Q_{m2}, O_2, H_2)$, with $O_1 = O_2$, and an accuracy $\theta > 0$, the approximate composition of T_1 and T_2 is the system

$$T = T_1 \,||_\theta\, T_2 = (Q, Q_0, L, \longrightarrow, Q_m, O, H)$$

where:

- $Q = \{(q_1, q_2) \in Q_1 \times Q_2 : d(H_1(q_1), H_2(q_2)) \leq \theta\}$
- $Q_0 = Q \cap (Q_{01} \times Q_{02})$
- $L = L_1 \times L_2$
- $(q_1, q_2) \xrightarrow{(l_1, l_2)} (p_1, p_2)$, if $q_1 \xrightarrow{l_1} p_1$ and $q_2 \xrightarrow{l_2} p_2$
- $Q_m = Q \cap (Q_{m1} \times Q_{m2})$
- $O = O_1 = O_2$
- $H(q_1, q_2) = H_1(q_1)$

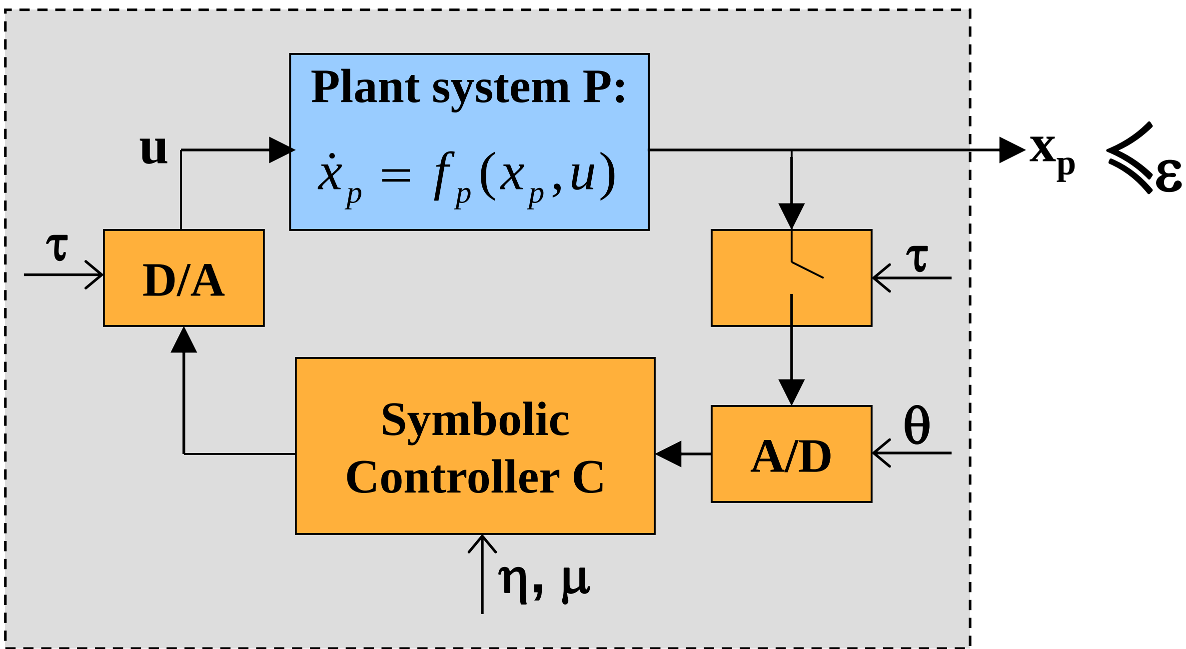


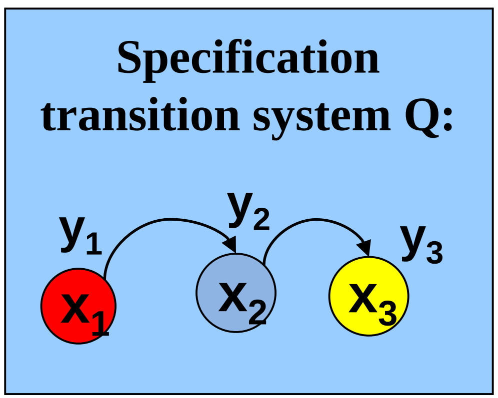
Design of symbolic controllers

Control problem

Given a plant P, a deterministic specification Q and a desired accuracy $\varepsilon > 0$, find a symbolic controller C such that

1. $T_\tau(P) \|_\theta C \preccurlyeq_\varepsilon Q$
2. $T_\tau(P) \|_\theta C$ is alive

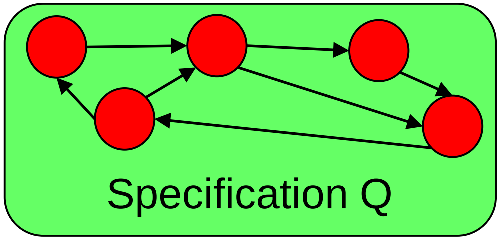


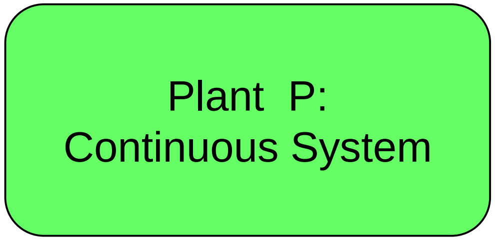


Solution

Synthesis through a three-step process:

1. Compute the symbolic model $T_{\tau,\eta,\mu}(P)$ of P
2. Compute the symbolic controller $C^* = T_{\tau,\eta,\mu}(P) ||_{\eta} Q$
3. Compute the alive part Alive(C*) of C*

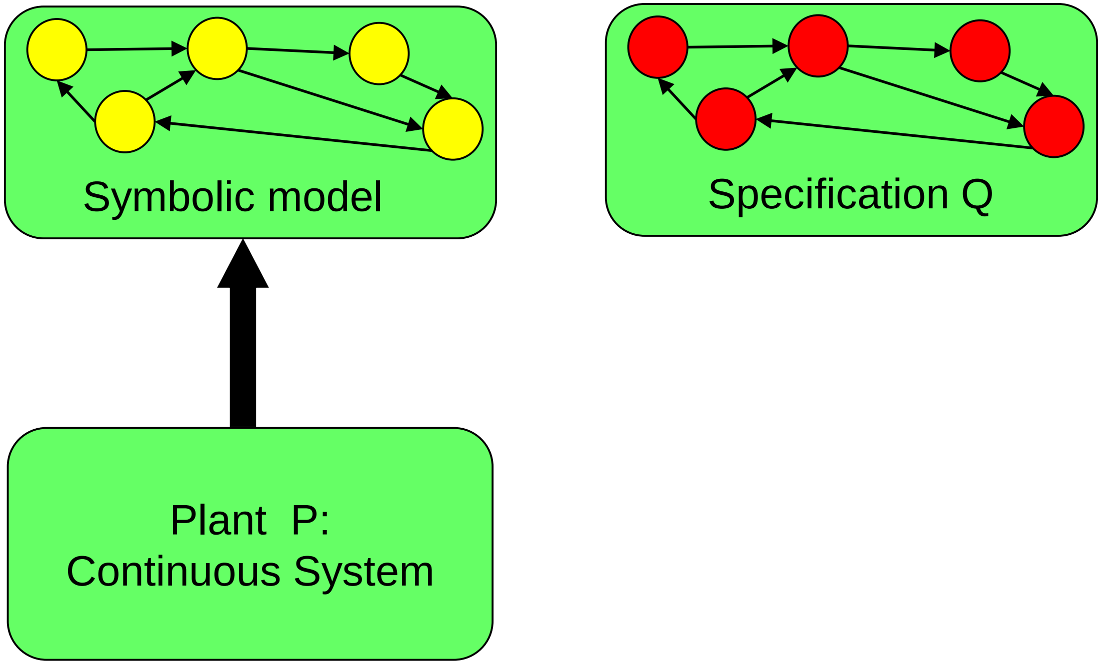




Solution

Synthesis through a three-step process:

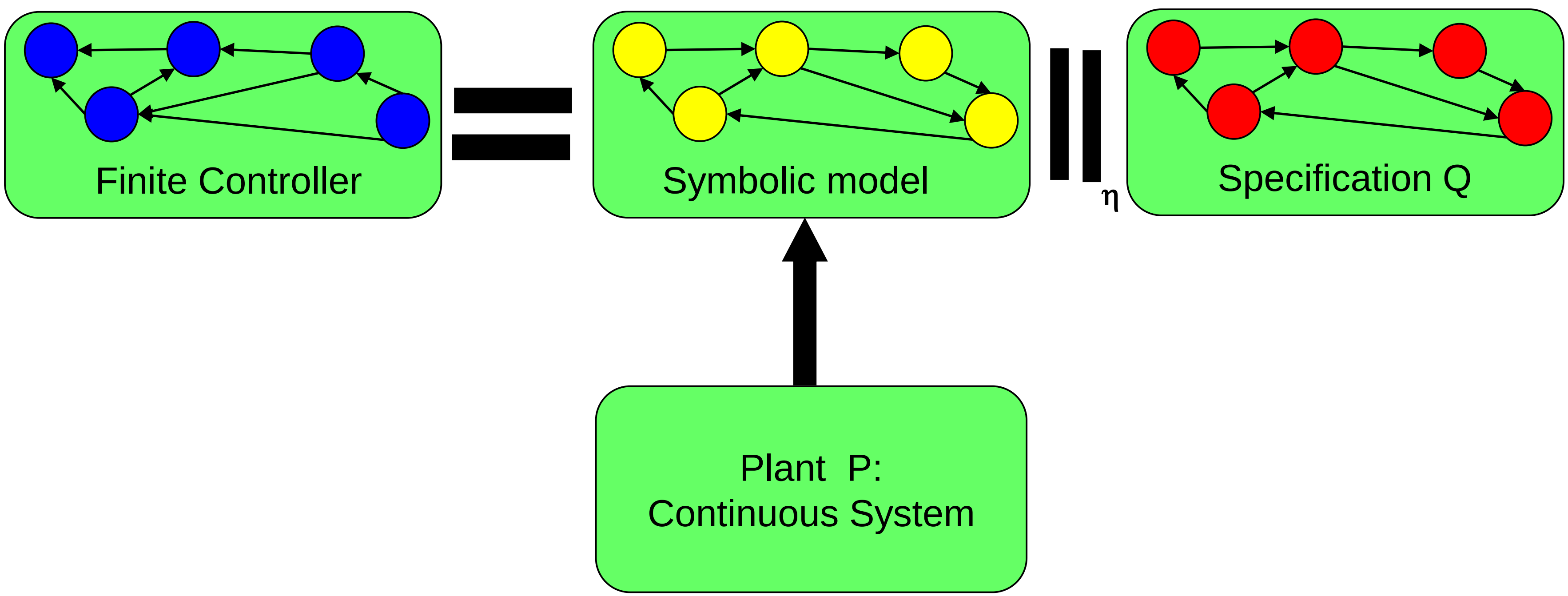
1. Compute the symbolic model $T_{\tau,\eta,\mu}(P)$ of P
2. Compute the symbolic controller $C^* = T_{\tau,\eta,\mu}(P) ||_{\eta} Q$
3. Compute the alive part Alive(C*) of C*



Solution

Synthesis through a three-step process:

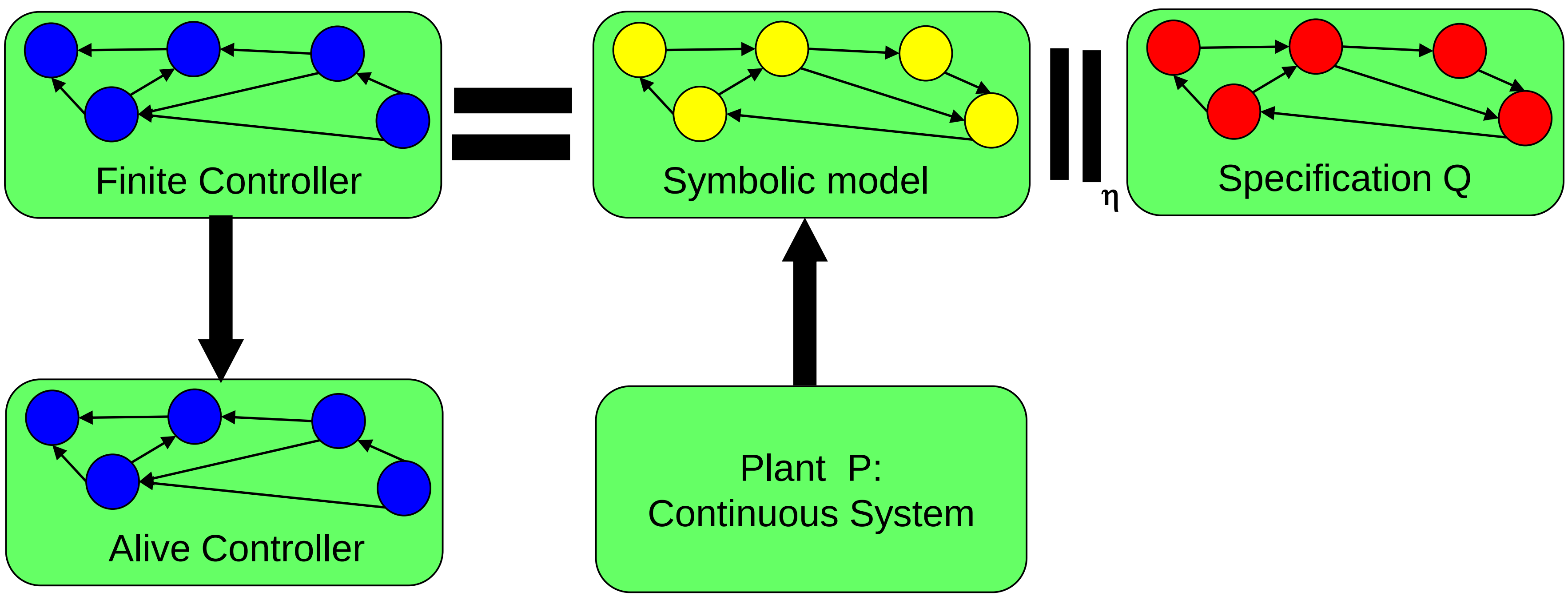
1. Compute the symbolic model $T_{\tau,\eta,\mu}(P)$ of P
2. Compute the symbolic controller $C* = T_{\tau,\eta,\mu}(P) ||_{\eta} Q$
3. Compute the alive part Alive(C*) of C*



Solution

Synthesis through a three-step process:

1. Compute the symbolic model $T_{\tau,\eta,\mu}(P)$ of P
2. Compute the symbolic controller $C* = T_{\tau,\eta,\mu}(P)||_{\eta}\, Q$
3. Compute the alive part Alive(C*) of C*



Solution

Synthesis through a three-step process:

1. Compute the symbolic model $T_{\tau,\eta,\mu}(P)$ of P
2. Compute the symbolic controller $C* = T_{\tau,\eta,\mu}(P) ||_{\eta} Q$
3. Compute the alive part $Alive(C*)$ of $C*$

Theorem Suppose that P is δ–ISS and choose parameters $\tau, \eta, \mu, \theta > 0$ satisfying:

$$\beta(\theta,\tau) + \gamma(\mu) + 2\eta \leq \theta + \eta \leq \varepsilon$$

The symbolic controller $Alive(C*)$ solves the control problem.

Design of symbolic controllers

Drawbacks

- It considers the whole sets of states of $T_{\tau,\eta,\mu}(P)$ and Q
- For any source state x and target state y, it includes all transitions $x \xrightarrow{u} y$ with any control input u by which state x reaches state y
- It first constructs $T_{\tau,\eta,\mu}(P)$ and Q, then $C*$, to finally eliminate blocking states from $C*$

To cope with space and time complexity, instead of computing separately

(1) Discrete abstraction $T_{\tau,\eta,\mu}(P)$ of P
(2) Symbolic controller $C^* = T_{\tau,\eta,\mu}(P) \|_{\eta} Q$
(3) Alive part $\mathrm{Alive}(C^*)$ of $C*$

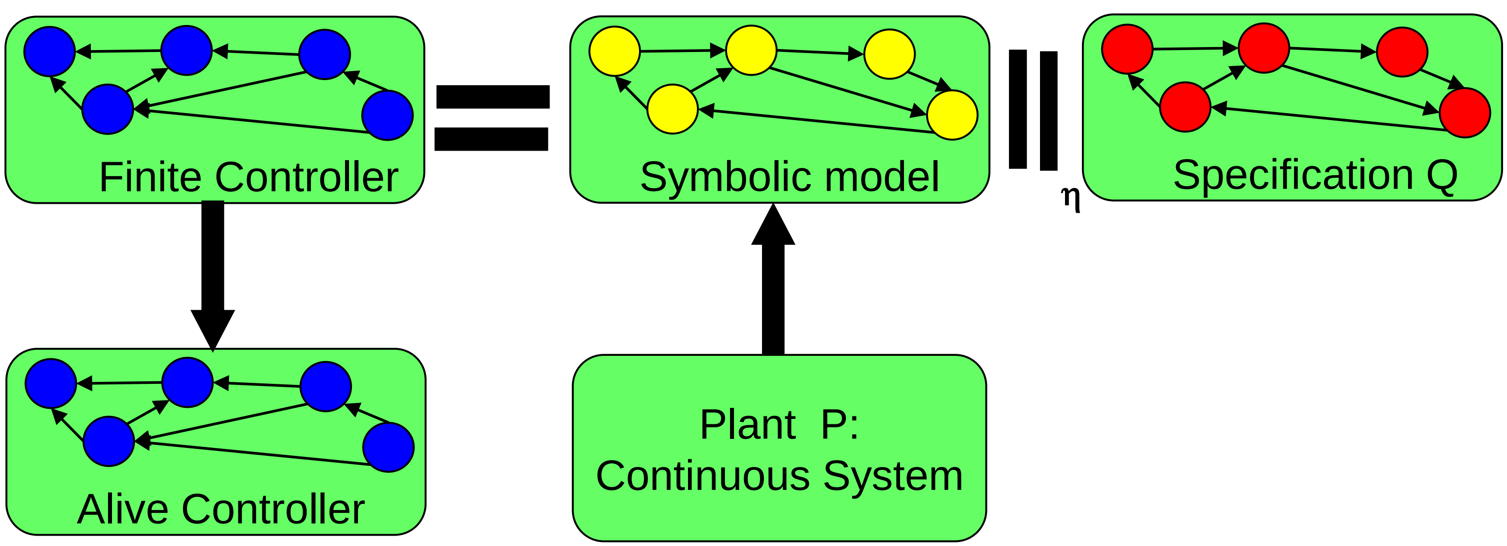
Integrated Approach: Compute (1) + (2) + (3) at once!

Space/time complexity analysis of the proposed algorithm formally quantifies the gain of the integrated approach

Integrated Algorithm

Basic ideas

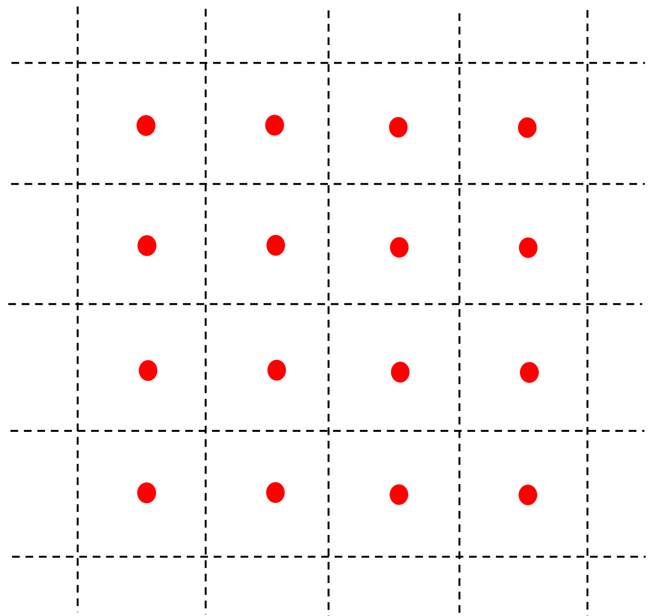
1. It only considers the intersection of the accessible parts of P and Q
2. For any given source state x and target state y, it considers only one transition (x,u,y)
3. It eliminates blocking states as soon as they show up



Integrated Algorithm

How does it work?

First, we consider the target space as the intersection of the sets of initial states of $T_{\tau,\eta,\mu}(P)$ and Q.

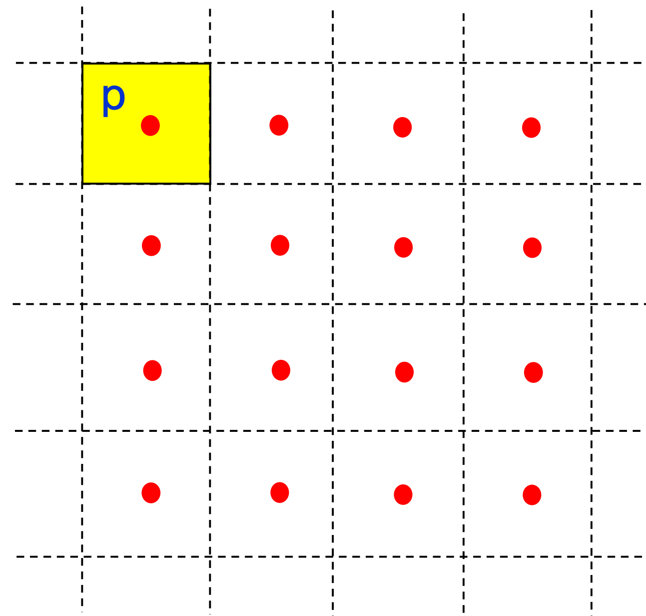


Integrated Algorithm

How does it work?

First, we consider the target space as the intersection of the sets of initial states of $T_{\tau,\eta,\mu}(P)$ and Q.

Pick a "symbolic" state p from the target space and compute the unique state q such that the transition $p \longrightarrow q$ is in Q.

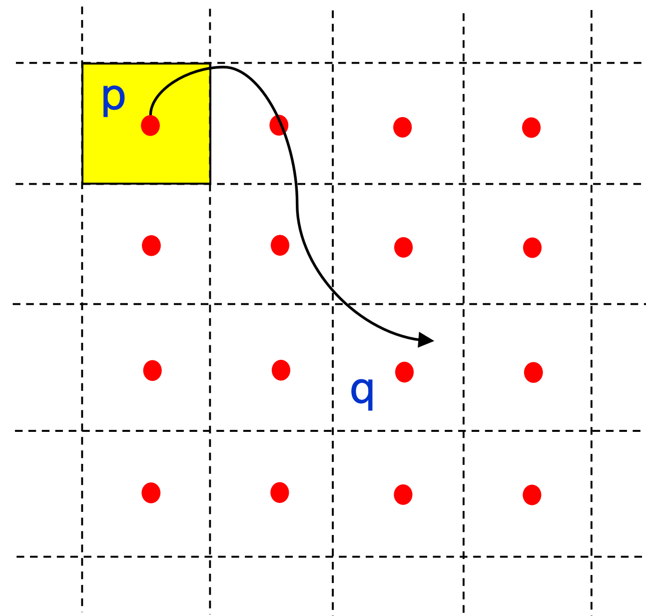


Integrated Algorithm

How does it work?

First, we consider the target space as the intersection of the sets of initial states of $T_{\tau,\eta,\mu}(P)$ and Q.

Pick a "symbolic" state p from the target space and compute the unique state q such that the transition $p \longrightarrow q$ is in Q.

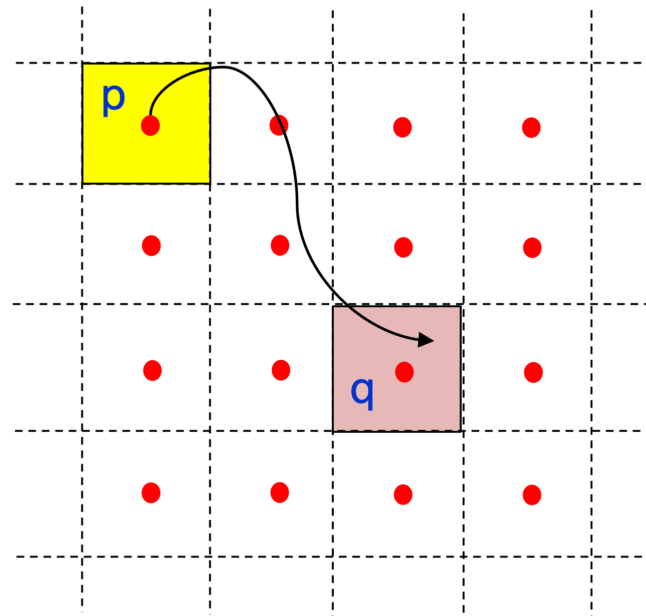


Integrated Algorithm

How does it work?

First, we consider the target space as the intersection of the sets of initial states of $T_{\tau,\eta,\mu}(P)$ and Q.

Pick a "symbolic" state p from the target space and compute the unique state q such that the transition $p \longrightarrow q$ is in Q.



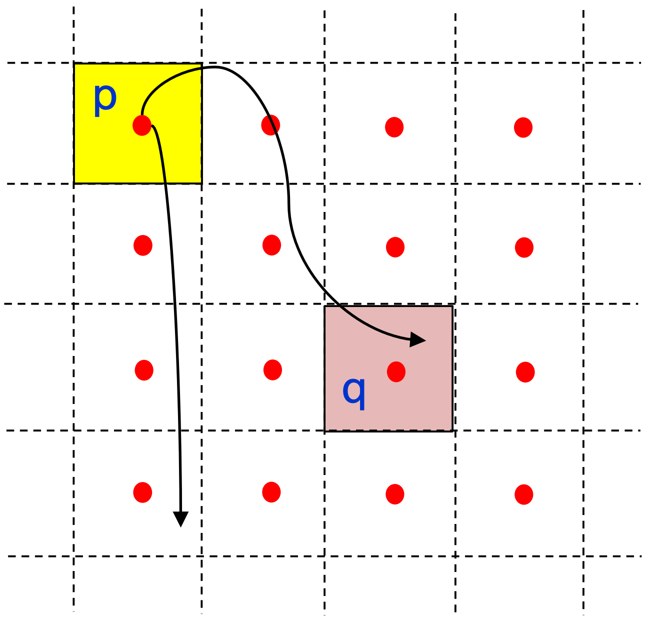
Integrated Algorithm

How does it work?

First, we consider the target space as the intersection of the sets of initial states of $T_{\tau,\eta,\mu}(P)$ and Q.

Pick a "symbolic" state p from the target space and compute the unique state q such that the transition $p \longrightarrow q$ is in Q.

Pick control inputs in $[U]_{2\mu}$ and integrate the plant differential equation until $q=[x(\tau,p,u)]_{2\eta}$ for some u.



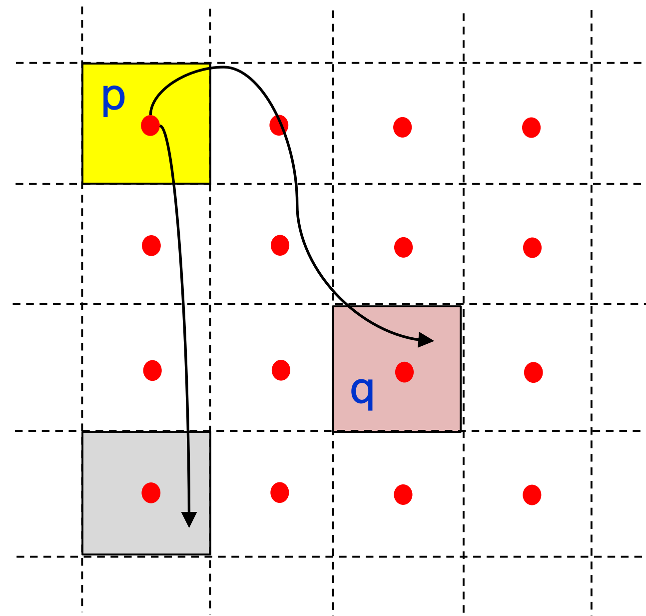
Integrated Algorithm

How does it work?

First, we consider the target space as the intersection of the sets of initial states of $T_{\tau,\eta,\mu}(P)$ and Q.

Pick a "symbolic" state p from the target space and compute the unique state q such that the transition $p \longrightarrow q$ is in Q.

Pick control inputs in $[U]_{2\mu}$ and integrate the plant differential equation until $q=[x(\tau,p,u)]_{2\eta}$ for some u.



No matching! Try another input!

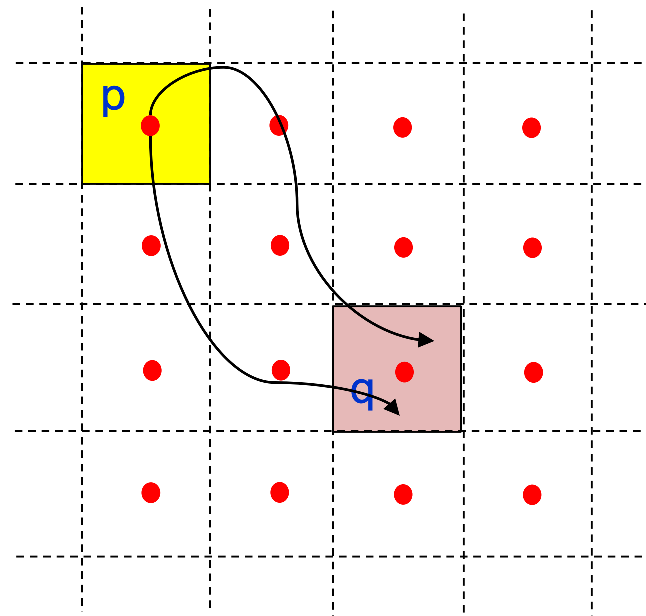
Integrated Algorithm

How does it work?

First, we consider the target space as the intersection of the sets of initial states of $T_{\tau,\eta,\mu}(P)$ and Q.

Pick a “symbolic” state p from the target space and compute the unique state q such that the transition $p \longrightarrow q$ is in Q.

Pick control inputs in $[U]_{2\mu}$ and integrate the plant differential equation until $q=[x(\tau,p,u)]_{2\eta}$ for some u.



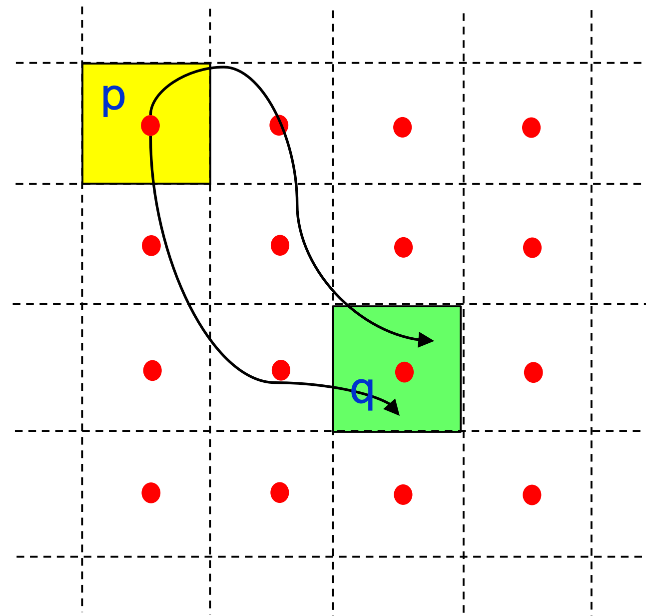
Integrated Algorithm

How does it work?

First, we consider the target space as the intersection of the sets of initial states of $T_{\tau,\eta,\mu}(P)$ and Q.

Pick a "symbolic" state p from the target space and compute the unique state q such that the transition $p \longrightarrow q$ is in Q.

Pick control inputs in $[U]_{2\mu}$ and integrate the plant differential equation until $q=[x(\tau,p,u)]_{2\eta}$ for some u.



Matching found!!

Integrated Algorithm

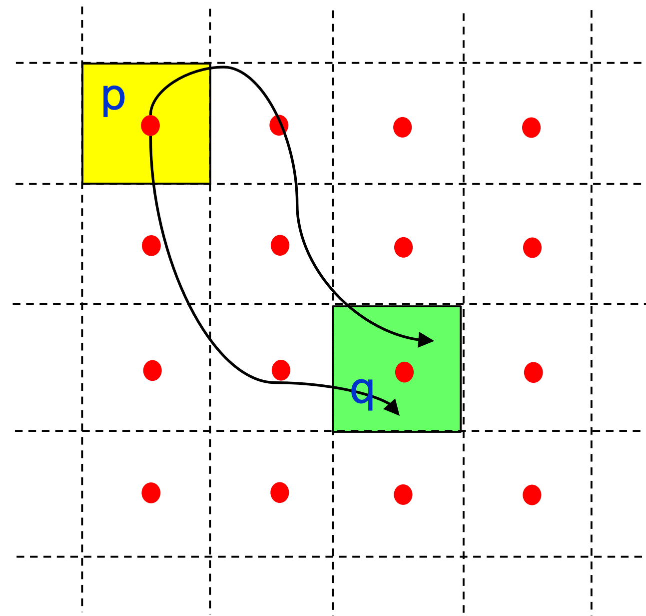
How does it work?

First, we consider the target space as the intersection of the sets of initial states of $T_{\tau,\eta,\mu}(P)$ and Q.

Pick a "symbolic" state p from the target space and compute the unique state q such that the transition $p \longrightarrow q$ is in Q.

Pick control inputs in $[U]_{2\mu}$ and integrate the plant differential equation until $q=[x(\tau,p,u)]_{2\eta}$ for some u.

Add the transition (p,u,q) to the controller.
Replace p with q in the target space.



Matching found!!

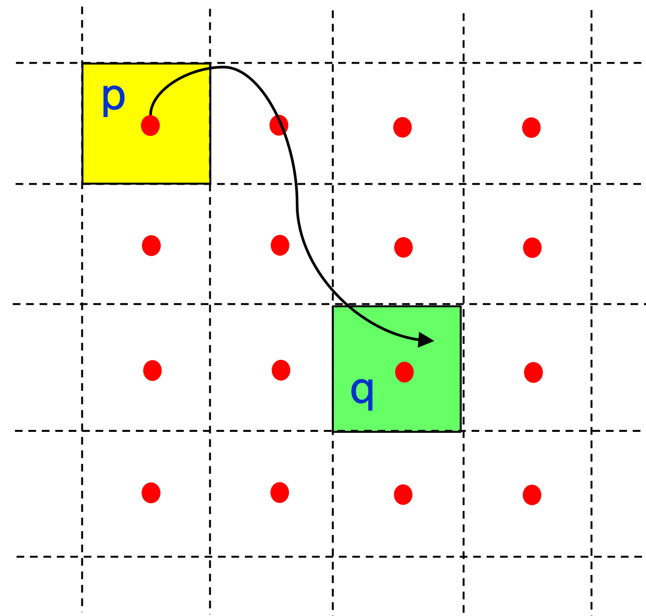
Integrated Algorithm

How does it work?

First, we consider the target space as the intersection of the sets of initial states of $T_{\tau,\eta,\mu}(P)$ and Q.

Pick a "symbolic" state p from the target space and compute the unique state q such that the transition $p \longrightarrow q$ is in Q.

Pick control inputs in $[U]_{2\mu}$ and integrate the plant differential equation until $q=[x(\tau,p,u)]_{2\eta}$ for some u.



Matching not found!!

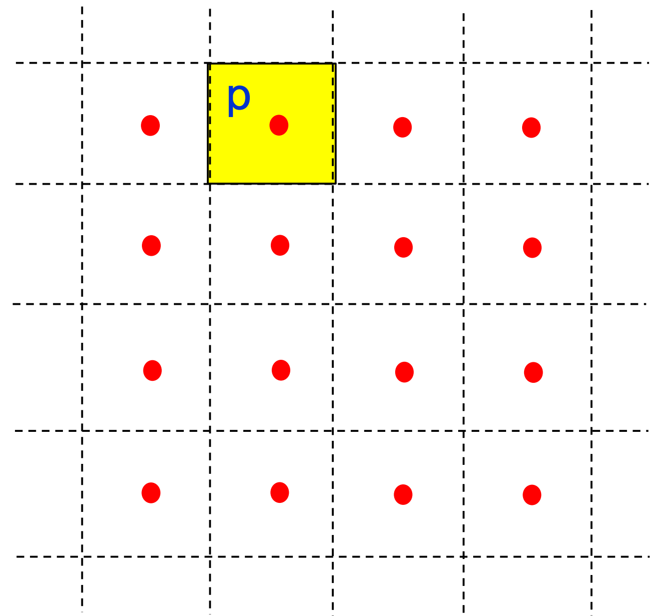
If any "good" input does not exist, then p is **blocking**!
A backwards procedure is executed to eliminate p and all its ingoing transitions from the controller, until a controller is found which is alive.

Integrated Algorithm

Successive iterations:

Repeat the procedure for all the target states.

The algorithm terminates when there are no more target states to be visited.

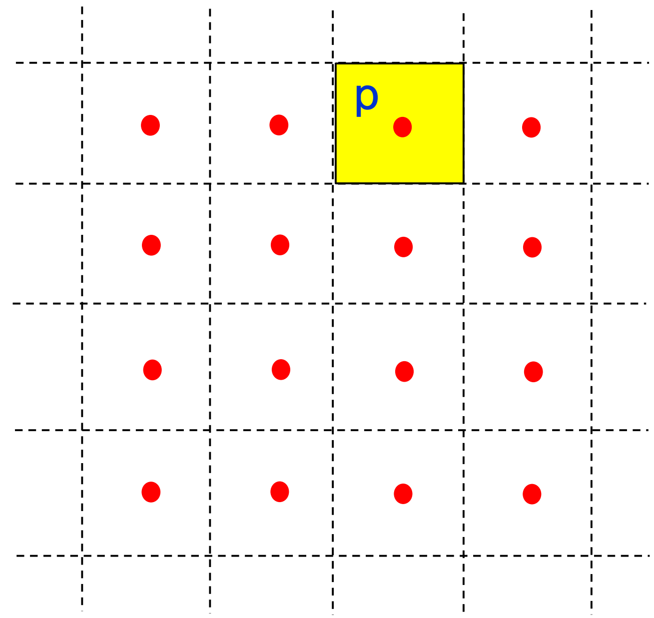


Integrated Algorithm

Successive iterations:

Repeat the procedure for all the target states.

The algorithm terminates when there are no more target states to be visited.

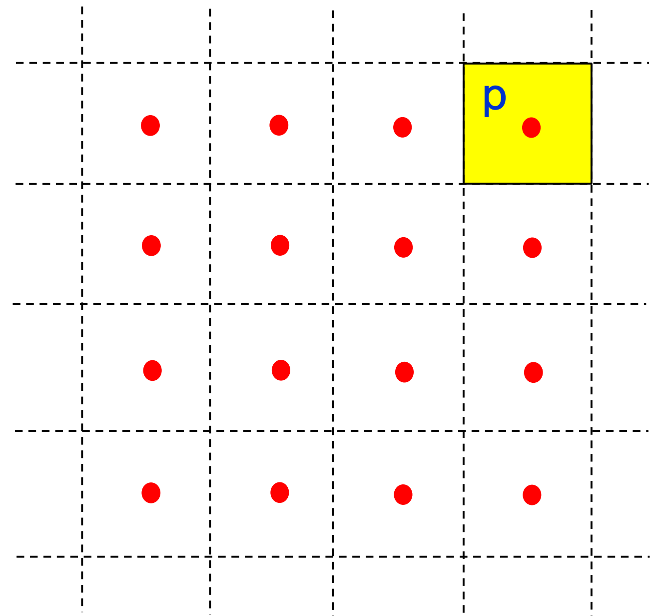


Integrated Algorithm

Successive iterations:

Repeat the procedure for all the target states.

The algorithm terminates when there are no more target states to be visited.



Integrated Algorithm

Properties

Let C** be the outcome of the integrated procedure:

1. The integrated algorithm terminates in a finite number of steps
2. C** and Alive(C*) are exactly bisimilar ⇨ C** solves the control problem
3. C** is the minimal 0−bisimilar system of Alive(C*)
4. C** is accessible
5. space/time complexity of the integrated procedure is not larger than the one of the classical procedure

Integrated symbolic control design

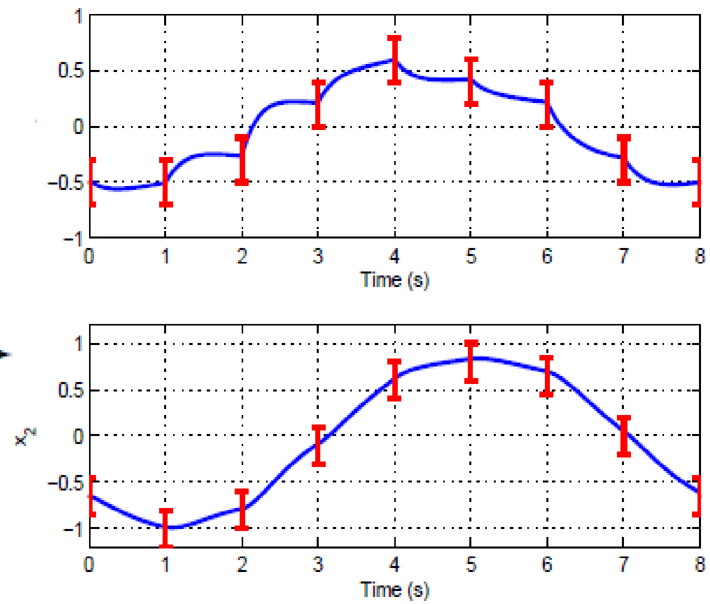
Example

Plant

$$P : \begin{cases} \dot{x}_1 = -4x_1 + x_2^2 - u \\ \dot{x}_2 = 2x_1 - 7\sin x_2 \end{cases}$$

Specification

$(-0.5, -0.65) \longrightarrow (-0.5, -1) \longrightarrow$
$(-0.3, -0.8) \longrightarrow (0.2, -0.1) \longrightarrow$
$(0.6, 0.6) \longrightarrow (0.4, 0.8) \longrightarrow$
$(0.2, 0.65) \longrightarrow (-0.3, 0) \longrightarrow$
$(-0.5, -0.65)$



Accuracy ε = 0.2

Comparison between Alive(C*) and C**	Alive(C*)	C**	Gain
Max memory occupation (no. of transitions)	2,759,580	48	$5.7 \cdot 10^4$
Time (s)	5,442	13	$4.2 \cdot 10^2$