Controllo di assetto di un veicolo mediante modelli simbolici

Alessandro Borri [a,b], Giordano Pola [b], Maria Domenica Di Benedetto[b]

[a] Istituto di Analisi dei Sistemi ed Informatica “A. Ruberti” (IASI)
Consiglio Nazionale delle Ricerche (CNR), Roma

[b] Dipartimento di Ingegneria e Scienze dell’Informazione e Matematica
Centro di Eccellenza per la Ricerca DEWS
Università degli Studi dell’Aquila, L’Aquila

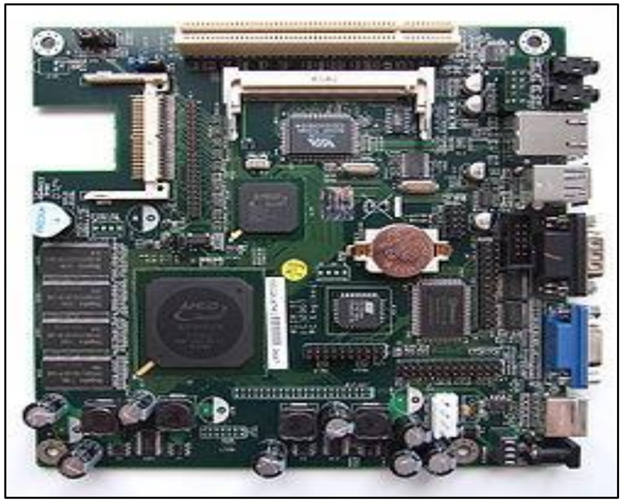
alessandro.borri@univaq.it
giordano.pola@univaq.it
mariadomenica.dibenedetto@univaq.it

Outline

- Modello del veicolo
- Tecniche di controllo simbolico per sistemi non lineari

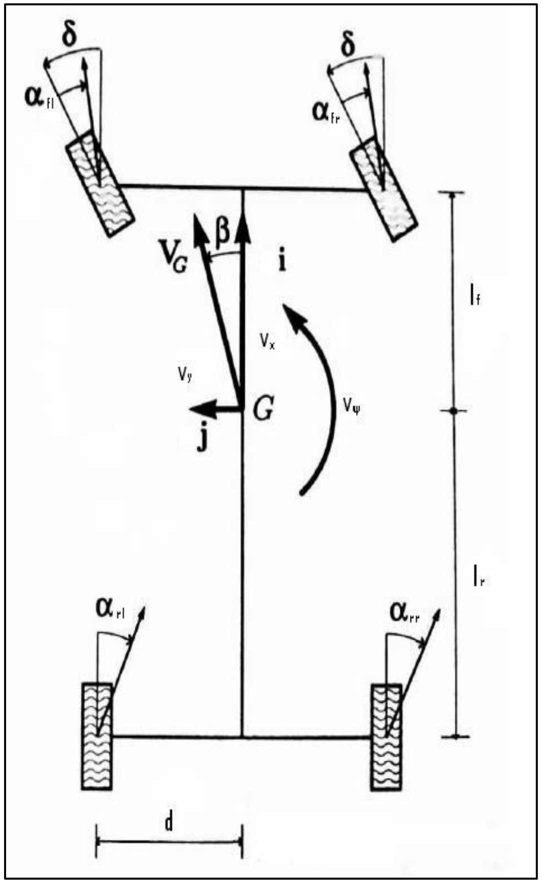
Motivazioni

- I veicoli moderni sono controllati elettronicamente per assicurare la stabilità del mezzo e per fornire un sempre maggiore comfort al guidatore e ai passeggeri.
- Poiché le centraline (ECU) delle auto elaborano segnali digitali, un approccio diffuso consiste nell'elaborare leggi di controllo di tipo continuo e poi procedere a campionamento (nel tempo) e discretizzazione (sulle ampiezze) per ottenere segnali digitali.
- Il comportamento dei veicoli è validato in maniera puramente sperimentale su un numero finito di manovre, senza fornire garanzie di tipo più generale.



Motivazioni

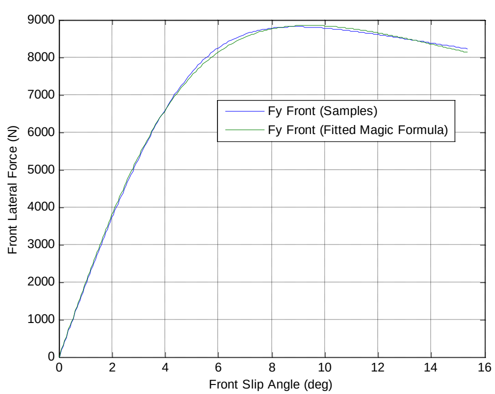
- Un approccio alternativo al controllo veicolo prevede l'utilizzo di metodi formali e controllori simbolici (finiti).
- La sintesi del controllore avviene direttamente nel dominio discreto (cioè la legge di controllo è digitale già in partenza) e le prestazioni sono garantite in modo formale con un'accuratezza decidibile a priori.
- Specifiche possibili: stabilità, inseguimento di traiettoria, ecc.
- Vantaggi:
  - ✓ Controllo quantizzato "in partenza "
  - ✓ Gestione nativa delle saturazioni
- Svantaggi:
  - X Complessità computazionale

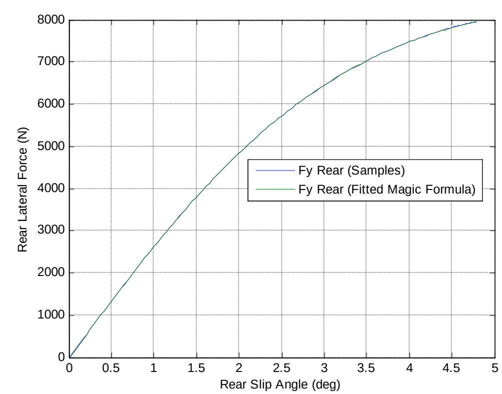


Modello di veicolo

- Modello monotraccia («a bicicletta») a 2 gradi di libertà (stati):
  - Velocità di imbardata (yaw rate) ω_z
  - Velocità laterale v_y

- Il modello è non lineare: le forze degli pneumatici sono funzioni non lineari degli angoli «di deriva» delle ruote





Modello di veicolo

- Controllo tramite **Active Front Steering (AFS)**, che è un attuatore in grado di fornire sulle ruote anteriori un angolo di correzione rispetto a quello imposto dal guidatore.

- L'azione del controllo è limitata per garantire la guidabilità e il comfort del guidatore e dei passeggeri.

- La dinamica dell'attuatore è assunta lineare e si comporta come filtro passa-basso rispetto alla legge di controllo digitale (costante a tratti)

Modello di veicolo

$$\dot{\omega}_z = \frac{F_{y,f}(\alpha_f) l_f - F_{y,r}(\alpha_r) l_r}{J}$$

yaw rate

$$\dot{v}_y = -v_x \omega_z + \frac{F_{y,f}(\alpha_f) + F_{y,r}(\alpha_r)}{m}$$

lateral velocity

$$\dot{\delta}_{AFS} = -\frac{1}{\tau_{AFS}} \delta_{AFS} + \frac{1}{\tau_{AFS}} u_c$$

AFS actuation

$$\alpha_f = \delta_{AFS} + \delta_{DRI} - \frac{v_y + l_f \omega_z}{v_x}$$

$$\alpha_r = -\frac{v_y - l_r \omega_z}{v_x}$$

δ_{DRI} = angolo driver

α_f, α_r= angoli di deriva

l_f, l_r= semipassi

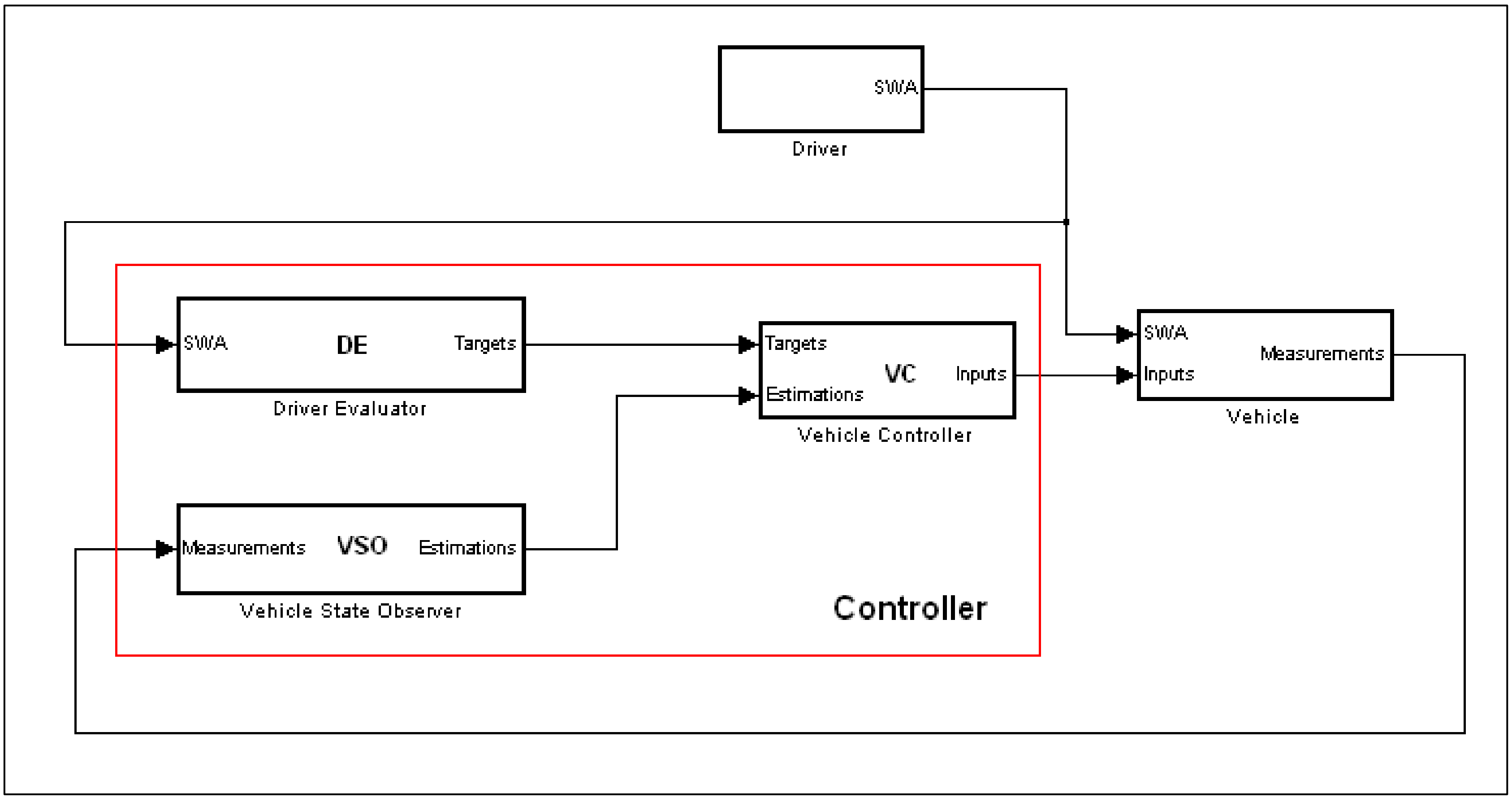
v_x= velocità di avanzamento

m= massa veicolo

J = momento di inerzia veicolo

τ_{AFS} = costante di tempo attuatore

Architettura di controllo veicolo



Assunzioni:

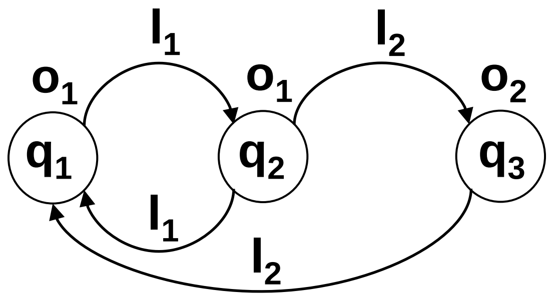
- Stato misurabile (non consideriamo il design dell'osservatore).
- Il driver evaluator (o riferimento) è un sistema (assunto stabile) che modella il comportamento desiderato dal guidatore e prende come input l'ingresso del guidatore (imposto tramite lo sterzo).
- Il controllore simbolico, misurando l'azione del driver e lo stato del riferimento, genera un ingresso di controllo che porta il veicolo vicino il più possibile al comportamento desiderato.
- La stabilità del veicolo è garantita mantenendo gli angoli di deriva all'interno del tratto crescente delle caratteristiche dei pneumatici.

Definition A transition system is a tuple:

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

consisting of:

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



- T is said countable if Q and L are countable sets
- T is said symbolic/finite if Q and L are finite sets
- T is non-blocking if for $q \in Q$ there exists $l \in L$ and $q' \in Q$ such that $(q,l,q') \in \longrightarrow$
- T is deterministic if for any $q \in Q$ and any $l \in L$ there exists at most one state $q' \in Q$ s.t. $(q,l,q') \in \longrightarrow$

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

Review: Transition Systems modeling Control Systems

A **nonlinear control system** Σ

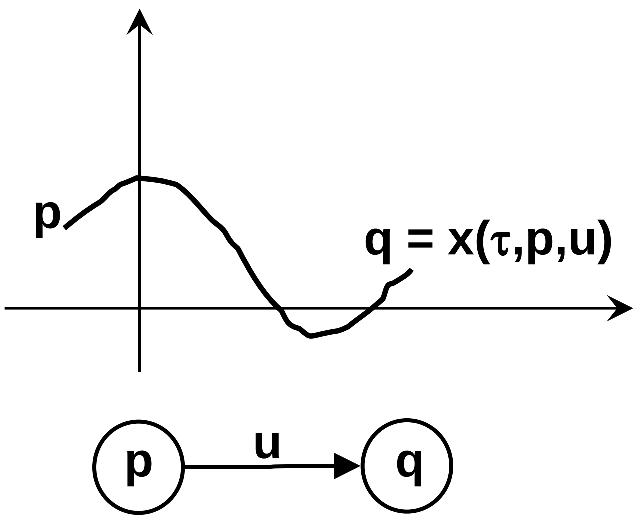
$$dx/dt = f(x,u),\ x \in X \subseteq R^n,\ u \in U \subseteq R^m$$

can be modeled by the transition system

$$T(\Sigma) = (X, X_0, U, \longrightarrow, Y, H),$$

where:

- $X_0 = X$ is a set of initial states
- U is the collection of control signals $u : R \rightarrow U$
- $p \xrightarrow{u} q$, if $x(\tau,p,u) = q$ for some $\tau \geq 0$
- $Y = X$
- H is the identity function



$T(\Sigma)$ captures information contained in Σ but it is not a symbolic model because X and U are infinite sets!

Review: Transition Systems modeling Control Systems

A **nonlinear control system** Σ

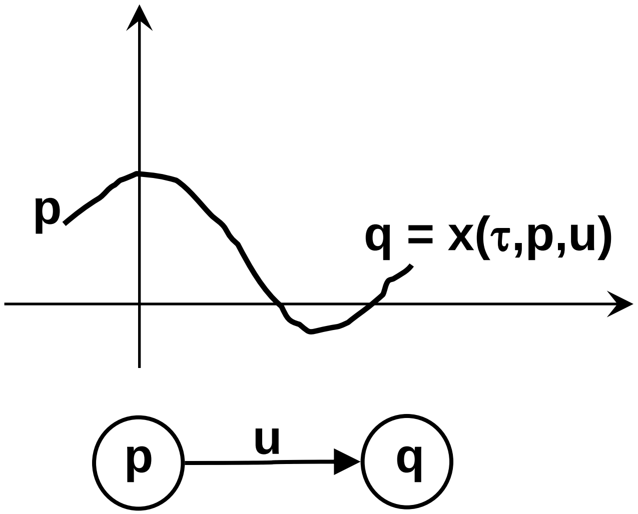
$$dx/dt = f(x,u),\ x \in X \subseteq R^n,\ u \in U \subseteq R^m$$

can be modeled by the transition system

$$T(\Sigma) = (X, X_0, U, \longrightarrow, Y, H),$$

where:

- $X_0 = X$ is a set of initial states
- U is the collection of control signals $u : R \to U$
- $p \xrightarrow{u} q$, if $x(\tau,p,u) = q$ for some $\tau \geq 0$
- $Y = X$
- H is the identity function



$T(\Sigma)$ captures information contained in Σ but it is not a symbolic model because X and U are infinite sets!

Software can be modelled by transition systems

The states are all possible memory configurations and the transition relation describes how the memory contents are changed by the execution of instructions

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

$$T(\Sigma) = (X, X_0, U, \longrightarrow, Y, H),$$

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

$$T_\tau(\Sigma) = (X, X_0, U_\tau, \longrightarrow_\tau, Y, H),$$

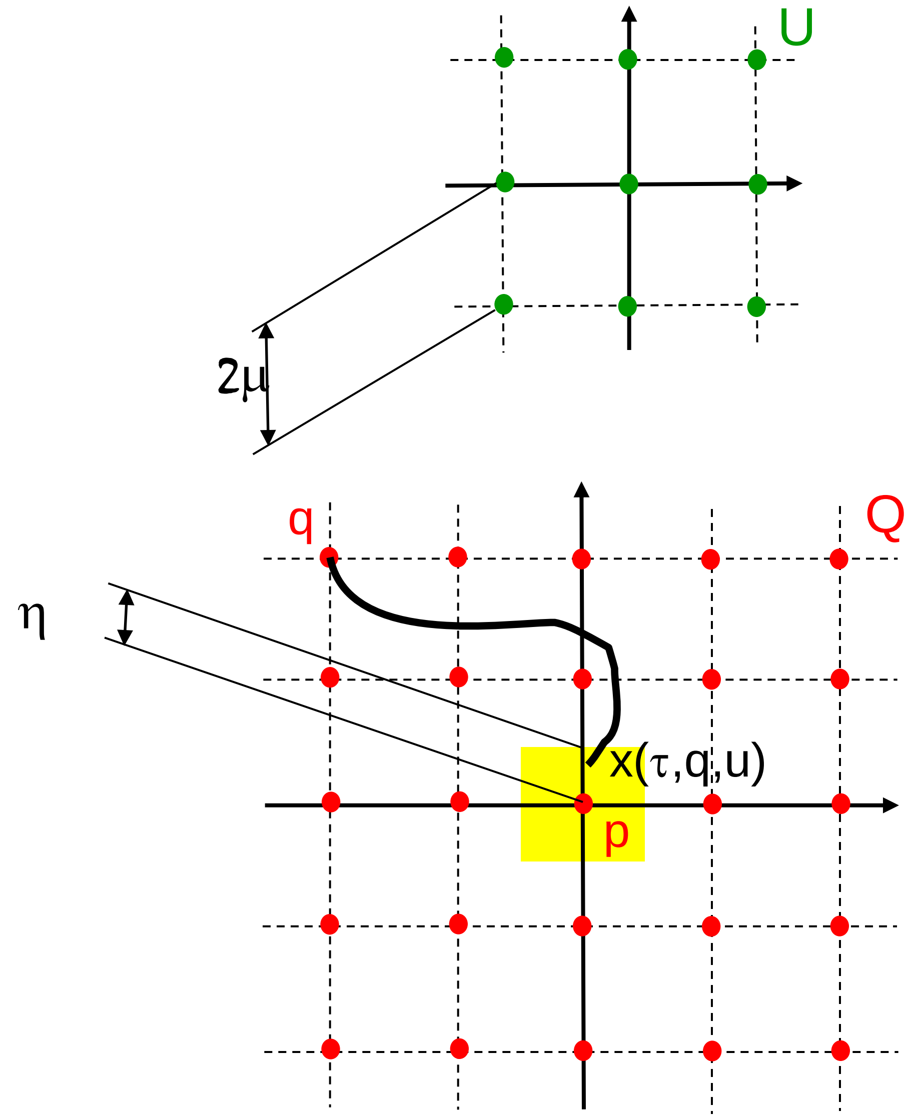
where:

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

Review: Construction of symbolic models

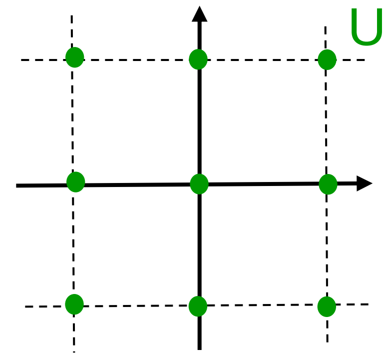
Consider the following parameters:

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



Review: Construction of symbolic models

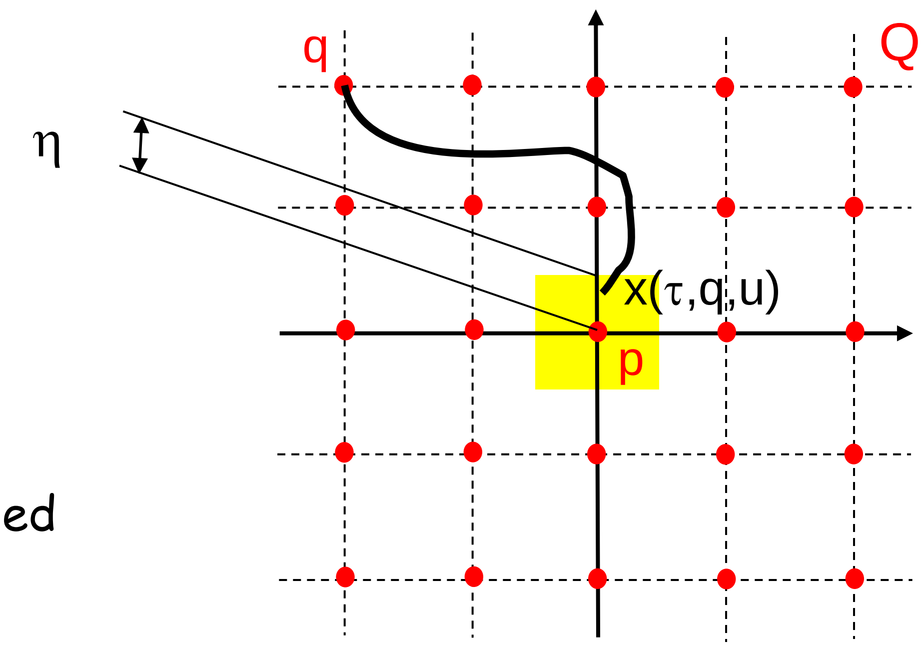
Consider the following parameters:



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

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

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



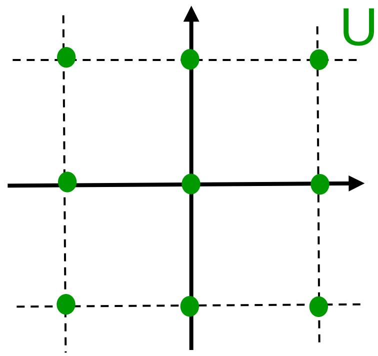
Remark

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

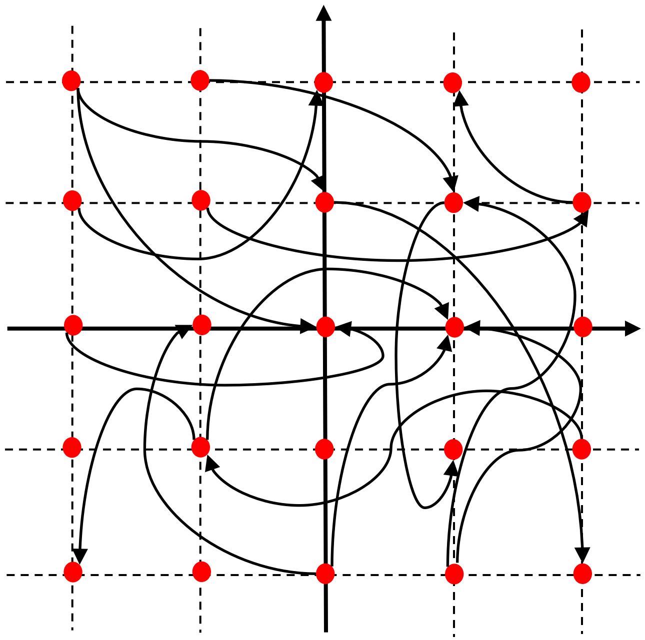
Consider the following parameters:

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



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

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



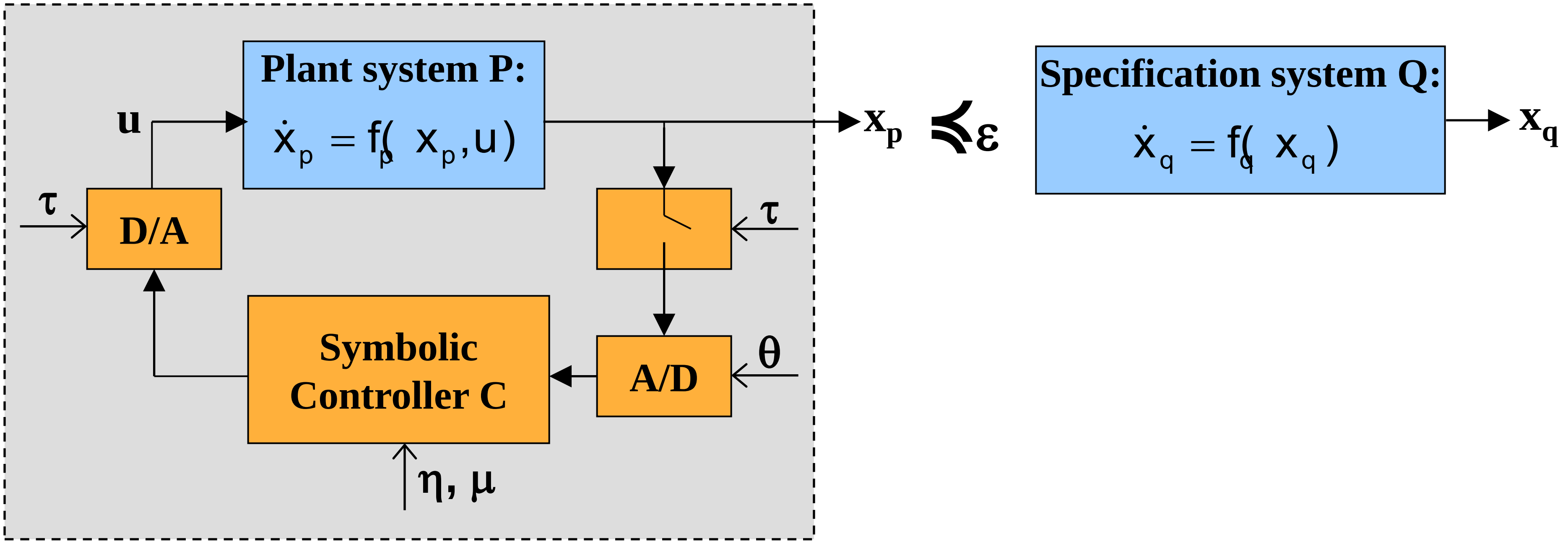
Theorem If Σ is δ-ISS, for any desired precision $\varepsilon > 0$ and for any $\tau, \eta, \mu > 0$ satisfying

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

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

Problem 1: Continuous Specifications [cf. Borri, Pola, Di Benedetto, CDC 2010]

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



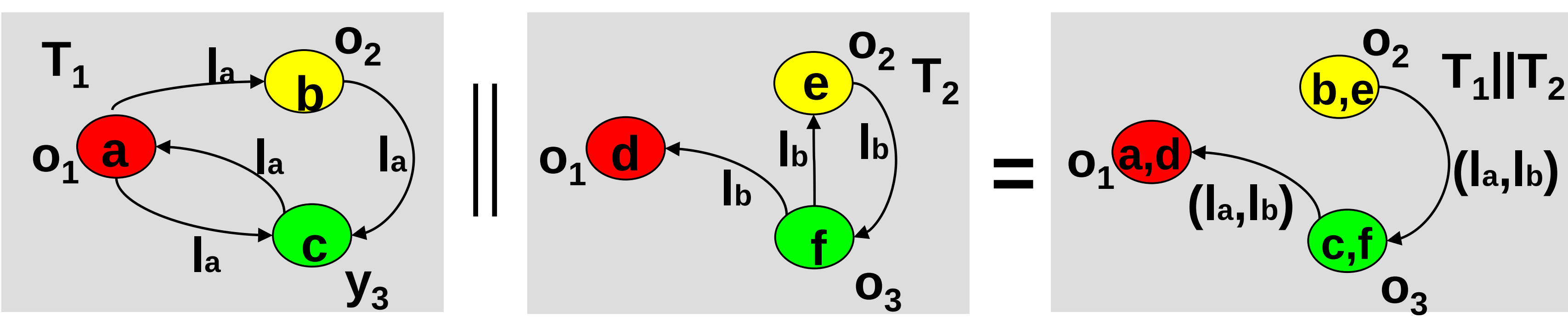
Approximate composition [Tabuada IEEE TAC 08]

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

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

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 $q_2 \xrightarrow{l_2} p_2$
- $O = O_1 = O_2$
- $H(q_1,q_2) = H_1(q_1)$

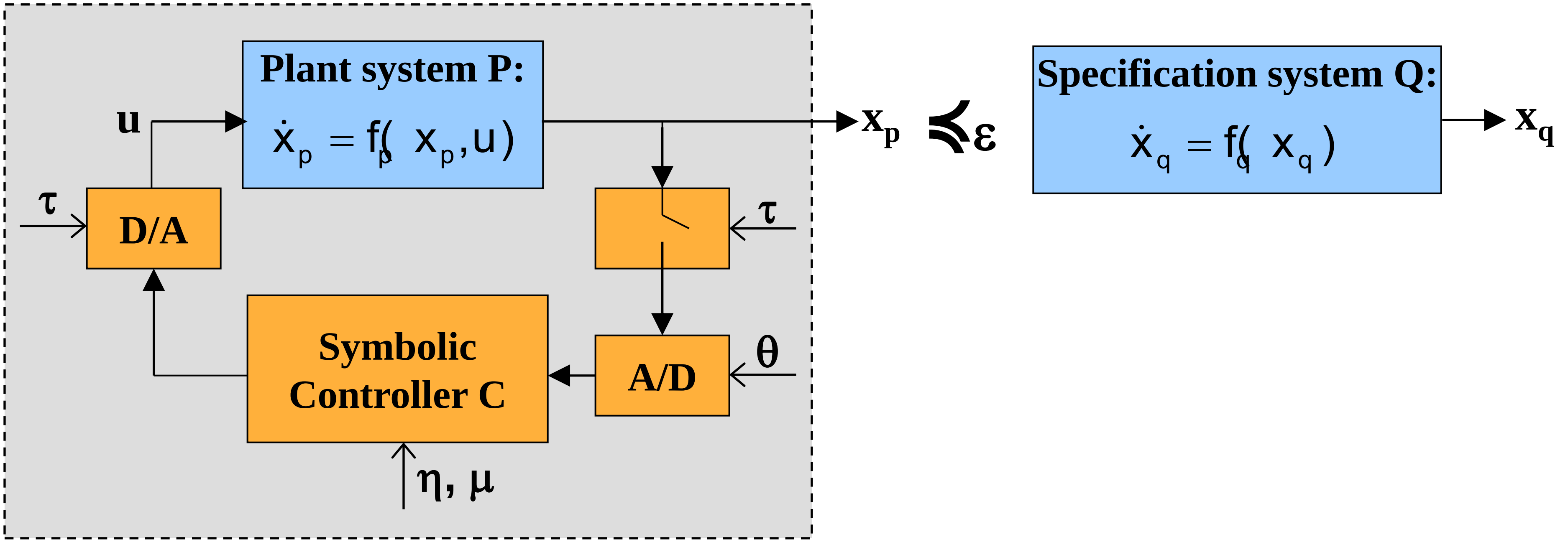


Design of symbolic controllers

Problem 1

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

1. $T_{\tau,\eta,\mu}(P)||_{\theta}C \preccurlyeq_{\varepsilon} T_{\tau,\eta,0}(Q)$
2. $T_{\tau,\eta,\mu}(P)||_{\theta}C$ is non-blocking



Solution of Problem 1

Synthesis through a four-step process:

1. Compute the symbolic model $T_{\tau,\eta,\mu}(P)$ of P
2. Compute the symbolic model $T_{\tau,\eta,0}(Q)$ of Q
3. Compute the symbolic controller $C^* = T_{\tau,\eta,\mu}(P) || T_{\tau,\eta,0}(Q)$
4. Compute the non-blocking part Nb(C^*) of C^*

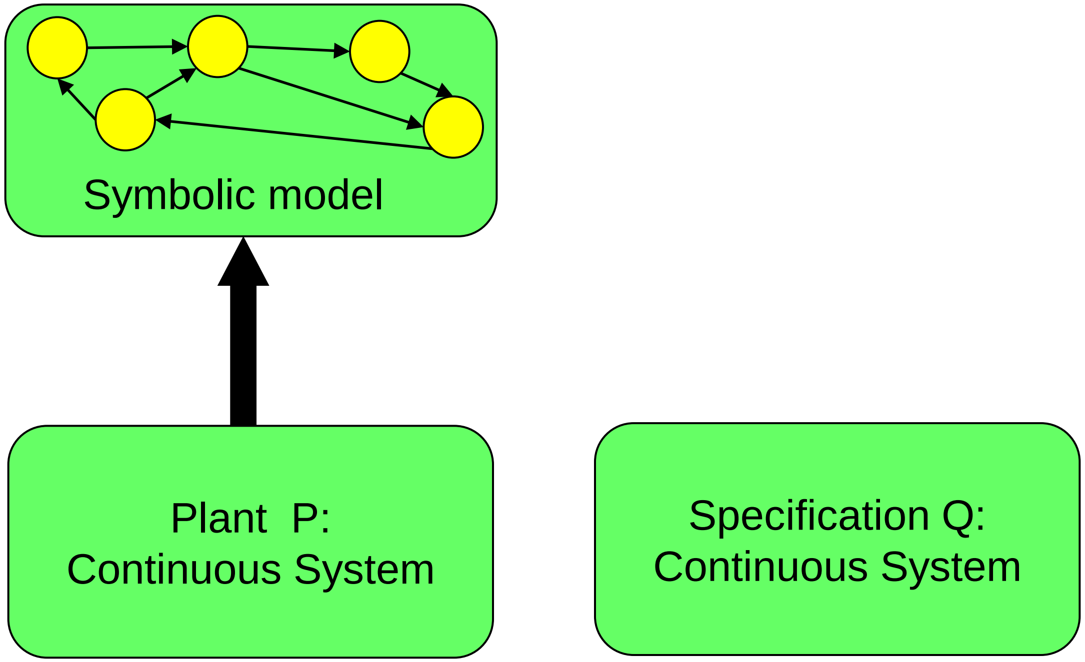
Plant P:
Continuous System

Specification Q:
Continuous System

Solution of Problem 1

Synthesis through a four-step process:

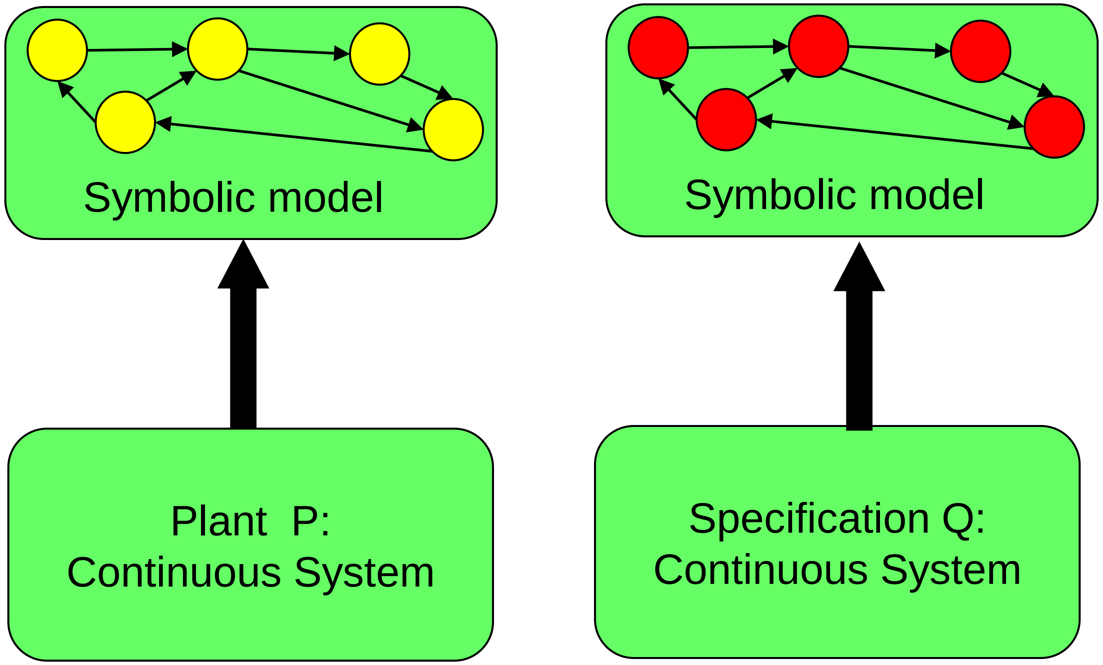
1. Compute the symbolic model $T_{\tau,\eta,\mu}(P)$ of P
2. Compute the symbolic model $T_{\tau,\eta,0}(Q)$ of Q
3. Compute the symbolic controller $C^* = T_{\tau,\eta,\mu}(P) || T_{\tau,\eta,0}(Q)$
4. Compute the non-blocking part Nb(C^*) of C^*



Solution of Problem 1

Synthesis through a four-step process:

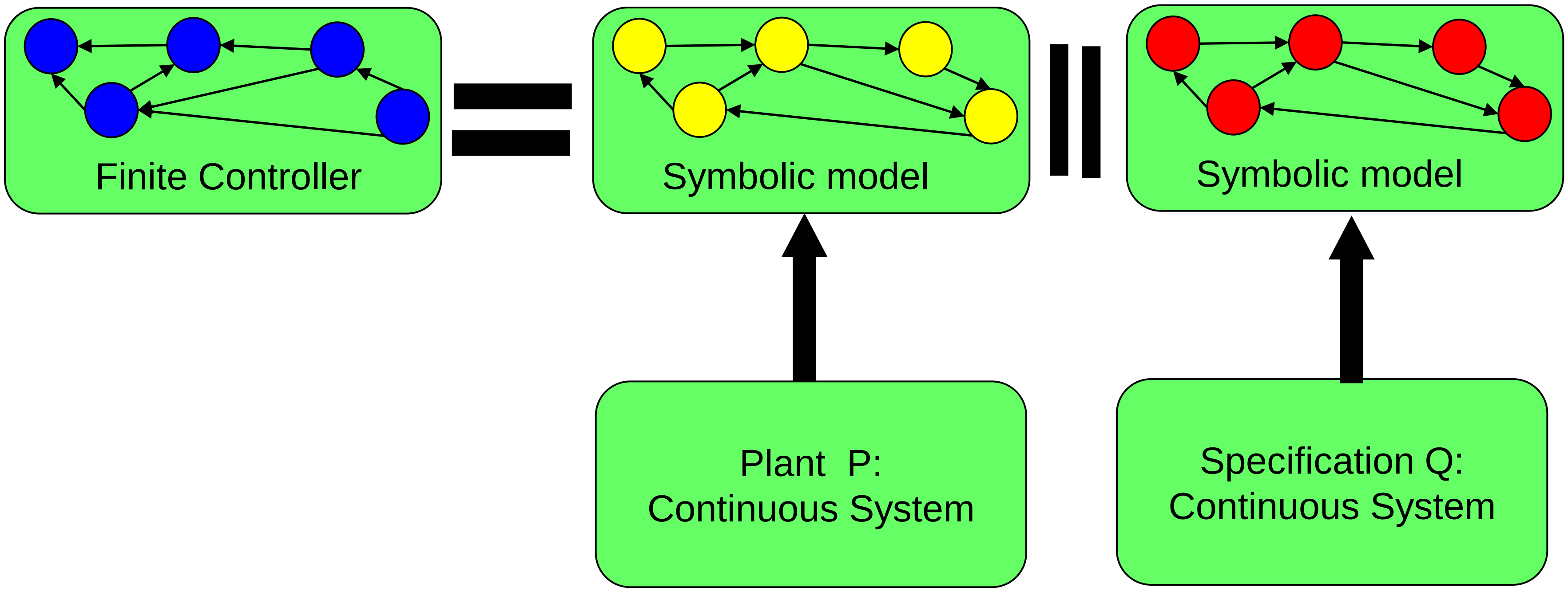
1. Compute the symbolic model $T_{\tau,\eta,\mu}(P)$ of P
2. Compute the symbolic model $T_{\tau,\eta,0}(Q)$ of Q
3. Compute the symbolic controller $C^* = T_{\tau,\eta,\mu}(P) || T_{\tau,\eta,0}(Q)$
4. Compute the non-blocking part $Nb(C^*)$ of C^*



Solution of Problem 1

Synthesis through a four-step process:

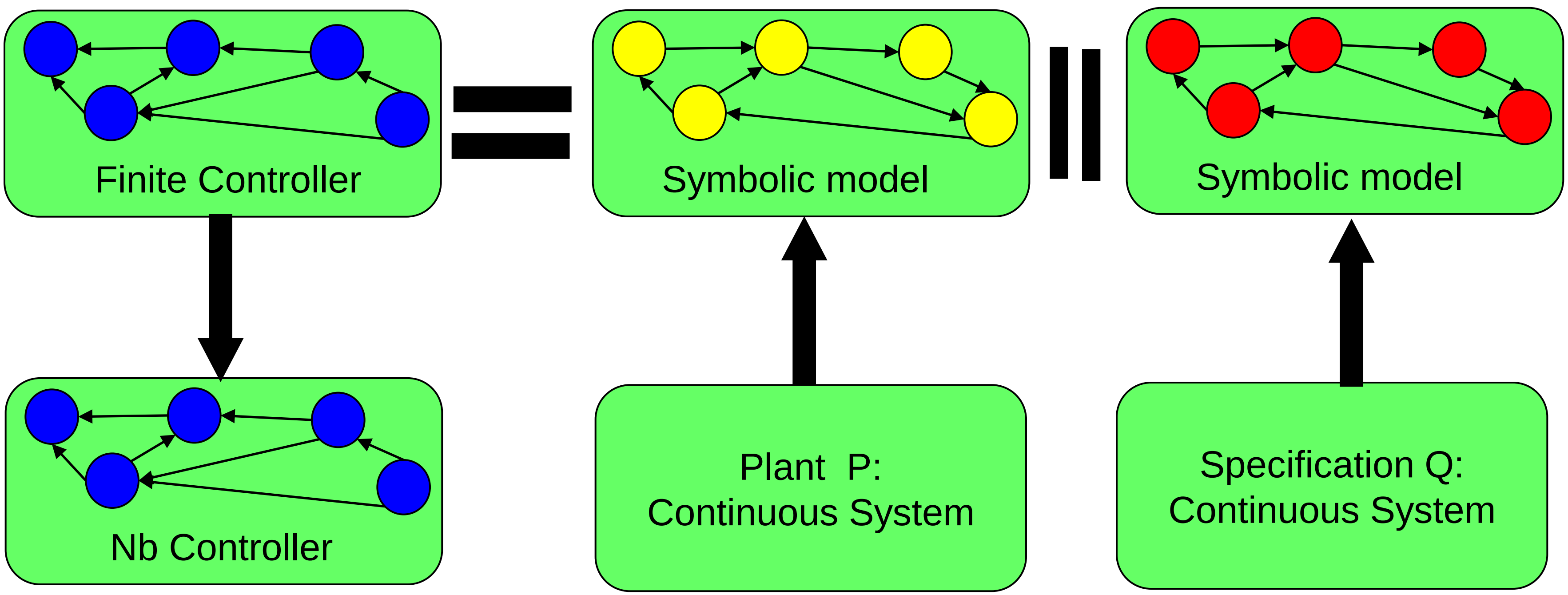
1. Compute the symbolic model $T_{\tau,\eta,\mu}(P)$ of P
2. Compute the symbolic model $T_{\tau,\eta,0}(Q)$ of Q
3. Compute the symbolic controller $C^* = T_{\tau,\eta,\mu}(P) || T_{\tau,\eta,0}(Q)$
4. Compute the non-blocking part $Nb(C^*)$ of C^*



Solution of Problem 1

Synthesis through a four-step process:

1. Compute the symbolic model $T_{\tau,\eta,\mu}(P)$ of P
2. Compute the symbolic model $T_{\tau,\eta,0}(Q)$ of Q
3. Compute the symbolic controller $C^* = T_{\tau,\eta,\mu}(P) || T_{\tau,\eta,0}(Q)$
4. Compute the non-blocking part $Nb(C^*)$ of C^*



Solution of Problem 1

Synthesis through a four-step process:

1. Compute the symbolic model $T_{\tau,\eta,\mu}(P)$ of P
2. Compute the symbolic model $T_{\tau,\eta,0}(Q)$ of Q
3. Compute the symbolic controller $C^* = T_{\tau,\eta,\mu}(P) || T_{\tau,\eta,0}(Q)$
4. Compute the non-blocking part $Nb(C^*)$ of C^*

Theorem Suppose that P and Q are δ–ISS and choose parameters ε_p, $\varepsilon_q > 0$ so that

(1) $$\varepsilon_p + \varepsilon_q \leq \varepsilon$$

Choose parameters τ, η, $\mu > 0$ satisfying

(2) $$\beta_p(\varepsilon_p,\tau) + \eta + \gamma_p(\mu) \leq \varepsilon_p$$

(3) $$\beta_q(\varepsilon_q,\tau) + \eta \leq \varepsilon_q$$

The symbolic controller $Nb(C^*)$ solves Problem 1 with $\theta = \varepsilon_p$

Design of symbolic controllers

Drawbacks

- It considers the whole sets of states of $T_{\tau,\eta,\mu}(P)$ and $T_{\tau,\eta,0}(Q)$
- 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 $T_{\tau,\eta,0}(Q)$, then C^*, to finally eliminate blocking states from C^*

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

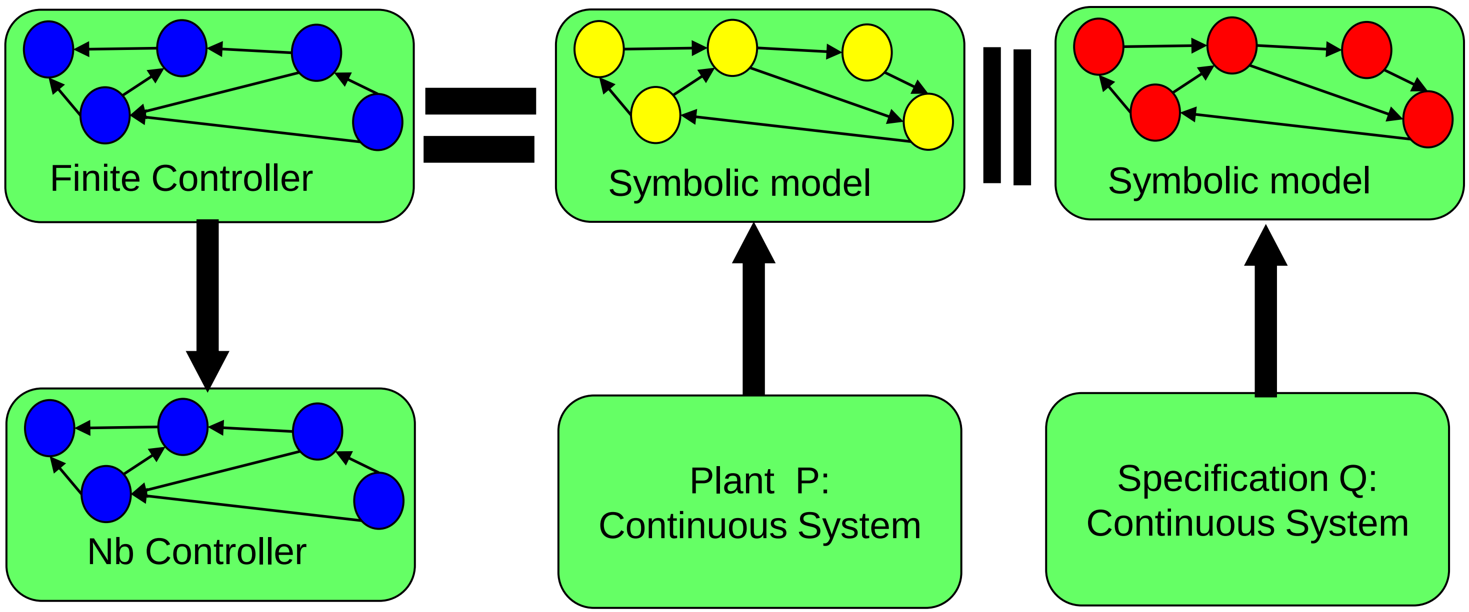
(1) Discrete abstraction $T_{\tau,\eta,\mu}(P)$ of P
(2) Discrete abstraction $T_{\tau,\eta,0}(Q)$ of Q
(3) Symbolic controller $C^* = T_{\tau,\eta,\mu}(P) || T_{\tau,\eta,0}(Q)$
(4) Non-blocking part $Nb(C)$ of C^*

Integrated Approach: Compute (1) + (2) + (3) + (4) at once!

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

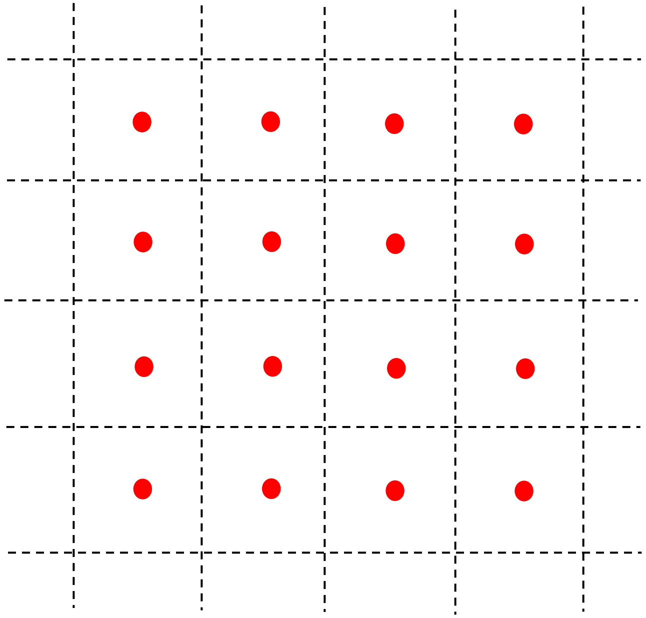
Basic ideas

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 show up



How does it work?

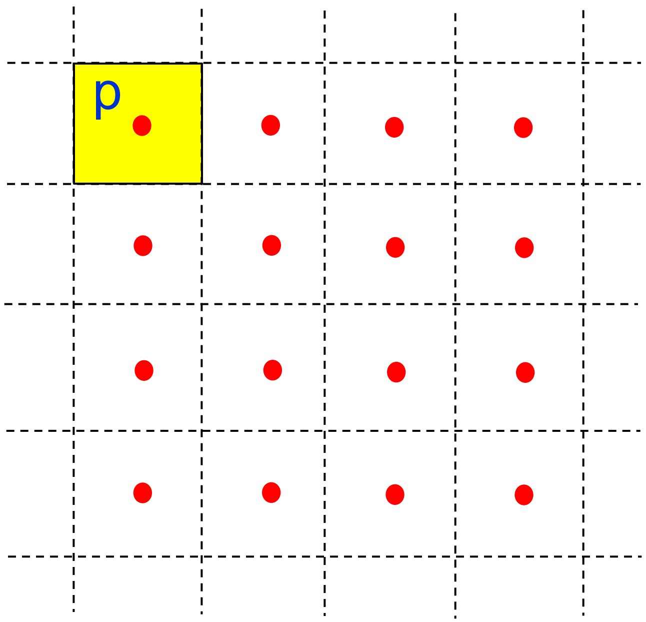
First, we consider the target space as the intersection of the sets of initial states of $T_{\tau,\eta,\mu}(P)$ and $T_{\tau,\eta,0}(Q)$.



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 $T_{\tau,\eta,0}(Q)$.

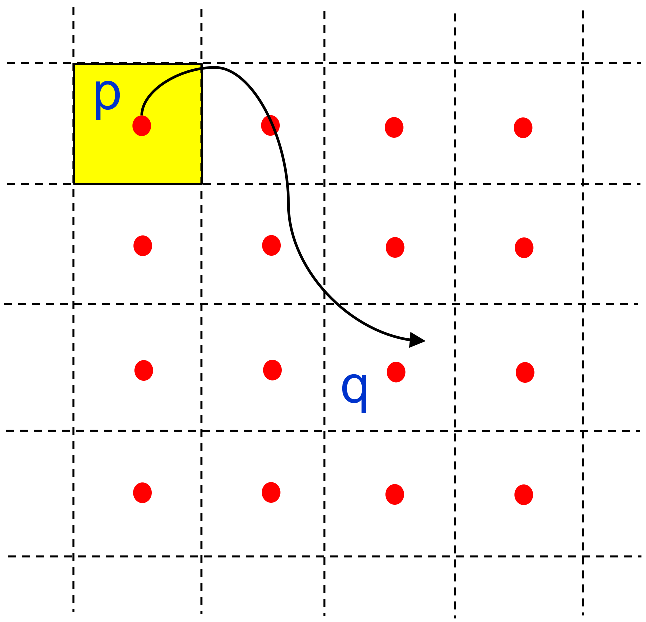
Pick a "symbolic" state p from the target space and compute $[x_q(\tau,p)]_{2\eta}=q$ by integrating the specification differential equation



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 $T_{\tau,\eta,0}(Q)$.

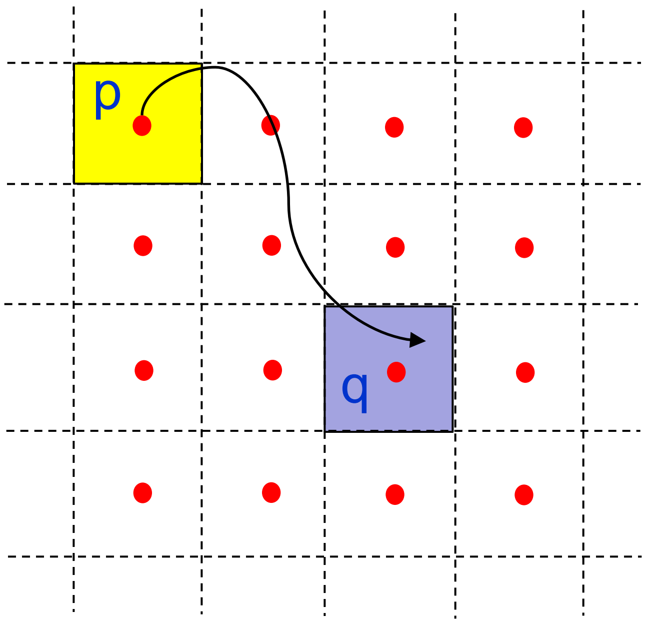
Pick a "symbolic" state p from the target space and compute $[x_q(\tau,p)]_{2\eta}=q$ by integrating the specification differential equation



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 $T_{\tau,\eta,0}(Q)$.

Pick a "symbolic" state p from the target space and compute $[x_q(\tau,p)]_{2\eta}=q$ by integrating the specification differential equation

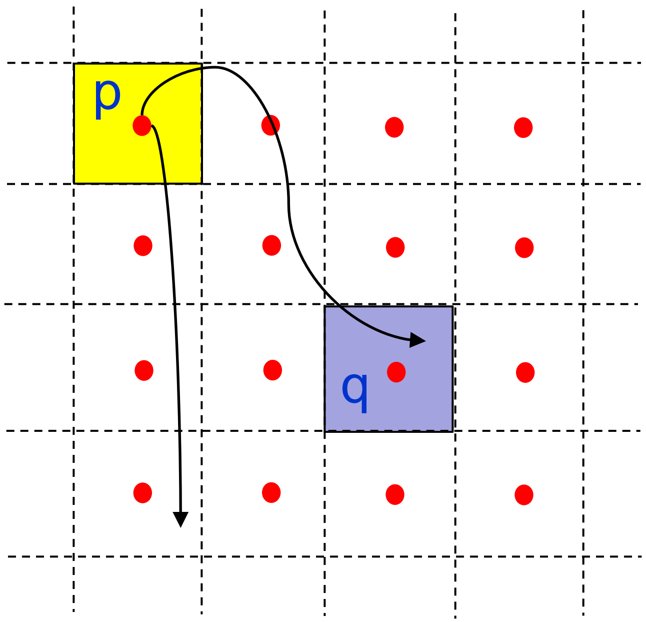


How does it work?

First, we consider the target space as the intersection of the sets of initial states of $T_{\tau,\eta,\mu}(P)$ and $T_{\tau,\eta,0}(Q)$.

Pick a "symbolic" state p from the target space and compute $[x_q(\tau,p)]_{2\eta}=q$ by integrating the specification differential equation

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

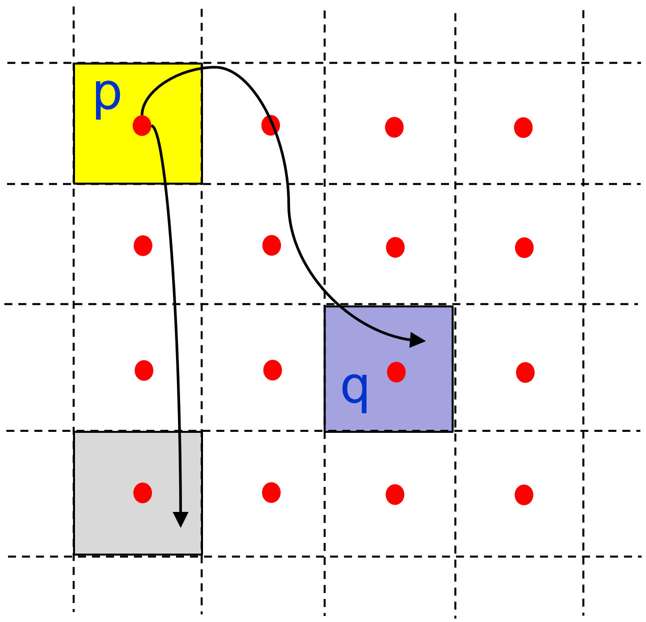


How does it work?

First, we consider the target space as the intersection of the sets of initial states of $T_{\tau,\eta,\mu}(P)$ and $T_{\tau,\eta,0}(Q)$.

Pick a "symbolic" state p from the target space and compute $[x_q(\tau,p)]_{2\eta}=q$ by integrating the specification differential equation

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



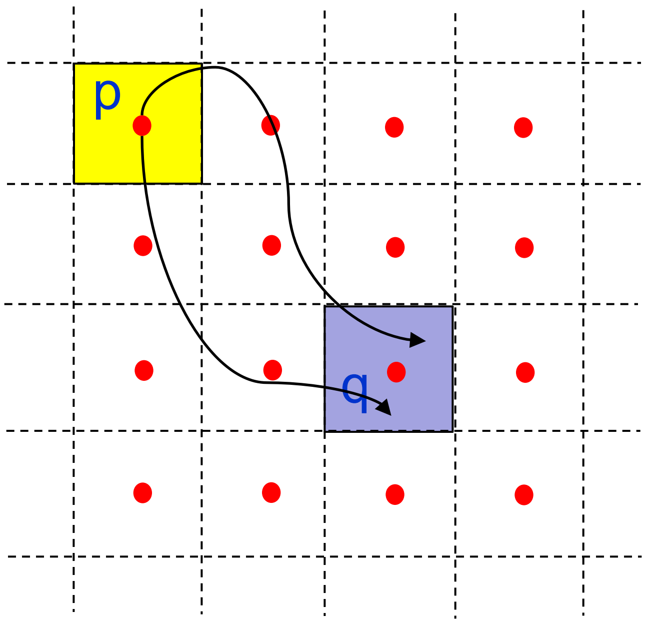
No matching! Try another input!

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 $T_{\tau,\eta,0}(Q)$.

Pick a "symbolic" state p from the target space and compute $[x_q(\tau,p)]_{2\eta}=q$ by integrating the specification differential equation

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

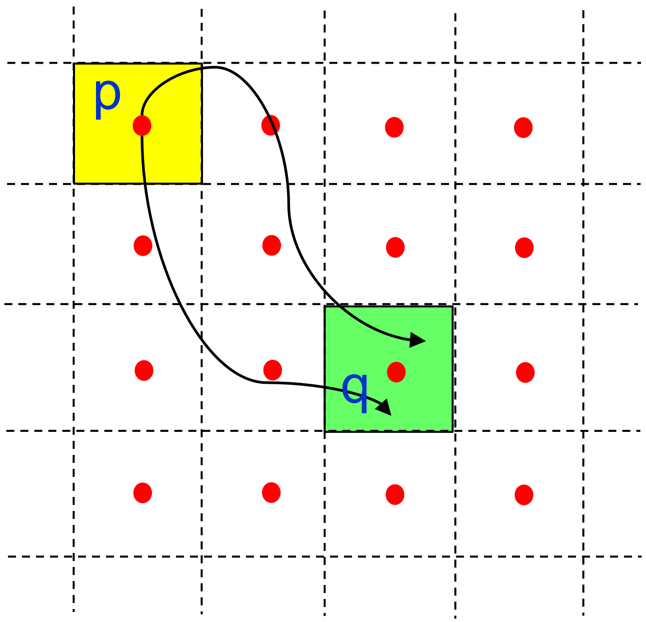


How does it work?

First, we consider the target space as the intersection of the sets of initial states of $T_{\tau,\eta,\mu}(P)$ and $T_{\tau,\eta,0}(Q)$.

Pick a "symbolic" state p from the target space and compute $[x_q(\tau,p)]_{2\eta}=q$ by integrating the specification differential equation

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



Matching found!!

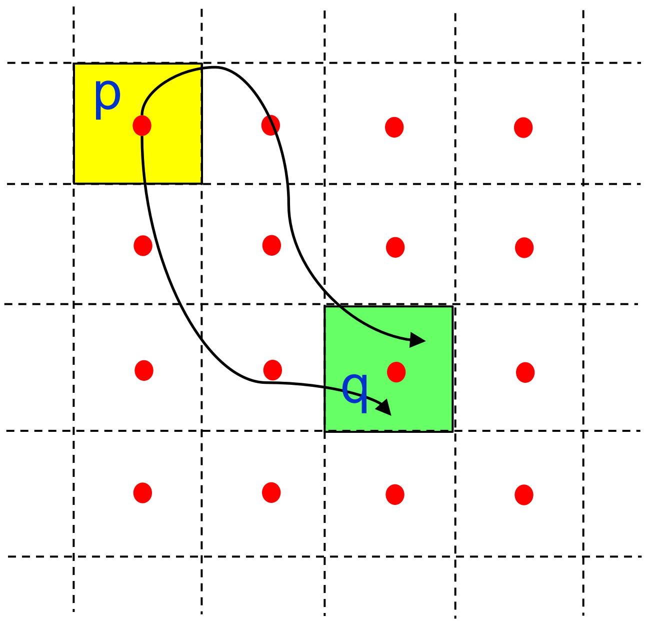
How does it work?

First, we consider the target space as the intersection of the sets of initial states of $T_{\tau,\eta,\mu}(P)$ and $T_{\tau,\eta,0}(Q)$.

Pick a "symbolic" state p from the target space and compute $[x_q(\tau,p)]_{2\eta}=q$ by integrating the specification differential equation

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

Add the transition (p,u,q) to the controller.



Matching found!!

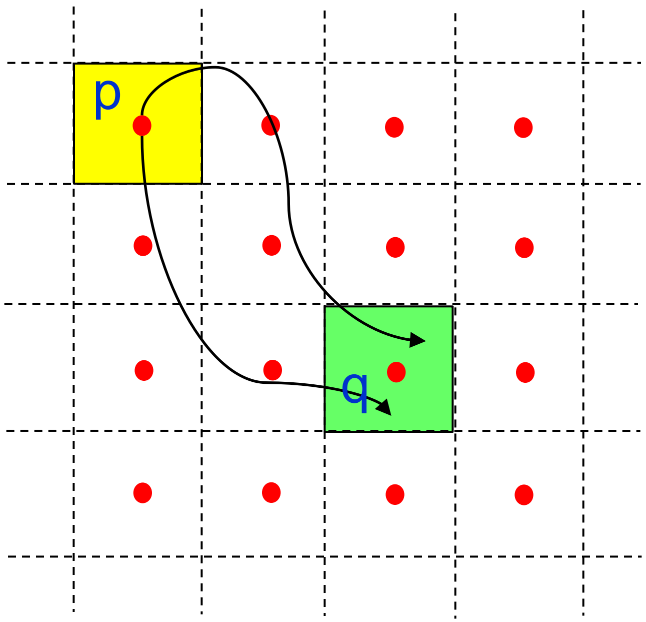
How does it work?

First, we consider the target space as the intersection of the sets of initial states of $T_{\tau,\eta,\mu}(P)$ and $T_{\tau,\eta,0}(Q)$.

Pick a "symbolic" state p from the target space and compute $[x_q(\tau,p)]_{2\eta}=q$ by integrating the specification differential equation

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

Add the transition (p,u,q) to the controller.



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 NB.

Properties

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

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