Reset Controller Synthesis

A Correct-by-Construction to the Design of CPS

Naijun Zhan

State Key Lab. of Computer Scienc
Institute of Software, Chinese Academy of Sciences
(Joint work with Han Su, Jiang Liu, Yunjun Bai, Bin Gu, Mengfei Yang and Jiyu Zhu)

Algebra and Logic Seminar in Sofia
On-line, Sept. 29, 2023

Outline

1 Motivation

2 Only with Safety

3 Safety Together with Liveness

4 Taking Delay into Account

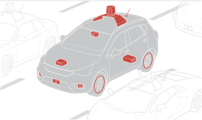
Cyber-Physical Systems (CPS)

"[...] cyber-physical systems (CPS) refers to a new generation of systems with integrated computational and physical capabilities that can interact with humans through many new modalities. The ability to interact with, and expand the capabilities of, the physical world through computation, communication, and control is a key enabler for future technology developments."

[Radhakisan Baheti and Helen Gill: CPS. The Impact of Control Technology, 2011]

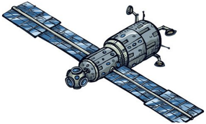
Cyber-Physical Systems (CPS)

An open, interconnected form of embedded systems; many are safety-critical.



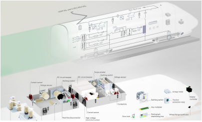
©TheOneBrief

automobiles



©clipartlogo

spacecrafts

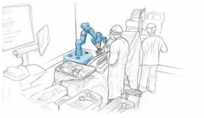


©Sécheron

high-speed rail

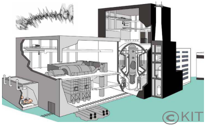
©VectorStock

nuclear reactors



©Scoop.it

robot surgeon

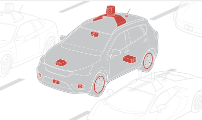


©KIT

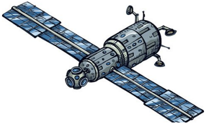
robust control

Cyber-Physical Systems (CPS)

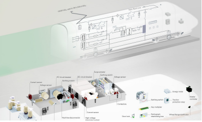
An open, interconnected form of embedded systems; many are safety-critical.



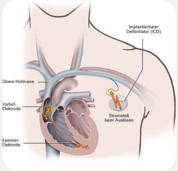
©TheOneBrief



©clipartlogo



©Sécheron



212 patients died of defibrillator failure

(USA, 1997 – 2003)

40 passengers died plus 172 injured

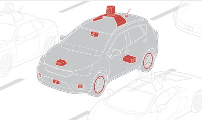
(China, 2011.7.23)

31 billion Yen loss on ASTRO-H

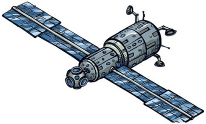
(Japan, 2016.3.26)

Cyber-Physical Systems (CPS)

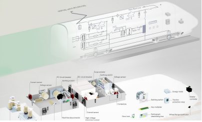
An open, interconnected form of embedded systems; many are safety-critical.



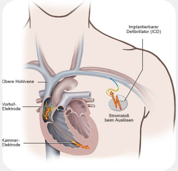
©TheOneBrief



©clipartlogo



©Sécheron



212 patients died of defibrillator failure

(USA, 1997 – 2003)

40 passengers died plus 172 injured

(China, 2011.7.23)

31 billion Yen loss on ASTRO-H

(Japan, 2016.3.26)

"How can we provide people with CPS they can bet their lives on?"

— Jeannette M. Wing, former AD for CISE at NSF

Controller Synthesis

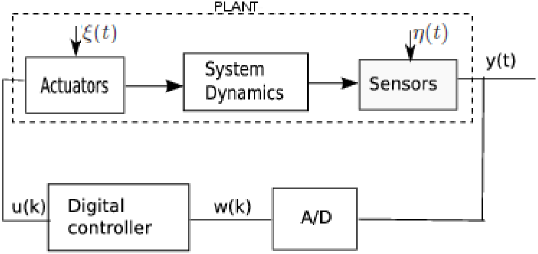
Controller Synthesis [from Wikipedia]

Given a model of the assumed behaviours of the environment and a system goal, controller synthesis means to construct an operational behaviour model for a component s.t. the system is guaranteed to satisfy the given goal when the environment is consistent with the given assumptions.

• An operation could be either inputs to dynamics, switching conditions, initial conditions, or reset functions.

Feedback controller

- Changing inputs impulsed on the dynamics (continuous or discrete)
- Steering the system to satisfy **stability**,**safety**, etc.



Controller Synthesis

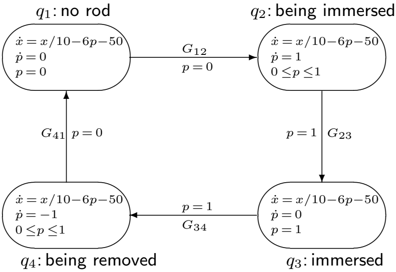
Controller Synthesis [from Wikipedia]

Given a model of the assumed behaviours of the environment and a system goal, controller synthesis means to construct an operational behaviour model for a component s.t. the system is guaranteed to satisfy the given goal when the environment is consistent with the given assumptions.

• An operation could be either inputs to dynamics, switching conditions, initial conditions, or reset functions.

Switching logic controller

- Refining the guard associated with each jump and the domain constraint in each mode
- Restricting the behavior so that the refined system satisfies the system objective



Objective: $510 \leq x \leq 550$

Controller Synthesis

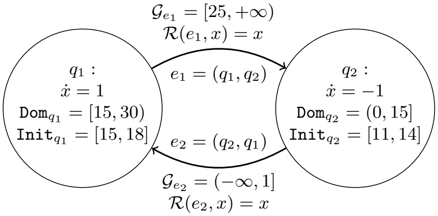
Controller Synthesis [from Wikipedia]

Given a model of the assumed behaviours of the environment and a system goal, controller synthesis means to construct an operational behaviour model for a component s.t. the system is guaranteed to satisfy the given goal when the environment is consistent with the given assumptions.

• An operation could be either inputs to dynamics, switching conditions, initial conditions, or reset functions.

Reset controller

- Redefining the reset map associated with each jump and refining the initial set of each mode
- Steering the modified system to achieve the system objective like **safety**, **stability**, **liveness**, etc.



Objective: $\mathcal{S}_1 = [15, 31)$, $\mathcal{S}_2 = (0, 14]$

Controller Synthesis

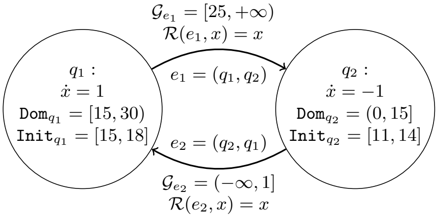
Controller Synthesis [from Wikipedia]

Given a model of the assumed behaviours of the environment and a system goal, controller synthesis means to construct an operational behaviour model for a component s.t. the system is guaranteed to satisfy the given goal when the environment is consistent with the given assumptions.

• An operation could be either inputs to dynamics, switching conditions, initial conditions, or reset functions.

Why a reset controller necessary?

- It is impossible to have feedback controllers for them to maintain safety
- Moreover, the system state will leave the safe set once a discrete jump happens

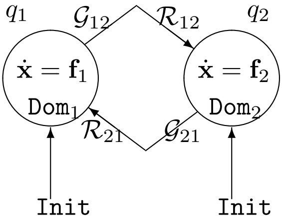


Objective: $\mathcal{S}_1 = [15, 31)$, $\mathcal{S}_2 = (0, 14]$

Hybrid Automaton

$\mathcal{H} \widehat{=} (\mathcal{Q}, X, \mathbf{f}, \mathtt{Init}, \mathtt{Dom}, \mathcal{E}, \mathcal{G}, \mathcal{R})$ [Tomlin et al 00] ,
where

- $\mathcal{Q} = \{q_1, \ldots, q_m\}$: discrete states, or modes
- $X = \{x_1, \ldots, x_n\}$: continuous state variables, $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$
- $\mathbf{f} : \mathcal{Q} \to (\mathbb{R}^n \to \mathbb{R}^n)$: continuous dynamics, $\mathbf{f}_q : \mathbb{R}^n \to \mathbb{R}^n$
- $\mathtt{Init} \subseteq \mathcal{Q} \times \mathbb{R}^n$: initial states
- $\mathtt{Dom} : \mathcal{Q} \to 2^{\mathbb{R}^n}$: domains $\mathtt{Dom}_q \subseteq \mathbb{R}^n$
- $\mathcal{E} \subseteq \mathcal{Q} \times \mathcal{Q}$: discrete transitions
- $\mathcal{G} : \mathcal{E} \to 2^{\mathbb{R}^n}$: switching guards $\mathcal{G}_e \subseteq \mathbb{R}^n$
- $\mathcal{R} : \mathcal{E} \to (\mathbb{R}^n \to \mathbb{R}^n)$: reset functions $\mathcal{R}(e, \cdot)$: $\mathbb{R}^n \to \mathbb{R}^n$



Problem Formulation

Given an HA $\mathcal{H}$, we are interested in the following two types of reset controller synthesis problems:

Problem I: only with safety

Given a safe set $\mathcal{S} \subseteq \mathcal{Q} \times \mathcal{X}$, whether one can redefine $\mathtt{Init}$ and $\mathcal{R}$, and obtain a redesigned HA $\mathcal{H}' = (\mathcal{Q}, X, \mathbf{f}, \mathtt{Init}^r, \mathtt{Dom}, \mathcal{E}, \mathcal{G}, \mathcal{R}^r)$, which is safe w.r.t. $\mathcal{S}$, and $\mathtt{Init}^r \subseteq \mathtt{Init}$;

Problem II: safety+liveness

Given a safe set $\mathcal{S} \subseteq \mathcal{Q} \times \mathcal{X}$ and a target set $\mathcal{T} \subseteq \mathcal{Q} \times \mathcal{X}$, whether one can redefine $\mathtt{Init}$ and $\mathcal{R}$, and obtain a redesigned HA $\mathcal{H}' = (\mathcal{Q}, X, \mathbf{f}, \mathtt{Init}^r, \mathtt{Dom}, \mathcal{E}, \mathcal{G}, \mathcal{R}^r)$, s.t. for any $(q, \boldsymbol{x}) \in \mathtt{Init}^r$, any trajectory starting from $(q, \boldsymbol{x})$ must reach $\mathcal{T}$, $\mathcal{H}'$ is safe w.r.t. $\mathcal{S}$ before reaching into $\mathcal{T}$, and $\mathtt{Init}^r \subseteq \mathtt{Init}$.

Outline

1 Motivation

2 Only with Safety

3 Safety Together with Liveness

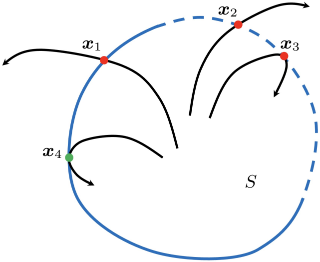
4 Taking Delay into Account

Transverse Set

Given a vector field $\mathbf{f}$ and a set $S \subseteq \mathbb{R}^n$, **the transverse set** of S w.r.t. $\mathbf{f}$, denoted by $\mathtt{trans}_{\mathbf{f}\uparrow S}$ of $\mathbf{f}$ over S, is defined by

$$\mathtt{trans}_{\mathbf{f}\uparrow S} = \left\{ \boldsymbol{x} \in \partial S \,\middle|\, \begin{array}{l} \forall \epsilon > 0\ \exists t \in [0, \epsilon). \\ \phi(\boldsymbol{x}, t) \notin S \end{array} \right\}$$

- $\boldsymbol{x}_1 \in \mathtt{trans}_{\mathbf{f}\uparrow S}$
- $\boldsymbol{x}_2 \in \mathtt{trans}_{\mathbf{f}\uparrow S}$
- $\boldsymbol{x}_3 \in \mathtt{trans}_{\mathbf{f}\uparrow S}$
- $\boldsymbol{x}_4 \notin \mathtt{trans}_{\mathbf{f}\uparrow S}$



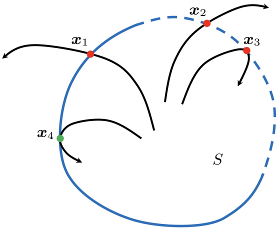
Differential Invariant (DI)

A set C is a **differential invariant** of vector field $\mathbf{f}$ w.r.t. a set S if for all $\boldsymbol{x} \in C$ and $T \geq 0$

$$\begin{pmatrix} \forall t \in [0,T]. \\ \phi(\boldsymbol{x}, t) \in S \end{pmatrix} \Longrightarrow \begin{pmatrix} \forall t \in [0,T]. \\ \phi(\boldsymbol{x}, t) \in C \end{pmatrix}$$

- In other words, $\mathtt{trans}_{\mathbf{f}\uparrow S\cap C} = \emptyset$.
- Any set C, if $S \subseteq C$, then C is a DI. So, we only consider DI contained in S afterwards.

- $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ do not belong to any DI w.r.t. S
- $\boldsymbol{x}_4$ belong to a DI w.r.t. S



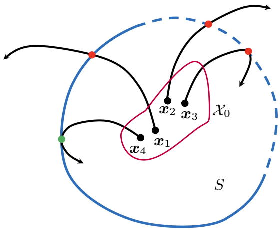
Reach-Avoid Set

Reach-Avoid Set

Given a vector field $\mathbf{f}$, an initial set $\mathcal{X}_0$, a safe set S and a target set $\mathcal{T}$, the generalized (maximal) reach-avoid set $\mathtt{RA}(\mathcal{X}_0 \xrightarrow[\mathbf{f}]{S} \mathcal{T})$ is defined

$$\mathtt{RA}(\mathcal{X}_0 \xrightarrow[\mathbf{f}]{S} \mathcal{T}) \widehat{=} \left\{ \boldsymbol{x} \in \mathcal{X}_0 \cap S \,\middle|\, \begin{array}{l} \exists T \geq 0. \forall t \in [0, T). \phi(\boldsymbol{x}, t) \in S \wedge \\ \forall \epsilon > 0. \exists t \in [T, T + \epsilon). \phi(\boldsymbol{x}, t) \in \mathcal{T} \end{array} \right\}.$$

- $\boldsymbol{x}_1 \in \mathtt{RA}(\mathcal{X}_0 \xrightarrow[f]{S} \mathtt{trans}_{\mathbf{f}\uparrow S})$
- $\boldsymbol{x}_2 \in \mathtt{RA}(\mathcal{X}_0 \xrightarrow[f]{S} \mathtt{trans}_{\mathbf{f}\uparrow S})$
- $\boldsymbol{x}_3 \in \mathtt{RA}(\mathcal{X}_0 \xrightarrow[f]{S} \mathtt{trans}_{\mathbf{f}\uparrow S})$
- $\boldsymbol{x}_4 \notin \mathtt{RA}(\mathcal{X}_0 \xrightarrow[f]{S} \mathtt{trans}_{\mathbf{f}\uparrow S})$



Computing TS, DI and GRA by SDP

Theorem

For a semialgebraic set S, let $C \widehat{=} S \setminus \mathtt{RA}(S \xrightarrow[\mathbf{f}]{S} \mathtt{trans}_{\mathbf{f}\uparrow S})$, then C is a semialgebraic DI of $\mathbf{f}$ w.r.t. S, if $\mathbf{f}$ is polynomial.

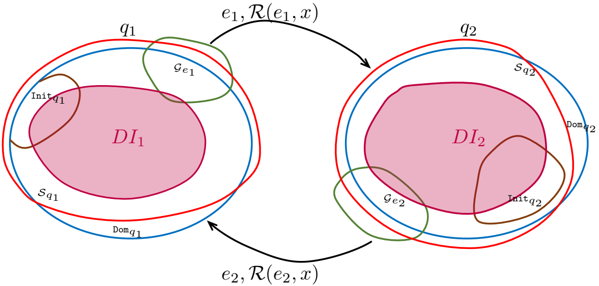
Theorem

Let S and D be a semialgebraic set, and $\mathbf{f}$ be polynomial, then $\mathtt{trans}_{f\uparrow S}$, $\mathtt{RA}(S \xrightarrow[f]{S} D)$, and DI defined by $S \setminus \mathtt{RA}(S \xrightarrow[f]{S} D)$ can be computed efficiently by SDP.

Reset synthesis only with safety

Basic idea

- **Stay within each mode forever**



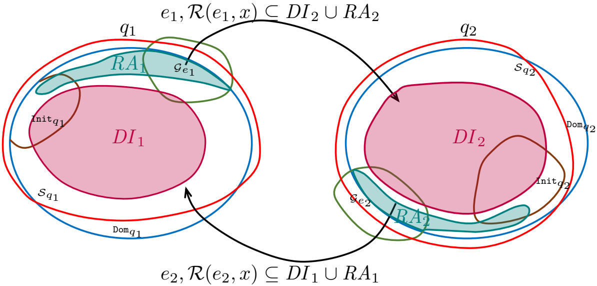
$$DI_1 = \mathtt{SD}_{q_1} \setminus \mathtt{RA}(\mathtt{SD}_{q_1} \xrightarrow[\mathbf{f}_{q_1}]{\mathtt{SD}_{q_1}} \mathtt{trans}_{\mathbf{f}_{q_1} \uparrow \mathtt{SD}_{q_1}})$$

$$DI_2 = \mathtt{SD}_{q_2} \setminus \mathtt{RA}(\mathtt{SD}_{q_2} \xrightarrow[\mathbf{f}_{q_2}]{\mathtt{SD}_{q_2}} \mathtt{trans}_{\mathbf{f}_{q_2} \uparrow \mathtt{SD}_{q_2}})$$

Reset synthesis only with safety

Basic idea

- **Reset to the switching part of the post-mode, but still inside a global invariant of the whole system.**



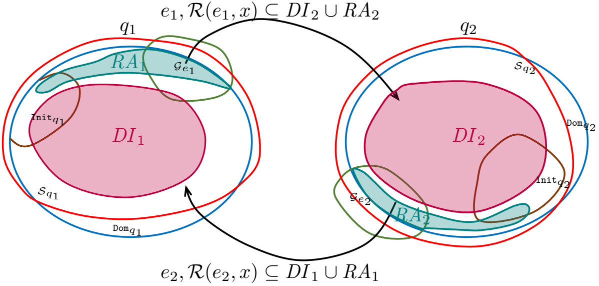
$$\mathtt{RA}_1 = \mathtt{RA}(\mathtt{SD}_{q_1} \xrightarrow[\mathbf{f}_{q_1}]{\mathtt{SD}_{q_1}} \mathtt{trans}_{\mathbf{f}_{q_1} \uparrow \mathtt{SD}^c_{q_1} \cap \mathcal{G}_{e_1}})$$

$$\mathtt{RA}_2 = \mathtt{RA}(\mathtt{SD}_{q_2} \xrightarrow[\mathbf{f}_{q_2}]{\mathtt{SD}_{q_2}} \mathtt{trans}_{\mathbf{f}_{q_2} \uparrow \mathtt{SD}^c_{q_2} \cap \mathcal{G}_{e_2}})$$

Reset synthesis only with safety

Basic idea

- **Finally, synthesize a reset controller.**



$$\mathtt{Init}^r_{q_1} = \mathtt{Init}_{q_1} \cap (\mathtt{DI}_1 \cup \mathtt{RA}_1)$$

$$\mathtt{Init}^r_{q_2} = \mathtt{Init}_{q_2} \cap (\mathtt{DI}_2 \cup \mathtt{RA}_2)$$

$$\mathcal{R}^r(e_1, x) \subseteq \mathtt{DI}_1 \cup \mathtt{RA}_1 \ \forall x \in \mathcal{G}_{e_1}$$

$$\mathcal{R}^r(e_2, x) \subseteq \mathtt{DI}_2 \cup \mathtt{RA}_2 \ \forall x \in \mathcal{G}_{e_2}$$

Reset synthesis only with safety

Algorithm

Algorithm Reset Control Synthesis Only with Safety

Require: $\mathcal{H} = (\mathcal{Q}, \mathcal{X}, \mathbf{f}, \mathtt{Init}, \mathtt{Dom}, \mathcal{E}, \mathcal{G}, \mathcal{R})$ and safe set $\mathcal{S}$
Ensure: $\mathcal{H}^r = (\mathcal{Q}, \mathcal{X}, \mathbf{f}, \mathtt{Init}^r, \mathtt{Dom}, \mathcal{E}, \mathcal{G}, \mathcal{R}^r)$ satisfying $\mathcal{S}$

for each $q \in \mathcal{Q}$ **do**
  $\mathtt{SD}_q \leftarrow \mathcal{S}_q \cap \mathtt{Dom}_q$;
  $\mathtt{Dom}_q^r \leftarrow \mathtt{SD}_q \setminus \mathtt{RA}(\mathtt{SD}_q \xrightarrow[\mathbf{f}_q]{\mathtt{SD}_q} \mathtt{trans}_{\mathbf{f}_q \uparrow \mathtt{SD}_q})$;
  for each $p \in \mathtt{Post}(q)$ **do**
    $\mathtt{Dom}_q^r \leftarrow \mathtt{Dom}_q^r \cup \mathtt{RA}(\mathtt{SD}_q \xrightarrow[\mathbf{f}_q]{\mathtt{SD}_q} \mathtt{Dom}_q^c \cap \mathcal{G}_e))$;
  end for
  for each $p \in \mathtt{Pre}(q)$ **do**
    set $\mathcal{R}^r(e = (p, q), x) \subset \mathtt{Dom}_q^r$, for $x \in \mathcal{G}_e$;
  end for
  $\mathtt{Init}_q^r \leftarrow \mathtt{Init}_q \cap \mathtt{Dom}_q^r$;
end for
return $\mathcal{H}^r = (\mathcal{Q}, \mathcal{X}, \mathbf{f}, \mathtt{Init}^r, \mathtt{Dom}, \mathcal{E}, \mathcal{G}, \mathcal{R}^r)$;

Reset synthesis only with safety

Correctness

Correctness

Problem I is solvable if and only if $\mathtt{Init}^r$ obtain from the above algorithm is not empty.

- **Soundness:** If $\mathtt{Init}^r$ obtained from the above algorithm is not empty, the resulting $\mathcal{H}^r$ solves **Problem I**.
- **Completeness:** If **Problem I** can be solved by some reset controller, $\mathtt{Init}^r$ obtained from the above algorithm is not empty.

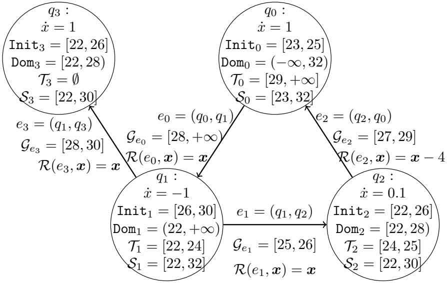
Outline

1. Motivation
2. Only with Safety
3. Safety Together with Liveness
4. Taking Delay into Account

Safety together with Liveness

Basic idea

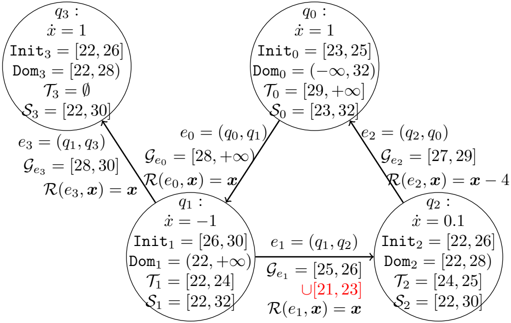
- block all trajectories that can reach to q_3, as $\mathcal{T}_3 = \emptyset$, which implies the liveness cannot be satisfied along these trajectories;
- meanwhile, also need to block all trajectories with a simple loop containing q_0, q_1 and q_2, as such trajectories could evole infinitely along the loop, and never reach to the target.;
- It can be done by blocking a selected discrete transition on the simple loop by redefining the reset maps associated with all incoming edges to and the initial set of the pre-mode of the transition.



Safety together with Liveness

Basic idea

- There may not exist a reset controller.



Safety together with Liveness

Algorithm

Require: $\mathcal{H} = (\mathcal{Q}, \mathcal{X}, \mathbf{f}, \mathtt{Init}, \mathtt{Dom}, \mathcal{E}, \mathcal{G}, \mathcal{R})$, safe set $\mathcal{S}$ and target set $\mathcal{T}$
Ensure: $\mathcal{H}^r = (\mathcal{Q}, \mathcal{X}, \mathbf{f}, \mathtt{Init}^r, \mathtt{Dom}, \mathcal{E}, \mathcal{G}, \mathcal{R}^r)$ that can guarantee that all trajectories can reach to $\mathcal{T}$ and satisfy $\mathcal{S}$ before reaching $\mathcal{T}$, or "No Such Reset Controllers Exist"

for each $q \in \mathcal{Q}$ **do**
  $\mathtt{SD}_q \leftarrow \mathcal{S}_q \cap \mathtt{Dom}_q$;
  $\mathtt{Dom}^r_q \leftarrow \mathtt{RA}(\mathtt{SD}_q \xrightarrow[\mathbf{f}_q]{\mathtt{SD}_q} \mathcal{T}_q)$;
  for each $p \in \mathtt{Post}(q)$ **do**
  $\mathtt{Dom}^r_q \leftarrow \mathtt{Dom}^r_q \cup \mathtt{RA}(\mathtt{SD}_q \xrightarrow[\mathbf{f}_q]{\mathtt{SD}_q} \mathtt{Dom}^c_q \cap \mathcal{G}_{e=(q,p)}))$ **end for**;
  $\mathtt{Init}^r_q \leftarrow \mathtt{Init}_q \cap \mathtt{Dom}^r_q$; $\mathtt{ST}_q \leftarrow \mathtt{ST}_q$ computed by (1);
end for
for each q with $\mathtt{Init}^r_q \neq \emptyset$ **do** *Refining_Dom*(q) **end for** ;
for each $q \in \mathcal{Q}$ **do** $\mathtt{Init}^r_q \leftarrow \mathtt{Init}^r_q \cap \mathtt{Dom}^r_q$ **end for**;
for each $e = (p, q) \in \mathcal{E}$ **do**
  $\mathcal{R}^r(e, x) \subseteq \mathtt{Dom}^r_q$ **if** $\mathcal{R}^r(e, x)$ is not redefined in Algorithm 3;
end for
if $\mathtt{Init}^r = \cup_{q\in Q}\mathtt{Init}^r_q \neq \emptyset$ **then**
  return $\mathcal{H}^r = (\mathcal{Q}, \mathcal{X}, \mathbf{f}, \mathtt{Init}^r, \mathtt{Dom}, \mathcal{E}, \mathcal{G}, \mathcal{R}^r)$;
else
  return "No Such Reset Controllers Exist";
end if

Safety together with Liveness

Correctness

Theorem [Correctness]

Problem II is solvable if and only if $\mathtt{Init}^r$ obtain from the above algorithm is not empty.

- **Soundness:** Our approach is sound, that is, any reset controller synthesized by the above approach does solve **Problem II**;
- **Completeness:** Our approach is also complete, that is, if **Problem II** can be solved by some reset controller, the above approach does synthesize such one.

Outline

1 Motivation

2 Only with Safety

3 Safety Together with Liveness

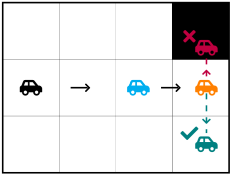
4 Taking Delay into Account

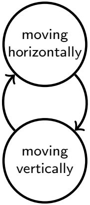
Delay Is Inevitable in the Design of CPS

Delay is inevitable in the design of CPS, because of

- conversions between analog and digital signal domains
- complex digital signal-processing chains enhancing
- filtering and fusing sensory signals before they enter control
- sensor networks harvesting multiple sensor sources before feeding them to control
- network delays in networked control applications physically removing the controller(s) from the control path, and just name a few

The delay-free assumption makes the problem mathematically simple, but physically impossible, even impractical, as it may lead to deteriorated control performance and invalid verification certificates obtained by abstracting away time-delay in practice.





Reach-avoid for DDE

Consider a DDE of the form

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{x}(t-\tau)), \quad \mathbf{f} \in \mathbb{R}\left[\mathbf{x}(t), \mathbf{x}(t-\tau)\right]^n \tag{1}$$

Reach-avoid set for DDE

Given a vector field $\mathbf{f} : \mathcal{C}([-\tau, 0], \mathbb{R}^n) \to \mathbb{R}^n$, a safe set $\mathcal{S} \in \mathbb{R}^n$ and a target set $\mathcal{T} \in \mathbb{R}^n$, a (the maximal) reach-avoid set $\mathcal{RA}(\mathbf{f}, \mathcal{S}, \mathcal{T})$ is defined as

$$\mathcal{RA} \widehat{=} \left\{ \phi \in \mathcal{C}([-\tau, 0], \mathcal{S}) \mid \exists\, t' \in \mathbb{R},\ x^{\phi}(t') \in T \wedge \forall\, t \in [-\tau, t'),\ x^{\phi}(t) \in \mathcal{S} \ \right\}$$

Reach-Avoid Barrier Functional

Given a DDE of the form (1) with domain $D \subseteq \mathbb{R}^n$, safe set $\mathcal{S}$ and target set $\mathcal{T}$ defined by

$$\mathcal{S} \widehat{=} \{\mathbf{x} \in D \mid s(\mathbf{x}) \leq 0\}, \mathcal{T} \widehat{=} \{\mathbf{x} \in D \mid g(\mathbf{x}) \leq 0\}.$$

We call $H : \mathcal{C}([-\tau, 0], D) \to \mathbb{R}$ a *reach-avoid barrier functional* if we can find a bounded function $w : D \to \mathbb{R}$ such that the following conditions are satisfied:

$$-\frac{dH(\mathbf{x}_t)}{dt} \geq 0,\ \forall \mathbf{x}_t \in \mathcal{C}([-\tau, 0], \mathcal{S}) \tag{2}$$

$$H(\mathbf{x}_t) \geq 0,\ \forall \mathbf{x}_t \in \mathcal{C}([-\tau, 0], \mathcal{S}),\ s.t.\ \mathbf{x}_t(0) \in \partial\mathcal{S} \tag{3}$$

$$H(\mathbf{x}_t) - \frac{dw(\mathbf{x}_t(0))}{dt} \geq g(\mathbf{x}_t(0)),\ \forall \mathbf{x}_t \in \mathcal{C}([-\tau, 0], \mathcal{S}) \tag{4}$$

Inner-approximation

The set $\mathcal{RA}_{in}$ defined by the 0-sublevel set of H, i.e.,

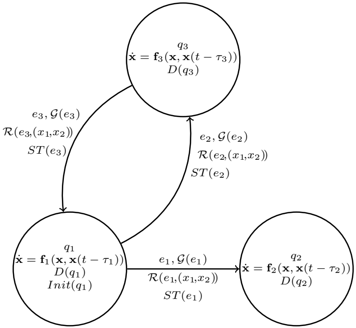
$$\mathcal{RA}_{in} \widehat{=} \{\phi \in \mathcal{C}([-\tau, 0], \mathcal{S}) \mid H(\phi) < 0\} \tag{5}$$

is an inner-approximation of $\mathcal{RA}$.

Delay Hybrid Automata (dHA)

$\mathcal{H} = (Q, X, \mathbf{f}, \mathit{Dom}, E, \mathcal{G}, \mathcal{R}, \mathtt{Init}, ST)$ [Bai et al 2021] , where

- $Q = \{q_1, \ldots, q_m\}$: discrete states, or modes
- $X = \{x_1, \ldots, x_n\}$: continuous state variables, $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$
- $\mathbf{f} : Q \to (\mathcal{C}([-\tau, 0], \mathbb{R}^n) \to \mathbb{R}^n)$: continuous dynamics with delay, $\mathbf{f}_q : \mathcal{C}([-\tau, 0], \mathbb{R}^n) \to \mathbb{R}^n$
- $\mathit{Dom} : Q \to 2^{\mathbb{R}^n}$: domains $\mathit{Dom}_q \subseteq \mathbb{R}^n$
- $E \subseteq Q \times Q$: discrete transitions
- $\mathcal{G} : E \to 2^{\mathbb{R}^n}$: switching guards $\mathcal{G}_e \subseteq \mathbb{R}^n$
- $\mathcal{R} : E \to (\mathbb{R}^n \to \mathcal{C}([-\tau, 0], \mathbb{R}^n))$: reset functions $\mathcal{R}_e : \mathbb{R}^n \to \mathcal{C}([-\tau, 0], \mathbb{R}^n)$
- $\mathtt{Init} \subseteq Q \times \mathcal{C}([-\tau, 0], \mathbb{R}^n)$: initial states
- $ST \subseteq E \times \mathbb{R}$: switching time



A Running Example

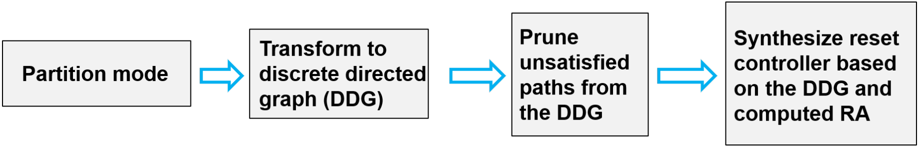
Problem Formulation

Given a dHA $\mathcal{H}$, we are interested in the following problem:

Problem III: Reset synthesis for dHA

Given a compact safe set $\mathcal{S} \subseteq Q \times X$ and target set $\mathcal{T} \subseteq Q \times X$, whether we can find a new $\mathtt{Init}^r$ and $\mathcal{R}^r$ such that all executions of the modified dHA $\mathcal{H}^r = (Q, X, \mathbf{f}, Dom, E, \mathcal{G}, \mathcal{R}^r, \mathtt{Init}^r, ST)$ will reach $\mathcal{T}$ while stay in $\mathcal{S}$ before reaching the target.

Main Idea:



Main Idea

Mode Partition

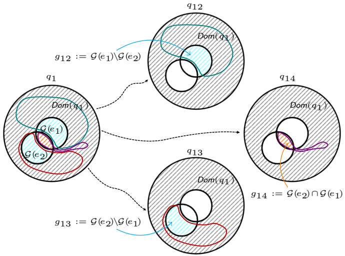
Partition mode ⇨ **Transform to discrete directed graph (DDG)** ⇨ **Prune unsatisfied paths from the DDG** ⇨ **Synthesize reset controller based on the DDG and computed RA**

Mode q_1 in the running example can be partitioned into three sub-modes

$$\mathcal{RA}_{in}(1,2) \widehat{=} \mathcal{RA}(\mathbf{f}_1, Dom(q_1) \cap \mathcal{S}_{q_1}, g_{12} \setminus Dom(q_1)),$$

$$\mathcal{RA}_{in}(1,3) \widehat{=} \mathcal{RA}(\mathbf{f}_1, Dom(q_1) \cap \mathcal{S}_{q_1}, g_{13} \setminus Dom(q_1)),$$

$$\mathcal{RA}_{in}(1,4) \widehat{=} \mathcal{RA}(\mathbf{f}_1, Dom(q_1) \cap \mathcal{S}_{q_1}, g_{14} \setminus Dom(q_1)).$$



Main Idea

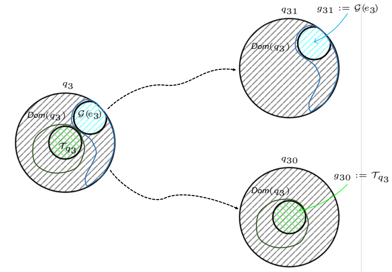
Mode Partition

Partition mode → **Transform to discrete directed graph (DDG)** → **Prune unsatisfied paths from the DDG** → **Synthesize reset controller based on the DDG and computed RA**

Mode q_2 in the running example can be partitioned into three sub-modes

$$\mathcal{R}A_{in}(3,0) \widehat{=} \mathcal{R}A(\mathbf{f}_3, \mathit{Dom}(q_3) \cap Sq_3, g_{30}),$$

$$\mathcal{R}A_{in}(3,1) \widehat{=} \mathcal{R}A(\mathbf{f}_3, \mathit{Dom}(q_3) \cap S_{q_3}, g_{31} \setminus \mathit{Dom}(q_1)).$$

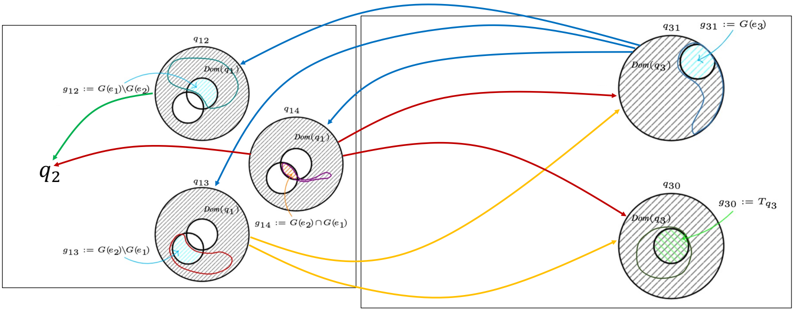


Main Idea

Mode Partition

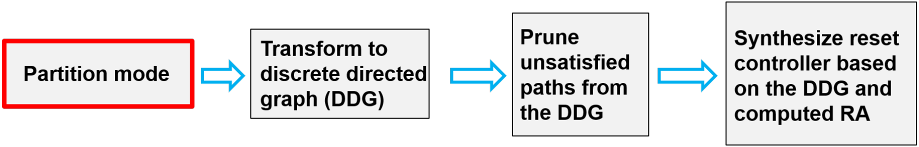
Partition mode → **Transform to discrete directed graph (DDG)** → **Prune unsatisfied paths from the DDG** → **Synthesize reset controller based on the DDG and computed RA**

Introduce edges



Main Idea

Mode Partition



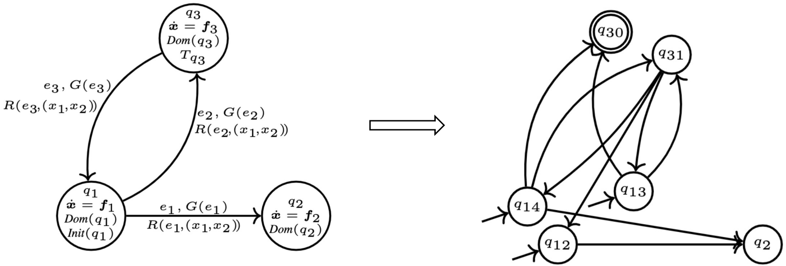
Redefine a reset map

- $\mathcal{R}^m((q_{31}, q_{12}), g_{31}) = \mathcal{RA}_{in}(1, 2)$,
- $\mathcal{R}^m((q_{31}, q_{13}), g_{31}) = \mathcal{RA}_{in}(1, 3)$,
- $\mathcal{R}^m((q_{31}, q_{14}), g_{31}) = \mathcal{RA}_{in}(1, 4)$,
- $\mathcal{R}^m((q_{13}, q_{30}), g_{13}) = \mathcal{RA}_{in}(3, 0)$,
- $\mathcal{R}^m((q_{13}, q_{31}), g_{13}) = \mathcal{RA}_{in}(3, 1)$,
- $\vdots$

Main Idea

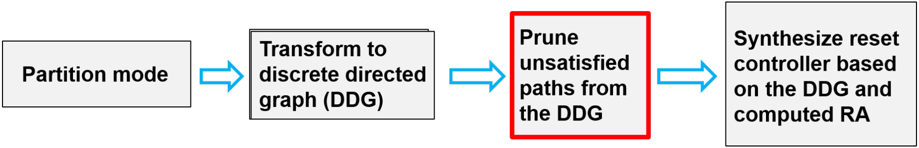
Transform to DDG

Partition mode ⇒ **Transform to discrete directed graph (DDG)** ⇒ **Prune unsatisfied paths from the DDG** ⇒ **Synthesize reset controller based on the DDG and computed RA**



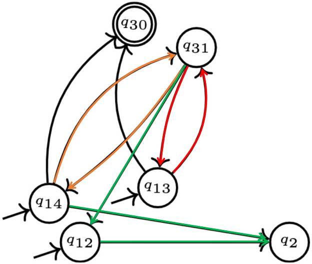
Main Idea

Prune Unsatisfied Paths



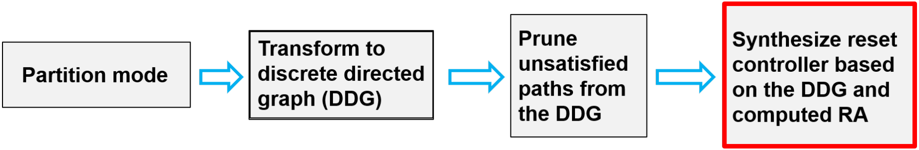
Two types:

- "unreachable paths": $\langle q_{14}, q_2\rangle$, $\langle q_{12}, q_2\rangle$, $\langle q_{31}, q_{12}, q_2\rangle$
- "simple loop": $\langle q_{14}, q_{31}\rangle$, $\langle q_{13}, q_{31}\rangle$

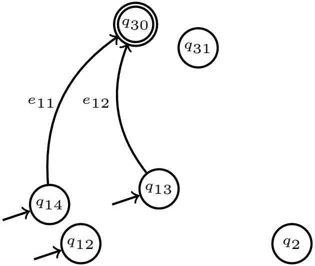


Main Idea

Synthesize Reset Controller



- $\mathcal{R}^r(e_2, \mathcal{G}(e_2) \setminus \mathcal{G}(e_1)) = R^m(e_{12}, \mathcal{G}^m(e_{12}))$,
- $\mathcal{R}^r(e_2, \mathcal{G}(e_2) \cap \mathcal{G}(e_1)) = R^m(e_{11}, \mathcal{G}^m(e_{11}))$,
- $\mathtt{Init}^r(q_1) = \mathtt{Init}^m(q_{13}) \cup \mathtt{Init}^m(q_{14})$.



Summary

Problem: We face

- controller synthesis providing a mechanism of correct-by-construction;
- many system goal unable being achieved by feedback controller and/or switching logic controller;
- **reset controller playing an important role in the design of CPS**, but little attention paid in the literature.

Status: We present

- **reset controller synthesis** w.r.t. safety possibly with liveness by reduction to **invariant generation and reach-aovid set computation**;
- **reset controller synthesis** by taking delays into account.

Future Work: We'd like to explore

- reset controller synthesis for more complex CPS, e.g. *stochastic hybrid systems, and so on*,
- optimal controller synthesis by integrating feedback, switching logic, and reset controller together.

Thanks & Questions?