Taming Delays in Cyber-Physical Systems

Towards a Theory of Networked Hybrid Systems

Naijun Zhan, Mingshuai Chen

Online Tutorial · ESWEEK · October 2022

Tutorial Speakers

Naijun Zhan

Distinguished Professor
State Key Lab. of Computer Science
Institute of Software, Chinese Academy of Sciences

Formal Methods · Cyber-Physical Systems ·
Program Verification · Modal and Temporal Logics

✉ znj@ios.ac.cn lcs.ios.ac.cn/~znj/

Mingshuai Chen

Postdoctoral Researcher
Dept. of Computer Science, RWTH Aachen University

Assistant Professor (2023)
College of Computer Sci. and Tech., Zhejiang University

Formal Methods · Quantitative Verification ·
Logic and Programming Theory · Cyber-Physical Systems

✉ chenms@cs.rwth-aachen.de moves.rwth-aachen.de/people/chenms/

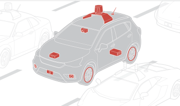
Cyber-Physical Systems (CPS)

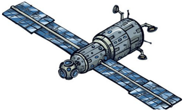
"[...] cyber-physical systems (CPS) refers to a new generation of systems with integrated computational and physical capabilities that can interact with humans through many new modalities. The ability to interact with, and expand the capabilities of, the physical world through computation, communication,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

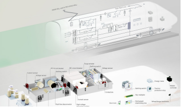
[Radhakisan Baheti and Helen Gill : CPS. The Impact of Control Technology, 2011]

Cyber-Physical Systems (CPS)

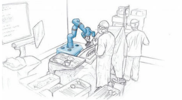
An open, interconnected form of embedded systems; many are safety-critical.

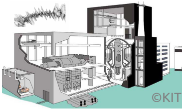

©TheOneBrief
automobiles


©clipartlogo
spacecrafts


©Sécheron
high-speed rail

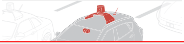
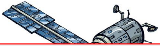
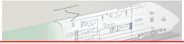

©VectorStock
nuclear reactors

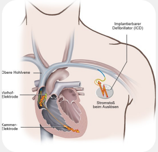

©Scoop.it
robot surgeon


©KIT
robust control

Cyber-Physical Systems (CPS)

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



212 patients died of defibrillator failure

(USA, 1997 – 2003)

40 passengers died plus 172 injured

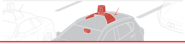
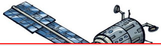
(China, 2011.7.23)

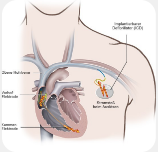
31 billion Yen loss on ASTRO-H

(Japan, 2016.3.26)

Cyber-Physical Systems (CPS)

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

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

Formal Methods

Joseph Sifakis

2007 Turing Awardee

"[...] the challenge of designing embedded systems offers a unique opportunity for reinvigorating computer science. The challenge, and thus the opportunity, spans the spectrum from theoretical foundations to engineering practice. To begin with, we need a mathematical basis for systems modeling and analysis which integrates both computation and physical constraints in a consistent, operative manner [...]"

— Embed. Syst. Design Challenge, invited talk at FM '06

Tom Henzinger

President, IST Austria

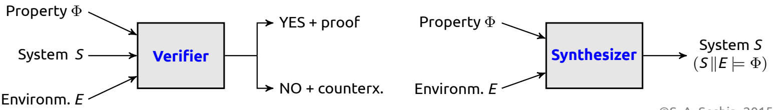
Formal Methods

"[...] the challenge of designing embedded systems offers a unique opportunity for reinvigorating computer science. The challenge, and thus the opportunity, spans the spectrum from theoretical foundations to engineering practice. To begin with, we need a mathematical basis for systems modeling and analysis which integrates both computation and physical constraints in a consistent, operative manner [...]"

— Embed. Syst. Design Challenge, invited talk at FM '06

Joseph Sifakis
2007 Turing Awardee

Tom Henzinger
President, IST Austria



©S. A. Seshia, 2015

Aim : Develop mathematically rigorous techniques for designing safety-critical CPS while pushing the limits of automation as far as possible.

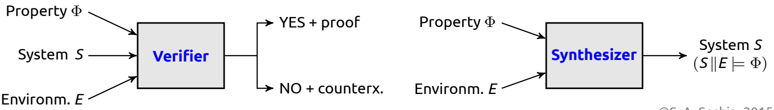
Formal Methods

Joseph Sifakis
2007 Turing Awardee

"[…] the challenge of designing embedded systems offers a unique opportunity for reinvigorating computer science. The challenge, and thus the opportunity, spans the spectrum from theoretical foundations to engineering practice. To begin with, we need a mathematical basis for systems modeling and analysis which integrates both computation and physical constraints in a consistent, operative manner […]"

— Embed. Syst. Design Challenge, invited talk at FM '06

Tom Henzinger
President, IST Austria



©S. A. Seshia, 2015

Safety, liveness, termination, cost, efficiency, … vs. intricacy, delays, randomness, uncertainty, …

Aim : Develop mathematically rigorous techniques for designing safety-critical CPS while pushing the limits of automation as far as possible.

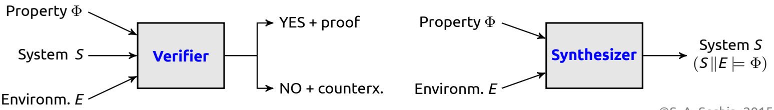
Formal Methods

Joseph Sifakis
2007 Turing Awardee

"[...] the challenge of designing embedded systems offers a unique opportunity for reinvigorating computer science. The challenge, and thus the opportunity, spans the spectrum from theoretical foundations to engineering practice. To begin with, we need a mathematical basis for systems modeling and analysis which integrates both computation and physical constraints in a consistent, operative manner [...]"

— Embed. Syst. Design Challenge, invited talk at FM '06

Tom Henzinger
President, IST Austria

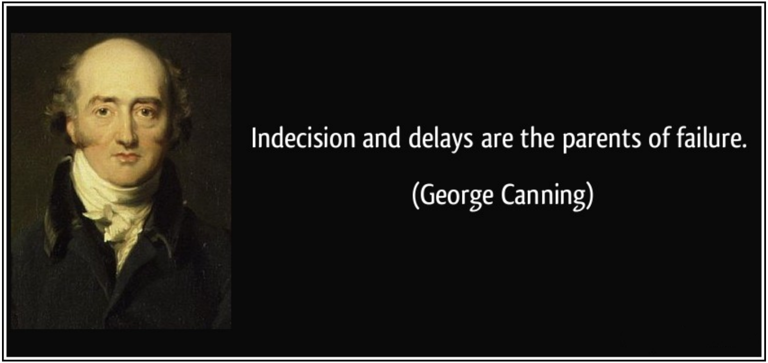


©S. A. Seshia, 2015

Safety, liveness, termination, cost, efficiency, ... vs. intricacy, delays, randomness, uncertainty, ...

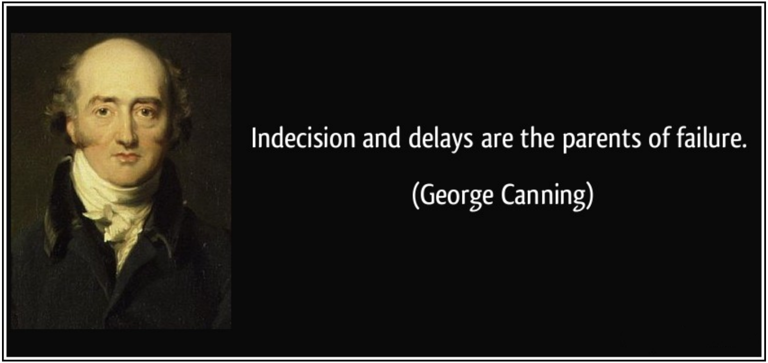
Aim : Develop mathematically rigorous techniques for designing safety-critical CPS while pushing the limits of automation as far as possible.

A Pearl of Wisdom



©izQuotes

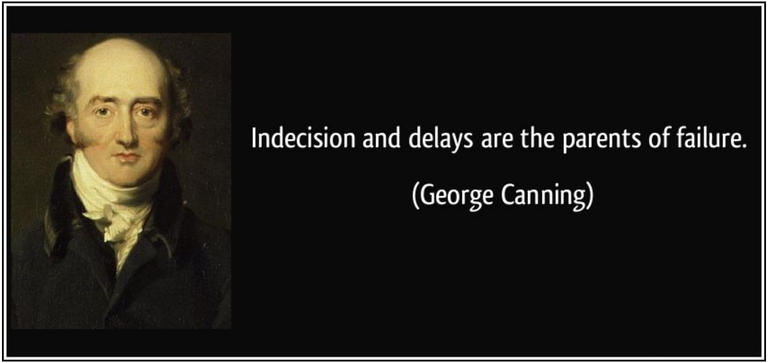
A Pearl of Wisdom



©izQuotes

- Only relevant to ordinary people's life ?

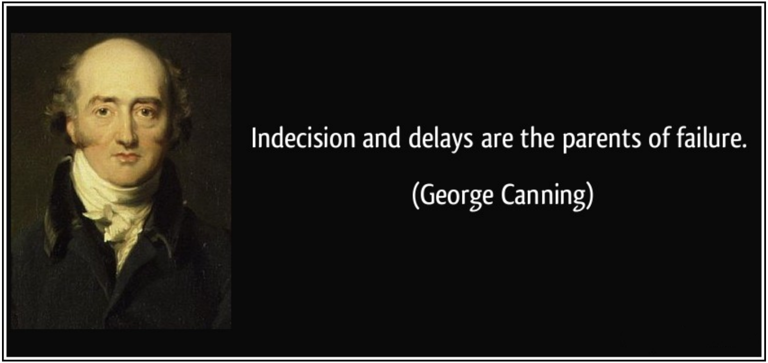
A Pearl of Wisdom



©izQuotes

- Only relevant to ordinary people's life?
- Or to scientists, in particular **comp. sci.** and **control folks**, too?

A Pearl of Wisdom

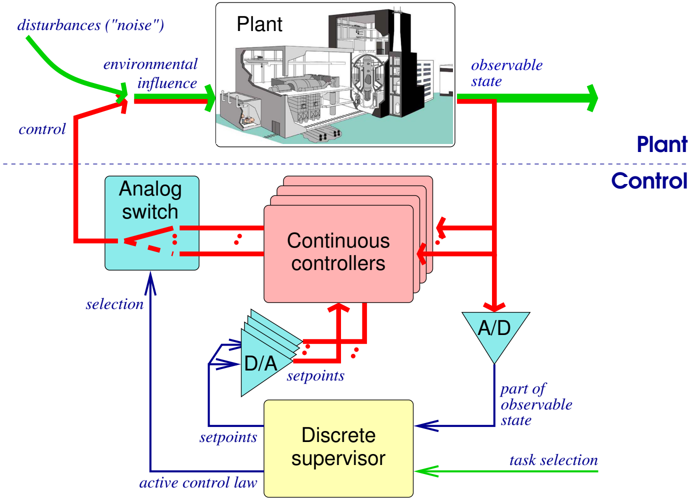


©izQuotes

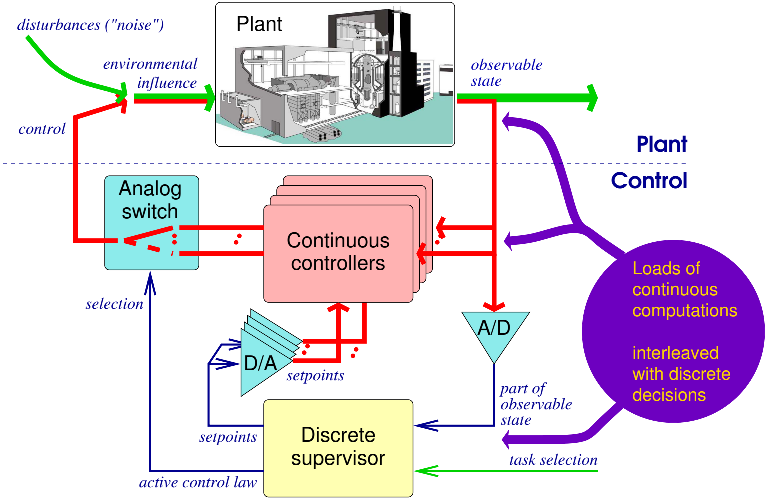
- Only relevant to ordinary people's life ?
- Or to scientists, in particular **comp. sci.** and **control folks**, too ?

Remember that Canning briefly **controlled** Great Britain !

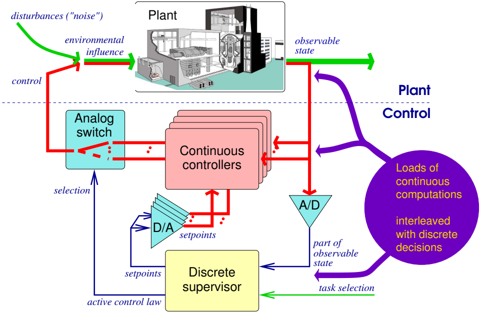
Hybrid Systems Modeling CPS



Hybrid Systems Modeling CPS



Hybrid Systems Modeling CPS

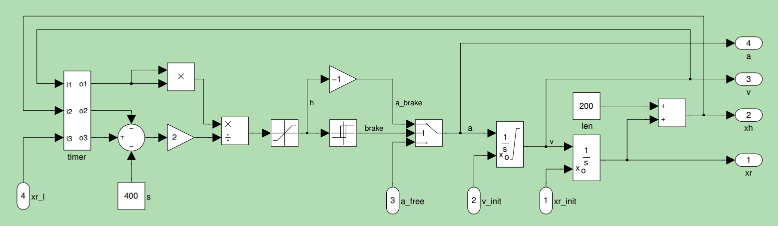


Crucial question : How do the controller and the plant **interact**?

Traditional answer : Coupling assumed to be (or at least modeled as) **delay-free** :

- **mode dynamics** is covered by the conjunction of individual ODEs;
- **switching btw. modes** is an immediate reaction to environmental conditions.

Instantaneous Coupling

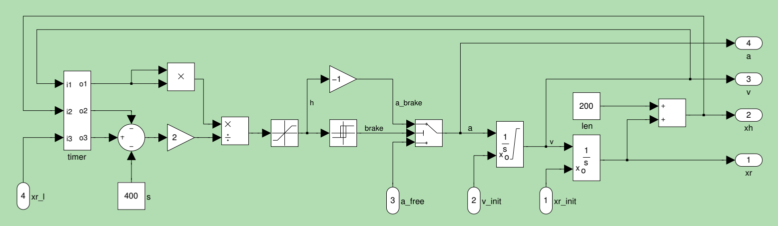


©ETCS-3

Following the tradition, the above (rather typical) Simulink model assumes

- delay-free coupling between all components;
- instantaneous feed-through within all functional blocks.

Instantaneous Coupling



©ETCS-3

Following the tradition, the above (rather typical) Simulink model assumes

- delay-free coupling between all components;
- instantaneous feed-through within all functional blocks.

Central questions :

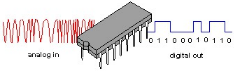
1. Is this realistic?
2. If not, does it have observable effects on control performance?
3. May those effects be detrimental or even harmful?

Q1 : Is Instantaneous Coupling Realistic?

Q1 : Is Instantaneous Coupling Realistic?

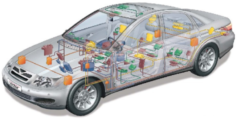
We are no better :
As soon as computer scientists enter the scene, serious delays are ahead …

Q1 : Is Instantaneous Coupling Realistic?



Digital control needs **A/D and D/A conversion**, which induces latency in signal forwarding.

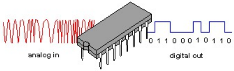
Digital **signal processing**, especially in complex sensors like CV, needs processing time, adding signal delays.



Networked control introduces communication latency into the feedback control loop.

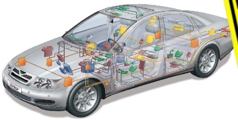
Harvesting, fusing, and forwarding data through **sensor networks** enlarge the communication latency by orders of magnitude.

Q1 : Is Instantaneous Coupling Realistic? – No.



Digital control needs **A/D and D/A conversion**, which induces latency in signal forwarding.

Digital sig… …pecially in complex sensors … e, adding signal delays.



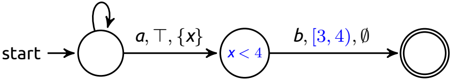
… communication latency … op.

Harvesting, fusing, and forwarding data through **sensor networks** enlarge the communication latency by orders of magnitude.

Q1a : Resultant Forms of Delay

Delayed reaction : Reaction to a stimulus is not immediate.

- Easy to model in timed/hybrid automata, etc. :



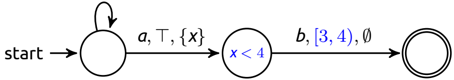
- Thus amenable to the pertinent analysis tools.

⇒ Not of interest today.

Q1a : Resultant Forms of Delay

Delayed reaction : Reaction to a stimulus is not immediate.

- Easy to model in timed/hybrid automata, etc. :



- Thus amenable to the pertinent analysis tools.

⇒ Not of interest today.

Network delay : Information of different age coexists and is queuing in the network when piped towards target.

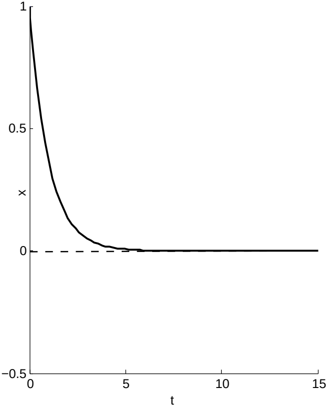
- End-to-end latency may exceed sampling intervals etc. by orders of magnitude.
- Not (efficiently) expressible in standard models.

⇒ Our theme today : discrete-time pipelined delay.

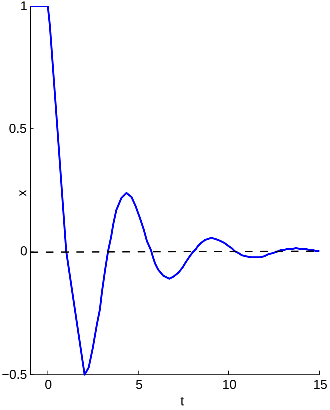
[Chen *et al.* : ATVA '18, Acta Inf. '21], [Bai *et al.* : HSCC '21, SCM '21];
[Zimmermann : LICS '18, GandALF '17], [Klein & Zimmermann : ICALP '15, CSL '15].

Q2 : Do Delays Have Observable Effects?

$$\begin{cases} \dot{x}(t) = -x(t) \\ x(0) = 1 \end{cases}$$



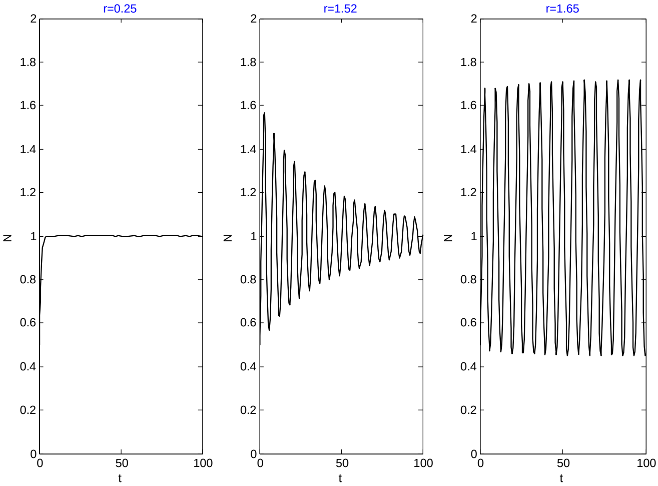
$$\begin{cases} \dot{x}(t) = -x(t-1) \\ x([-1,0]) \equiv 1 \end{cases}$$



Q2 : Do Delays Have Observable Effects?

Delayed logistic equation [G. Hutchinson, 1948] :

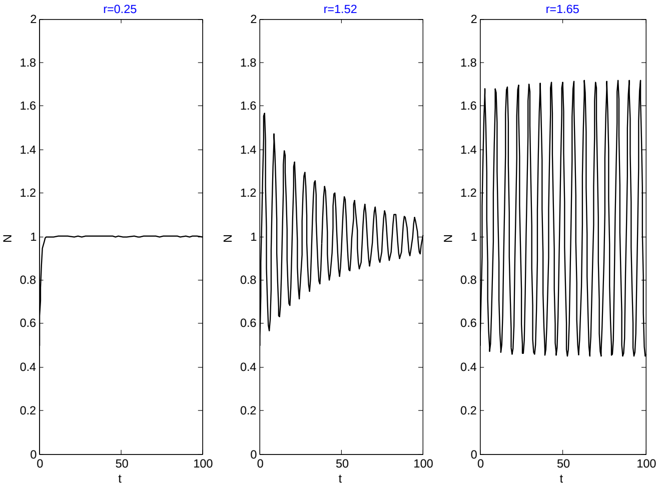
$$\dot{N}(t) = N(t)[1 - N(t - r)]$$



Q2 : Do Delays Have Observable Effects? – Yes, they have.

Delayed logistic equation [G. Hutchinson, 1948] :

$$\dot{N}(t) = N(t)[1 - N(t - r)]$$



Q3 : May the Effects be Harmful?

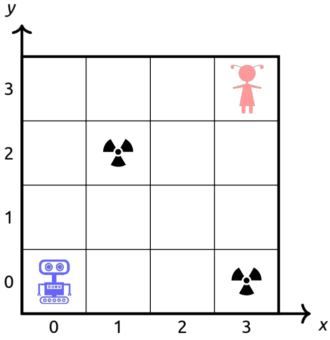


Figure – A robot escape game in a 4×4 room.

Q3 : May the Effects be Harmful?

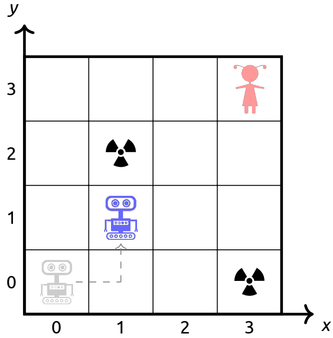


Figure – A robot escape game in a 4×4 room.

Q3 : May the Effects be Harmful ?

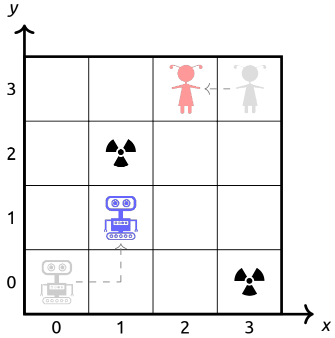


Figure – A robot escape game in a 4×4 room.

Q3 : May the Effects be Harmful?

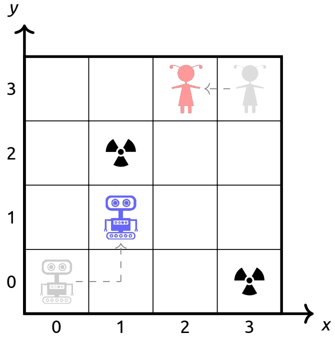


Figure – A robot escape game in a 4 × 4 room.

No delay :

Q3 : May the Effects be Harmful?

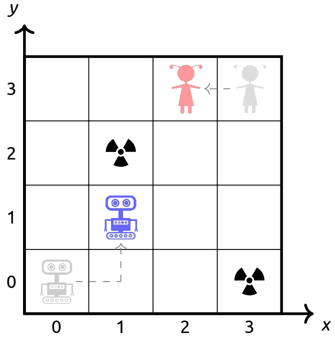


Figure – A robot escape game in a 4 × 4 room.

No delay :

Robot always wins by circling around the obstacle ☢ at (1,2).

Q3 : May the Effects be Harmful ?

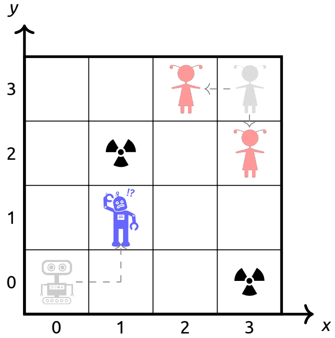


Figure – A robot escape game in a 4×4 room.

No delay :
Robot always wins by circling around the obstacle ☢ at (1,2).

1 step delay :

Q3 : May the Effects be Harmful?

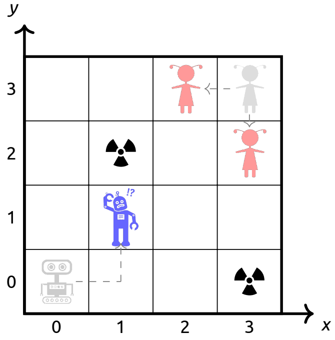


Figure – A robot escape game in a 4×4 room.

No delay :
Robot always wins by circling around the obstacle ☢ at (1,2).

1 step delay :
Robot wins by 1-step pre-decision.

Q3 : May the Effects be Harmful ?

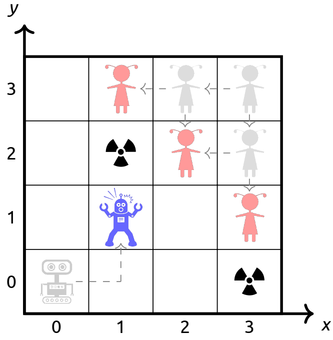


Figure – A robot escape game in a 4×4 room.

No delay :

Robot always wins by circling around the obstacle ☢ at (1,2).

1 step delay :

Robot wins by 1-step pre-decision.

2 steps delay :

Q3 : May the Effects be Harmful?

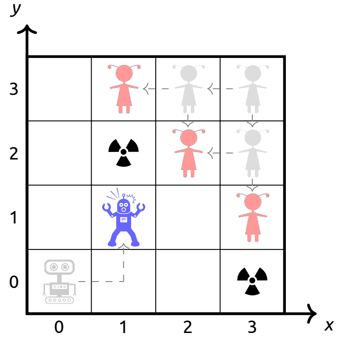


Figure – A robot escape game in a 4 × 4 room.

No delay :

Robot always wins by circling around the obstacle ☢ at (1,2).

1 step delay :

Robot wins by 1-step pre-decision.

2 steps delay :

Robot still wins, yet extra memory is needed.

Q3 : May the Effects be Harmful?

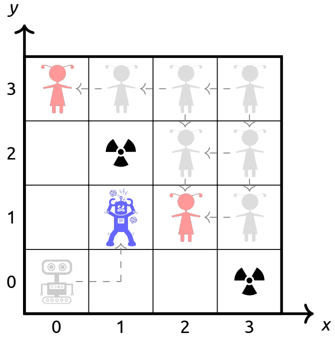


Figure – A robot escape game in a 4×4 room.

No delay :
Robot always wins by circling around the obstacle ☢ at (1,2).

1 step delay :
Robot wins by 1-step pre-decision.

2 steps delay :
Robot still wins, yet extra memory is needed.

3 steps delay :

Q3 : May the Effects be Harmful ?

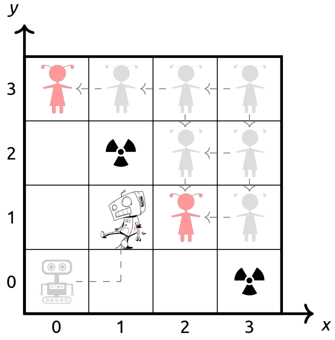


Figure – A robot escape game in a 4 × 4 room.

No delay :
Robot always wins by circling around the obstacle ☢ at (1,2).

1 step delay :
Robot wins by 1-step pre-decision.

2 steps delay :
Robot still wins, yet extra memory is needed.

3 steps delay :
Robot is unwinnable (uncontrollable) anymore.

Q3 : May the Effects be Harmful? – Yes, delays may well annihilate the control performance.

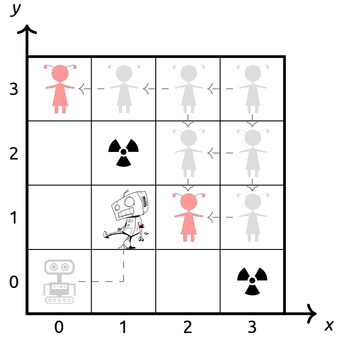


Figure – A robot escape game in a 4 × 4 room.

No delay :
Robot always wins by circling around the obstacle ☢ at (1,2).

1 step delay :
Robot wins by 1-step pre-decision.

2 steps delay :
Robot still wins, yet extra memory is needed.

3 steps delay :
Robot is unwinnable (uncontrollable) anymore.

Consequences

- Delays in feedback control loops are **ubiquitous**.
- They may well **invalidate** the safety/stability/...certificates obtained by verifying delay-free abstractions of the feedback control systems.

Automatic verification/synthesis methods addressing feedback delays in hybrid systems should therefore abound!

Consequences

- Delays in feedback control loops are **ubiquitous**.
- They may well **invalidate** the safety/stability/…certificates obtained by verifying delay-free abstractions of the feedback control systems.

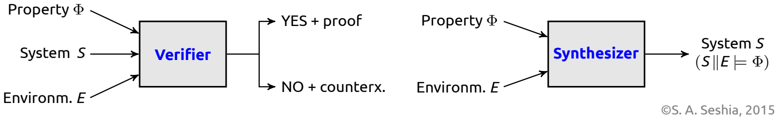
Automatic verification/synthesis methods addressing feedback delays in hybrid systems should therefore abound!

Surprisingly, they don't :

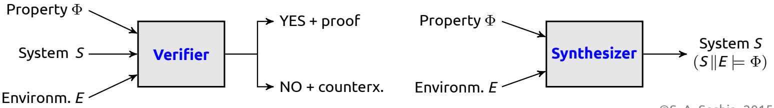
1. M. Peet, S. Lall : *Constructing Lyapunov functions for nonlinear DDEs using SDP* (NOLCOS'04)
2. S. Prajna, A. Jadbabaie : *Meth. f. safety verification of time-delay syst.* (CDC'05)
3. L. Zou, M. Fränzle, N. Zhan, P. N. Mosaad : *Autom. verific. of stabil. and safety* (CAV'15)
4. H. Trinh, P. T. Nam, P. N. Pathirana, H. P. Le : *On bwd.s and fwd.s reachable sets bounding for perturbed time-delay systems* (Appl. Math. & Comput. 269, '15)
5. Z. Huang, C. Fan, S. Mitra : *Bounded invariant verif. for time-delayed nonlinear networked dyn. syst.* (NAHS'16)
6. P. N. Mosaad, M. Fränzle, B. Xue : *Temporal logic verification for DDEs* (ICTAC'16)
7. M. Chen, M. Fränzle, Y. Li, P. N. Mosaad, N. Zhan : *Validat. simul.-based verific.* (FM'16)
8. B. Xue, P. N. Mosaad, M. Fränzle, M. Chen, Y. Li, N. Zhan : *Safe approx. of reach. sets for DDEs* (FORMATS'17)
9. E. Goubault, S. Putot, L. Sahlman : *Approximating flowpipes for DDEs* (CAV'18)
10. M. Chen, M. Fränzle, Y. Li, P. N. Mosaad, N. Zhan : *Synthesiz. controllers resilient to delayed interact.* (ATVA'18)
11. S. Feng, M. Chen, N. Zhan, M. Fränzle, B. Xue : *Taming delays in dyn. syst. : Unbounded verif. of DDEs* (CAV'19)
12. M. Chen, M. Fränzle, Y. Li, P. N. Mosaad, N. Zhan : *Indecision and delays are the parents of failure.* (Acta Inf.'21)
13. M. Zimmermann. LICS'18, GandALF'17], [F. Klein & M. Zimmermann. ICALP'15, CSL'15]

(plus a handful of related versions)

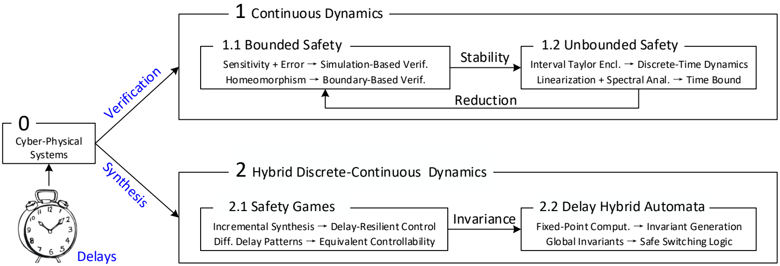
Overview of the Tutorial



Overview of the Tutorial



©S. A. Seshia, 2015



The Agenda

1. Verifying Safety of Delayed Differential Dynamics
2. Synthesizing Delay-Resilient Safe Control
3. Concluding Remarks

Verifying Safety of Delayed Differential Dynamics

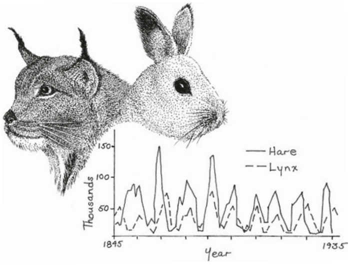
Addressing delayed feedback control in continuous dynamical systems

—Joint work w/ M. Fränzle, Y. Li, S. Feng, P. Mosaad, B. Xue, L. Zou—

Delayed Coupling in Differential Dynamics

©Wikipedia

Vito Volterra



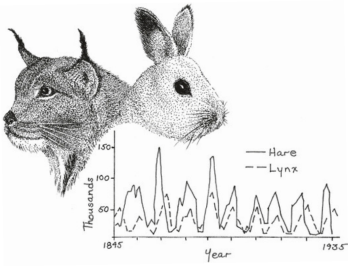
©J. Pastor, 2016

Predator-prey dynamics

Delayed Coupling in Differential Dynamics

©Wikipedia

Vito Volterra



©J. Pastor, 2016

Predator-prey dynamics

"Despite [...] very satisfactory state of affairs as far as [ordinary] differential equations are concerned, we are nevertheless forced to turn to the study of more complex equations. Detailed studies of the real world impel us, albeit reluctantly, to take account of the fact that the rate of change of physical systems depends not only on their present state, but also on their past history."

[Richard Bellman and Kenneth L. Cooke, 1963]

Delay Differential Equations (DDEs)

$$\begin{cases} \dot{\mathbf{x}}(t) = \boldsymbol{f}(\mathbf{x}(t), \mathbf{x}(t-r_1), \ldots, \mathbf{x}(t-r_k)), & t \in [0, \infty) \\ \mathbf{x}(t) = \phi(t), & t \in [-r_{\max}, 0] \end{cases}$$

Delay Differential Equations (DDEs)

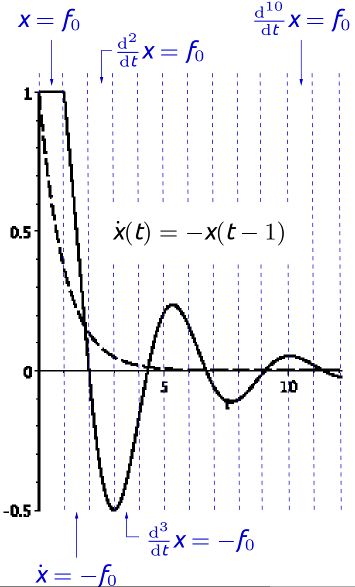
$$\begin{cases} \dot{\mathrm{x}}(t) = \boldsymbol{f}(\mathrm{x}(t), \mathrm{x}(t-r_1), \ldots, \mathrm{x}(t-r_k)), & t \in [0, \infty) \\ \mathrm{x}(t) = \phi(t), & t \in [-r_{\max}, 0] \end{cases}$$

Delay Differential Equations (DDEs)

$$\begin{cases} \dot{\mathrm{x}}(t) = \boldsymbol{f}(\mathrm{x}(t), \mathrm{x}(t-r_1), \ldots, \mathrm{x}(t-r_k)), & t \in [0, \infty) \\ \mathrm{x}(t) = \phi(t), & t \in [-r_{\max}, 0] \end{cases}$$

The unique *solution* (*trajectory*) : $\xi_\phi : [-r_{\max}, \infty) \to \mathbb{R}^n$.

Why DDEs are Hard(er)

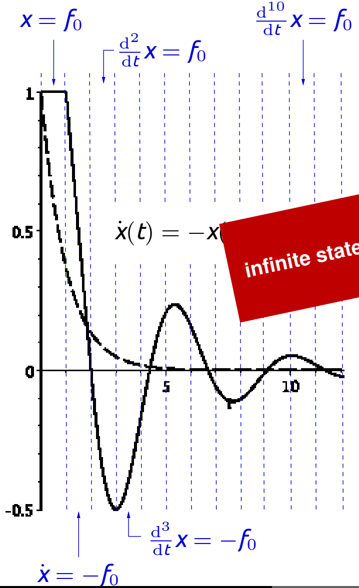


DDEs constitute a model of system dynamics beyond "state snapshots" :

- They feature "functional state" instead of state in the $\mathbb{R}^n$.
- Thus providing rather infallible, infinite-dimensional memory of the past.

N.B. : More complex transformations may be applied to the initial segment f_0 according to the DDE's right-hand side. f_0 will nevertheless hardly ever vanish from the state space.

Why DDEs are Hard(er)



Try only if infinite state no longer is scary enough to you!

DDEs co... model of system ... "state snapshots" :

... "functional state" ... state in the $\mathbb{R}^n$.

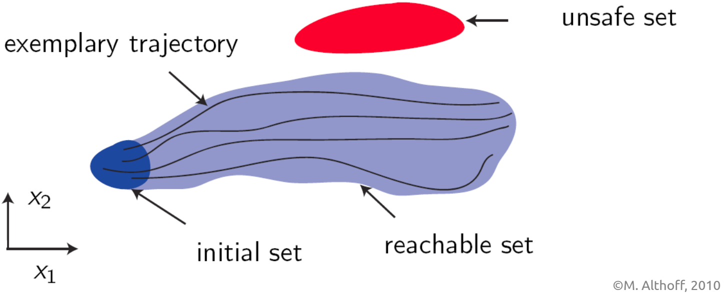
- Thus providing rather infallible, infinite-dimensional memory of the past.

N.B. : More complex transformations may be applied to the initial segment f_0 according to the DDE's right-hand side. f_0 will nevertheless hardly ever vanish from the state space.

Safety Verification Problem

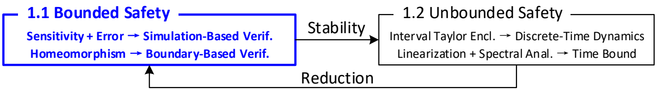
Given $T \in \mathbb{R}$, $\mathcal{X}_0 \subseteq \mathbb{R}^n$, $\mathcal{U} \subseteq \mathbb{R}^n$, weather

$$\forall \phi \in \{\phi \mid \phi(t) \in \mathcal{X}_0, \forall t \in [-r_{\max}, 0]\} : \quad \left(\bigcup_{t \leq T} \xi_\phi(t)\right) \cap \mathcal{U} = \emptyset \quad ?$$



- System is T-safe, if no trajectory enters $\mathcal{U}$ within $[-r_{\max}, T]$; Unbounded : ∞-safe.

Bounded Safety Verification of DDEs



Simulation-Based Verification Framework

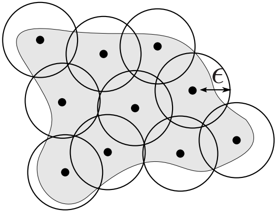


Figure – A finite ϵ-cover of the initial set of states.

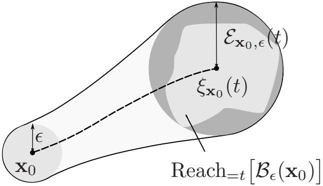
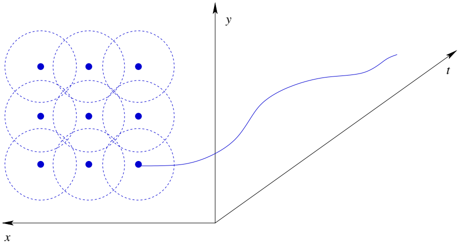


Figure – An over-approximation of the reachable set by bloating the simulation.

©A. Donzé & O. Maler, 2007

Validated Simulation-Based Verification

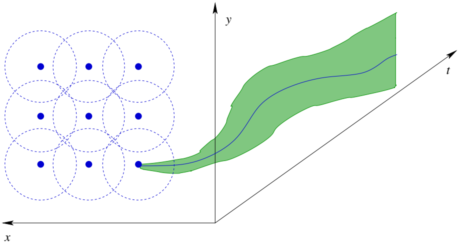
1. Do numerical **simulation** on a (sufficiently dense) sample of initial states.
2. Add (pessimistic) local-error by solving an **optimization** problem.
3. “Bloat” the resulting trajectories by **sensitivity analysis**.



⇒ M. Chen, M. Fränzle, Y. Li, P. N. Mosaad, N. Zhan : *Validat. simul.-based verific.*. FM'16.

Validated Simulation-Based Verification

1. Do numerical **simulation** on a (sufficiently dense) sample of initial states.
2. Add (pessimistic) local-error by solving an **optimization** problem.
3. "Bloat" the resulting trajectories by **sensitivity analysis**.

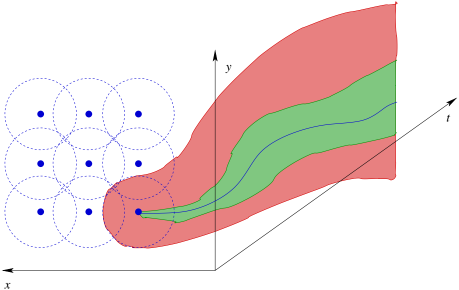


$$E(t) = \begin{cases} d_0\,, & \text{if } t = 0\,, \\ E(t_i) + (t - t_i)e_{i+1}\,, & \text{if } t \in [t_i, t_{i+1}]\,. \end{cases}$$

⇒ M. Chen, M. Fränzle, Y. Li, P. N. Mosaad, N. Zhan : *Validat. simul.-based verific.*. FM'16.

Validated Simulation-Based Verification

1. Do numerical **simulation** on a (sufficiently dense) sample of initial states.
2. Add (pessimistic) local-error by solving an **optimization** problem.
3. "Bloat" the resulting trajectories by **sensitivity analysis**.



$$\xi_{\mathbf{x}_0}(t) \in \mathcal{B}_{E(t)}\left(\frac{(t-t_i)\mathbf{y}_i + (t_{i+1}-t)\mathbf{y}_{i+1}}{t_{i+1}-t_i}\right), \forall t \in [t_i, t_{i+1}].$$

⇒ M. Chen, M. Fränzle, Y. Li, P. N. Mosaad, N. Zhan : *Validat. simul.-based verific.*. FM'16.

Example : Delayed Logistic Equation

[G. Hutchinson, 1948]

$$\dot{N}(t) = N(t)[1 - N(t - r)]$$

Example : Delayed Logistic Equation

[G. Hutchinson, 1948]

$$\dot{N}(t) = N(t)[1 - N(t - r)]$$

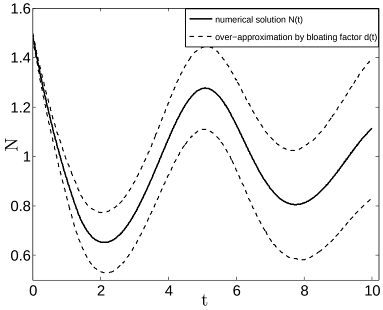


Figure – $\mathcal{X}_0 = \mathcal{B}_{0.01}(1.49)$, $r = 1.3$, $\tau_0 = 0.01$, $T = 10$s.

Example : Delayed Logistic Equation

[G. Hutchinson, 1948]

$$\dot{N}(t) = N(t)[1 - N(t - r)]$$

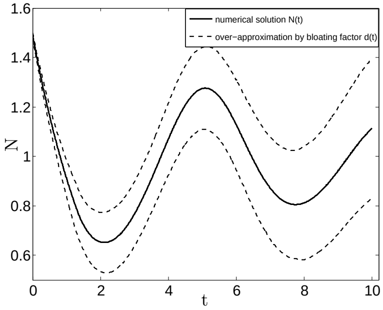


Figure – $\mathcal{X}_0 = \mathcal{B}_{0.01}(1.49)$, $r = 1.3$, $\tau_0 = 0.01$, $T = 10$s.

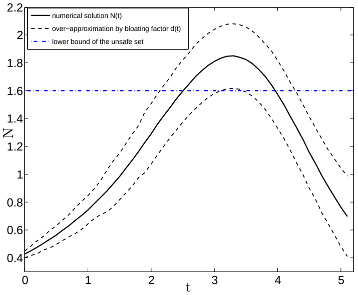
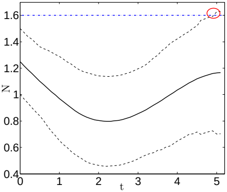


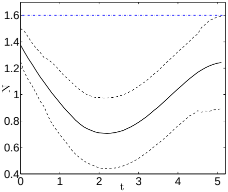
Figure – Over-approximation rigorously proving unsafe, with $r = 1.7$, $\mathcal{X}_0 = \mathcal{B}_{0.025}(0.425)$, $\tau_0 = 0.1$, $T = 5$s, $\mathcal{U} = \{N | N > 1.6\}$.

Example : Delayed Logistic Equation

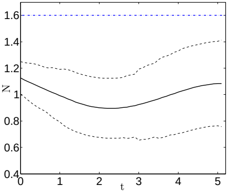
[G. Hutchinson, 1948]



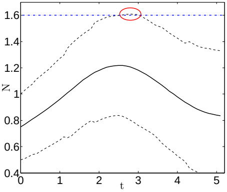
(a) An initial over-approximaion of trajectories starting from $\mathcal{B}_{0.225}(1.25)$. It overlaps with the unsafe set (s. circle). Initial set is consequently split (cf. Figs. 3b, 3c).



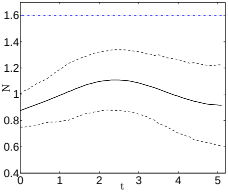
(b) All trajectories starting from $\mathcal{B}_{0.125}(1.375)$ are proven safe within the time bound, as the over-approximation does not intersect with the unsafe set.



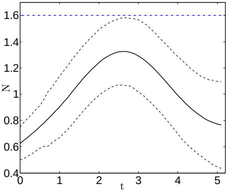
(c) Initial state set $\mathcal{B}_{0.125}(1.125)$ is verified to be safe as well.



(d) $\mathcal{B}_{0.25}(0.75)$ yields overlap w. unsafe; the ball is partitioned again (Figs. 3e, 3f).



(e) All trajectories originating from $\mathcal{B}_{0.125}(0.875)$ are provably safe.



(f) All trajectories originating from $\mathcal{B}_{0.125}(0.625)$ are provably safe as well.

Fig. 3: The logistic system is proven safe through 6 rounds of simulation with base stepsize $\tau_0 = 0.1$. Delay $r = 1.3$, initial state set $\mathcal{X}_0 = \{N | N \in [0.5, 1.5]\}$, time bound $T = 5$s, unsafe set $\{N | N > 1.6\}$.

Example : Delayed Microbial Growth

[S. F. Ellermeyer, 1994]

$$\begin{cases} \dot{S}(t) = 1 - S(t) - f(S(t))x(t) \\ \dot{x}(t) = e^{-r} f(S(t-r))x(t-r) - x(t) \end{cases}$$

Example : Delayed Microbial Growth

[S. F. Ellermeyer, 1994]

$$\begin{cases} \dot{S}(t) = 1 - S(t) - f(S(t))x(t) \\ \dot{x}(t) = e^{-r} f(S(t-r))x(t-r) - x(t) \end{cases}$$

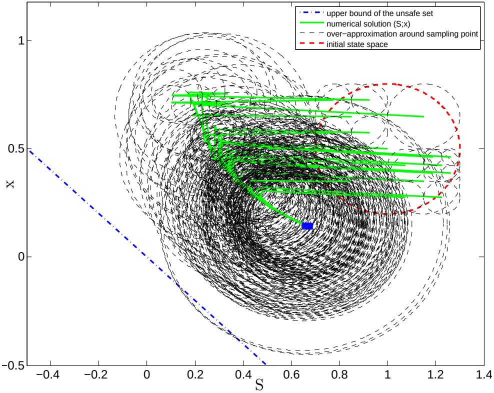
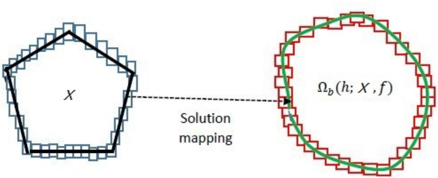


Figure – The microbial system is proven safe by 17 rounds of simulation with $\tau_0 = 0.45$. Here, $f(S) = 2eS/(1+S)$, $r = 0.9$, $\mathcal{X}_0 = \mathcal{B}_{0.3}((1; 0.5))$, $\mathcal{U} = \{(S; x) | S + x < 0\}$, $T = 8$s.

Boundary Propagation-Based Approximation of Reachable Sets

1. Impose a **homeomorphism** by bounding the time-lag through sensitivity analysis.
2. Compute an enclosure of the reachable set's **boundary**.
3. **Over- (under-)approximate** the reachable set by incl. (excl.) the enclosure.

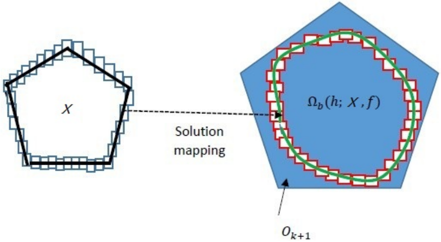


$$r \leq \min \left\{ \frac{\epsilon - 1}{\epsilon n^2 M' R}, \frac{\ln R}{2\sqrt{n} n M'}, \frac{\epsilon - 1}{\epsilon(n^2 MR + n^2 NR\epsilon)}, \frac{\ln \frac{R^2+1}{2}}{\sqrt{n}(2nM + n^2 NR\epsilon)} \right\}$$

⇒ B. Xue, P. Mosaad, M. Fränzle, M. Chen, Y. Li, N. Zhan : *Safe approx. of reachable sets for DDEs.* FORMATS'17.

Boundary Propagation-Based Approximation of Reachable Sets

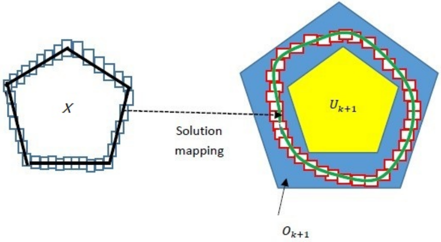
1. Impose a **homeomorphism** by bounding the time-lag through sensitivity analysis.
2. Compute an enclosure of the reachable set's **boundary**.
3. **Over- (under-)approximate** the reachable set by incl. (excl.) the enclosure.



⇒ B. Xue, P. Mosaad, M. Fränzle, M. Chen, Y. Li, N. Zhan : *Safe approx. of reachable sets for DDEs.* FORMATS'17.

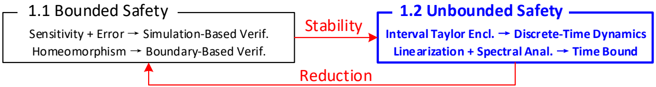
Boundary Propagation-Based Approximation of Reachable Sets

1. Impose a **homeomorphism** by bounding the time-lag through sensitivity analysis.
2. Compute an enclosure of the reachable set's **boundary**.
3. **Over- (under-)approximate** the reachable set by incl. (excl.) the enclosure.



⇒ B. Xue, P. Mosaad, M. Fränzle, M. Chen, Y. Li, N. Zhan : *Safe approx. of reachable sets for DDEs.* FORMATS'17.

Unbounded Safety Verification of DDEs



Unbounded Analysis for Simple DDE $\dot{\mathbf{x}}(t) = \boldsymbol{f}(\mathbf{x}(t-r))$

Main Ingredients

1. Generate Taylor series for the segment $\mathbf{x}|_{[nr,(n+1)r]}$ by integrating $\boldsymbol{f}(\mathbf{x})|_{[(n-1)r,nr]}$.
   - ☹ Degree of Taylor series grows indefinitely (and rapidly).
   - ☹ Computationally intractable.
   - ☹ Lacking means for analyzing unbounded behaviors.

⇒ L. Zou, M. Fränzle, N. Zhan, P. N. Mosaad : *Automatic stability and safety verification for DDEs*. CAV '15.

Unbounded Analysis for Simple DDE $\dot{\mathbf{x}}(t) = \boldsymbol{f}(\mathbf{x}(t-r))$

Main Ingredients

1. Generate Taylor series for the segment $\mathbf{x}\,|_{[nr,(n+1)r]}$ by integrating $\boldsymbol{f}(\mathbf{x})\,|_{[(n-1)r,nr]}$.
   - ☹ Degree of Taylor series grows indefinitely (and rapidly).
   - ☹ Computationally intractable.
   - ☹ Lacking means for analyzing unbounded behaviors.
2. Overapproximate segments by Interval Taylor Series (ITS) of fixed degree.
   - ☺ Tractable (if degree low enough).
   - ☺ Thus permits bounded model checking.
   - ☹ Still no immediate means for unbounded analysis.

⇒ L. Zou, M. Fränzle, N. Zhan, P. N. Mosaad : *Automatic stability and safety verification for DDEs*. CAV '15.

Unbounded Analysis for Simple DDE $\dot{\mathbf{x}}(t) = \boldsymbol{f}(\mathbf{x}(t-r))$

Main Ingredients

1. Generate Taylor series for the segment $\mathbf{x}\,|_{[nr,(n+1)r]}$ by integrating $\boldsymbol{f}(\mathbf{x})\,|_{[(n-1)r,nr]}$.
   - ☹ Degree of Taylor series grows indefinitely (and rapidly).
   - ☹ Computationally intractable.
   - ☹ Lacking means for analyzing unbounded behaviors.
2. Overapproximate segments by Interval Taylor Series (ITS) of fixed degree.
   - ☺ Tractable (if degree low enough).
   - ☺ Thus permits bounded model checking.
   - ☹ Still no immediate means for unbounded analysis.
3. Extract operator computing next ITS from current one; analyse its properties.
   - ☺ Unbounded safety and stability analysis become feasible.

⇒ L. Zou, M. Fränzle, N. Zhan, P. N. Mosaad : *Automatic stability and safety verification for DDEs*. CAV '15.

Analysis of a Linear DDE by Example

Recall the DDE $\dot{x}(t) = -x(t-1)$ with the initial condition $x([0,1]) \equiv 1$.

Analysis of a Linear DDE by Example

Recall the DDE $\dot{x}(t) = -x(t-1)$ with the initial condition $x([0,1]) \equiv 1$.

- Segmentwise integration yields

$$x(n+t) = x(n) + \int_{n-1}^{n-1+t} -x(s)\,\mathrm{d}s, \quad t \in [0,1].$$

Analysis of a Linear DDE by Example

Recall the DDE $\dot{x}(t) = -x(t-1)$ with the initial condition $x([0,1]) \equiv 1$.

- Segmentwise integration yields

$$x(n+t) = x(n) + \int_{n-1}^{n-1+t} -x(s)\,\mathrm{d}s, \quad t \in [0,1].$$

- Rename and shift $x|_{[n,n+1]}$, with $n \in \mathbb{N}$, to $f_n \colon [0,1] \mapsto \mathbb{R}$ by setting $f_n(t) \mathrel{\widehat{=}} x(n+t)$ for $t \in [0,1]$:

$$f_n(t) = f_{n-1}(1) + \int_0^t -f_{n-1}(s)\,\mathrm{d}s, \quad t \in [0,1].$$

Analysis of a Linear DDE by Example

Recall the DDE $\dot{x}(t) = -x(t-1)$ with the initial condition $x([0,1]) \equiv 1$.

- Segmentwise integration yields

$$x(n+t) = x(n) + \int_{n-1}^{n-1+t} -x(s)\,\mathrm{d}s, \quad t \in [0,1].$$

- Rename and shift $x|_{[n,n+1]}$, with $n \in \mathbb{N}$, to $f_n\colon [0,1] \mapsto \mathbb{R}$ by setting $f_n(t) \mathrel{\widehat{=}} x(n+t)$ for $t \in [0,1]$:

$$f_n(t) = f_{n-1}(1) + \int_0^t -f_{n-1}(s)\,\mathrm{d}s, \quad t \in [0,1].$$

☹ f_n is a polynomial of degree n, i.e., degree 86,400 after a day, ...

☹ Intractable beyond the first few steps!

Analysis of a Linear DDE by Example

- Employ interval Taylor series to enclose the segmentwise solutions by Taylor series of fixed degree
  - fixing degree 2, e.g., yields template $f_n(t) = a_{n0} + a_{n1} * t + a_{n2} * t^2$,
  - interval coefficients a_{nj} incorporate the approximation error.

Analysis of a Linear DDE by Example

- Employ interval Taylor series to enclose the segmentwise solutions by Taylor series of fixed degree
  - fixing degree 2, e.g., yields template $f_n(t) = a_{n0} + a_{n1} * t + a_{n2} * t^2$,
  - interval coefficients a_{n_i} incorporate the approximation error.
- For computing the ITS, we need to obtain the first and second derivatives $f_{n+1}^{(1)}(t)$ and $f_{n+1}^{(2)}(t)$ based on f_n :

$$
\begin{aligned}
f_{n+1}^{(1)}(t) &= -f_n(t) = -a_{n0} - a_{n1} * t - a_{n2} * t^2, \\
f_{n+1}^{(2)}(t) &= \frac{\mathrm{d}}{\mathrm{d}t} f_{n+1}^{(1)}(t) = -a_{n1} - 2 * a_{n2} * t.
\end{aligned}
$$

Analysis of a Linear DDE by Example

- Employ interval Taylor series to enclose the segmentwise solutions by Taylor series of fixed degree
  - fixing degree 2, e.g., yields template $f_n(t) = a_{n0} + a_{n1} * t + a_{n2} * t^2$,
  - interval coefficients a_{n_i} incorporate the approximation error.
- For computing the ITS, we need to obtain the first and second derivatives $f^{(1)}_{n+1}(t)$ and $f^{(2)}_{n+1}(t)$ based on f_n :

$$\begin{aligned} f^{(1)}_{n+1}(t) &= -f_n(t) = -a_{n0} - a_{n1} * t - a_{n2} * t^2, \\ f^{(2)}_{n+1}(t) &= \frac{\mathrm{d}}{\mathrm{d}t} f^{(1)}_{n+1}(t) = -a_{n1} - 2 * a_{n2} * t. \end{aligned}$$

- Using a Lagrange remainder with fresh variable $\eta_n \in [0, 1]$, we obtain

$$\begin{aligned} f_{n+1}(t) &= f_n(1) + \frac{f^{(1)}_n(0)}{1!} * t + \frac{f^{(2)}_n(\eta_n)}{2!} * t^2 \\ &= (a_{n0} + a_{n1} + a_{n2}) - a_{n0} * t - \frac{a_{n1} + 2 * a_{n2} * \eta_n}{2} * t^2. \end{aligned}$$

Analysis of a Linear DDE by Example

- Substituting $f_{n+1}(t)$ by its Taylor form ${a_{n+1}}_0 + {a_{n+1}}_1 * t + {a_{n+1}}_2 * t^2$ and matching coefficients, one obtains a time-variant, parametric linear operator

$$\begin{bmatrix} {a_{n+1}}_0 \\ {a_{n+1}}_1 \\ {a_{n+1}}_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & 0 \\ 0 & -\frac{1}{2} & -\eta_n \end{bmatrix} * \begin{bmatrix} a_{n0} \\ a_{n1} \\ a_{n2} \end{bmatrix}$$

  which can be made time-invariant by replacing η_n with its interval $[0, 1]$.

☺ Have thus obtained a **discrete-time** interval-linear system $\mathrm{a}' = \mathcal{M}\mathrm{a}$!

Stability of Linear DDEs

Observation : The global solution x to the DDE stabilizes asymptotically
if the sequence of segments f_n converges to $\mathbf{0}$,
iff the coefficients $\mathbf{A}_n$ of the interval Taylor forms converge to $\mathbf{0}$.

Stability of Linear DDEs

Observation : The global solution x to the DDE stabilizes asymptotically
if the sequence of segments f_n converges to $\mathbf{0}$,
iff the coefficients A_n of the interval Taylor forms converge to $\mathbf{0}$.

Consequence : Can reduce asymptotic stability verification of the DDE to that of the interval-linear time-invariant system $\mathbf{A}' = \mathcal{M}\mathbf{A}$, which boils down to

Theorem (J. Daafouz and J. Bernussou, 2001)

The time-variant system $\mathrm{x}(n+1) = T(\boldsymbol{\eta}(n)) * \mathrm{x}(n)$, $T(\boldsymbol{\eta}(n)) = \sum_{i=1}^{q} \boldsymbol{\eta}_i(n) * T_i$, *with* $\boldsymbol{\eta}_i(n) \geq 0$, $\sum_{i=1}^{q} \boldsymbol{\eta}_i(n) = 1$, *is asymptotically/robustly stable iff there exist symmetric positive definite matrices* S_i, S_j *and matrices* G_i *of appropriate dimensions s.t.*

$$\begin{bmatrix} G_i + G_i^\mathsf{T} & G_i^\mathsf{T}\, T_i^\mathsf{T} \\ T_i\, G_i & S_j \end{bmatrix} > \mathbf{0}$$

for all $i = 1, \ldots, N$ *and* $j = 1, \ldots, N$. *Moreover, the corresponding Lyapunov function is*

$$V(\mathrm{x}(n), \boldsymbol{\eta}(n)) = \mathrm{x}(n)^\mathsf{T} * (\sum_{i=1}^{q} \boldsymbol{\eta}_i(n) * S_i^{-1}) * \mathrm{x}(n).$$

Just requires some technicalities to obtain appropriate interval forms for applicability of Rohn's method for solving linear interval inequalities.

Unbounded Safety Verification for Linear DDEs

☺ Verifying **unbounded safety** $\Box\mathcal{S}$ can be accomplished by

1. generating a Lyapunov function $V(\mathbf{A}, \eta)$ by above method,
2. computing a barrier value for the safe set by letting iSAT search for the largest c such that $V(\mathbf{A}(n), \eta(n)) \leq c \wedge \neg\mathcal{S}(f_n(t))$ is unsatisfiable,

⇒ existence of such c implies that $V(\mathbf{A}(n), \eta_n) \leq c \rightarrow \mathcal{S}(f_n(t))$ holds.

Unbounded Safety Verification for Linear DDEs

☺ Verifying **unbounded safety** $\square\mathcal{S}$ can be accomplished by

1. generating a Lyapunov function $V(\mathbf{A}, \eta)$ by above method,
2. computing a barrier value for the safe set by letting iSAT search for the largest c such that $V(\mathbf{A}(n), \eta(n)) \leq c \wedge \neg\mathcal{S}(f_n(t))$ is unsatisfiable,

⇒ existence of such c implies that $V(\mathbf{A}(n), \eta_n) \leq c \rightarrow \mathcal{S}(f_n(t))$ holds.

3. calculating a safe bound on the minimum reduction d_m on the condition $V(\mathbf{A}(n), \eta(n)) \geq c$, i.e.

$$d_m = \min\{V(\mathbf{A}(n), \eta(n)) - V(\mathbf{A}(n+1), \eta_{n+1}) \mid V(\mathbf{A}(n), \eta_n) \geq c\},$$

by iSAT optimization.

⇒ Existence of such d_m implies that after $k \mathrel{\widehat{=}} \max\left(\frac{V(\mathbf{A}(0), 0) - c}{d_m}, \frac{V(\mathbf{A}(0), 1) - c}{d_m}\right)$ we can be sure to reside inside the safety region $\mathcal{S}$.

Unbounded Safety Verification for Linear DDEs

☺ Verifying **unbounded safety** $\Box\mathcal{S}$ can be accomplished by

1. generating a Lyapunov function $V(\mathbf{A}, \eta)$ by above method,
2. computing a barrier value for the safe set by letting iSAT search for the largest c such that $V(\mathbf{A}(n), \eta(n)) \leq c \wedge \neg\mathcal{S}(f_n(t))$ is unsatisfiable,

⇒ existence of such c implies that $V(\mathbf{A}(n), \eta_n) \leq c \to \mathcal{S}(f_n(t))$ holds.

3. calculating a safe bound on the minimum reduction d_m on the condition $V(\mathbf{A}(n), \eta(n)) \geq c$, i.e.

$$d_m = \min\{V(\mathbf{A}(n), \eta(n)) - V(\mathbf{A}(n+1), \eta_{n+1}) \mid V(\mathbf{A}(n), \eta_n) \geq c\},$$

by iSAT optimization.

⇒ Existence of such d_m implies that after $k \mathrel{\widehat{=}} \max\left(\frac{V(\mathbf{A}(0), 0) - c}{d_m}, \frac{V(\mathbf{A}(0), 1) - c}{d_m}\right)$ we can be sure to reside inside the safety region $\mathcal{S}$.

4. Pursuing BMC for the first k steps, which completes proving unbounded invariance.

Multidimensional Polynomial DDEs

Consider a DDE of the form

$$\dot{\mathbf{x}}(t + r) = \boldsymbol{g}(\mathbf{x}(t)),\ \forall t \in [0, r] \colon \mathbf{x}(t) = \mathbf{p}_0(t),$$

where $\boldsymbol{g}$ and $\mathbf{p}_0(t)$ are vectors of polynomials in $\mathbb{R}^m[\mathbf{x}]$.

Multidimensional Polynomial DDEs

Consider a DDE of the form

$$\dot{\mathbf{x}}(t+r) = \boldsymbol{g}(\mathbf{x}(t)),\ \forall t \in [0, r] \colon \mathbf{x}(t) = \mathbf{p}_0(t),$$

where $\boldsymbol{g}$ and $\mathbf{p}_0(t)$ are vectors of polynomials in $\mathbb{R}^m[\mathbf{x}]$.

- Generalizing the linear case, the Lie derivatives $\boldsymbol{f}_{n+1}^{(1)}, \boldsymbol{f}_{n+1}^{(2)}, \ldots, \boldsymbol{f}_{n+1}^{(k)}$ can now be computed *symbolically* as follows :

$$\boldsymbol{f}_{n+1}^{(1)}(t) = \boldsymbol{g}(\boldsymbol{f}_n(t)), \quad \boldsymbol{f}_{n+1}^{(2)}(t) = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{f}_{n+1}^{(1)} = \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{g}(\boldsymbol{f}_n(t)), \ldots$$

- The corresponding Taylor expansion of $\boldsymbol{f}_{n+1}(t)$ with degree k is

$$\boldsymbol{f}_{n+1}(t) = \boldsymbol{f}_n(r) + \frac{\boldsymbol{f}_{n+1}^{(1)}(0)}{1!} * t + \cdots + \frac{\boldsymbol{f}_{n+1}^{(k-1)}(0)}{(k-1)!} * t^i + \frac{\boldsymbol{f}_{n+1}^{(k)}(\boldsymbol{\eta}_n)}{k!} * t^k,$$

where $\boldsymbol{\eta}_n$ is a vector ranging over $[0, r]^m$.

Multidimensional Polynomial DDEs

- Akin to the linear case, the above equation can be rephrased as a time-invariant *polynomial* interval operator

$$\mathbf{A}(n+1) = \mathbf{P}(\mathbf{A}(n), [0, r]), \qquad (\dagger)$$

where $\mathbf{P}$ this time is a vector of polynomials.

Multidimensional Polynomial DDEs

- Akin to the linear case, the above equation can be rephrased as a time-invariant *polynomial* interval operator

$$\mathbf{A}(n+1) = \mathbf{P}(\mathbf{A}(n), [0, r]), \qquad (\dagger)$$

where $\mathbf{P}$ this time is a vector of polynomials.

☺ Apply polynomial constraint solving to
  - pursue BMC exactly as before, unwinding relation ($\dagger$),
  - find a relaxed Lyapunov function by instantiating a polynomial Lyapunov function template w.r.t. ($\dagger$), using the method in [S. Ratschan and Z. She, SIAM J. of Control and Optimiz., 2010],
  - compute barrier values for a safe set,
  - ...

Stability of General Linear Dynamics by Spectral Analysis

For linear DDEs :

$$\dot{\mathrm{x}}(t) = A\mathrm{x}(t) + B\mathrm{x}(t - r)$$

Stability of General Linear Dynamics by Spectral Analysis

For linear DDEs :

$$\dot{\mathrm{x}}\left(t\right)=A\mathrm{x}\left(t\right)+B\mathrm{x}\left(t-r\right)$$

The characteristic equation :

$$\det\left(\lambda I-A-B\mathrm{e}^{-r\lambda}\right)=0$$

Stability of General Linear Dynamics by Spectral Analysis

For linear DDEs :

$$\dot{\mathrm{x}}(t) = A\mathrm{x}(t) + B\mathrm{x}(t - r)$$

The characteristic equation :

$$\det\left(\lambda I - A - B\mathrm{e}^{-r\lambda}\right) = 0$$

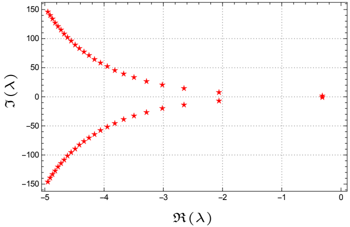
Stability of General Linear Dynamics by Spectral Analysis

For linear DDEs :

$$\dot{\mathrm{x}}(t) = A\mathrm{x}(t) + B\mathrm{x}(t - r)$$

The characteristic equation :

$$\det\left(\lambda I - A - B\mathrm{e}^{-r\lambda}\right) = 0$$



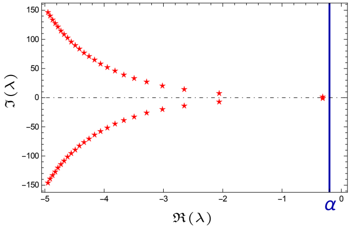
Stability of General Linear Dynamics by Spectral Analysis

For linear DDEs :

$$\dot{\mathrm{x}}(t) = A\mathrm{x}(t) + B\mathrm{x}(t-r)$$

The characteristic equation :

$$\det\left(\lambda I - A - B\mathrm{e}^{-r\lambda}\right) = 0$$



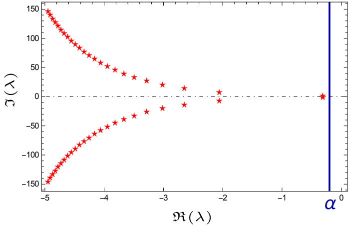
Stability of General Linear Dynamics by Spectral Analysis

For linear DDEs :

$$\dot{x}(t) = Ax(t) + Bx(t-r)$$

The characteristic equation :

$$\det\left(\lambda I - A - Be^{-r\lambda}\right) = 0$$



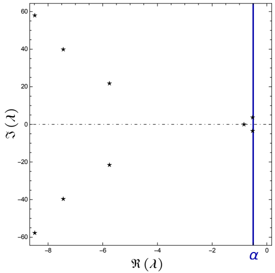
Globally exponentially stable if $\forall\lambda\colon \mathfrak{R}(\lambda) < 0$, i.e.,

$$\exists K > 0.\, \exists\alpha < 0\colon \ \left\|\xi_\phi(t)\right\| \leq K\|\phi\|\, e^{\alpha t}, \quad \forall t \geq 0,\ \forall\phi \in \mathcal{C}_r$$

Reduction to Bounded Verification

[PD-Controller, E. Goubault et al., CAV'18]

1. Identify the **rightmost eigenvalue** (and hence α) and construct K.
2. Compute T^* based on the **exponential estimation** spanned by α and K.
3. Reduce to **bounded verifi.**, i.e., $\forall T > T^*$, ∞-safe $\Longleftrightarrow$ T-safe.



$$K = \hat{K}\left(1 + \|B\| \int_0^r e^{-\alpha\tau}\, d\tau\right) \|\mathcal{X}\|$$

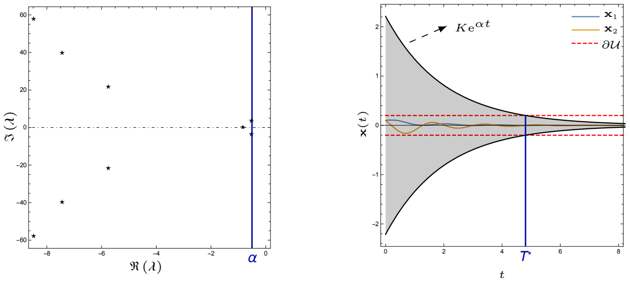
$$\hat{K} = \frac{1}{2\pi}\left(\int_{-M}^{M} \left\| \mathcal{O}\left(\frac{1}{(\alpha + i\nu)^2}\right)\right\| d\nu + \frac{8n}{M}\left(\|A\| + \|B\|\, e^{-r\alpha}\right)\right) + 1_0(\alpha)$$

⇒ S. Feng, M. Chen, N. Zhan, M. Fränzle, B. Xue : *Taming delays in dyn. syst. : Unbounded verif. of DDEs.* CAV'19.

Reduction to Bounded Verification

[PD-Controller, E. Goubault et al., CAV '18]

1. Identify the **rightmost eigenvalue** (and hence α) and construct K.
2. Compute T^* based on the **exponential estimation** spanned by α and K.
3. Reduce to **bounded verifi.**, i.e., $\forall T > T^*$, ∞-safe $\Longleftrightarrow$ T-safe.

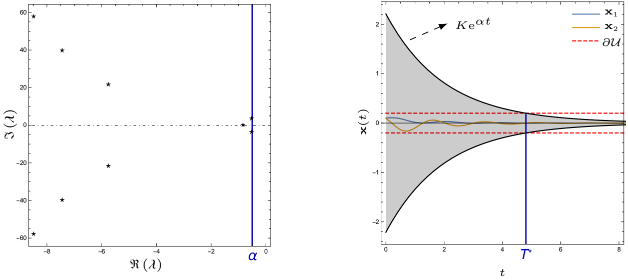


⇒ S. Feng, M. Chen, N. Zhan, M. Fränzle, B. Xue : *Taming delays in dyn. syst. : Unbounded verif. of DDEs.* CAV '19.

Reduction to Bounded Verification

[PD-Controller, E. Goubault et al., CAV '18]

1. Identify the **rightmost eigenvalue** (and hence α) and construct K.
2. Compute T^* based on the **exponential estimation** spanned by α and K.
3. Reduce to **bounded verifi.**, i.e., $\forall T > T^*$, ∞-safe $\iff$ T-safe.



⇒ S. Feng, M. Chen, N. Zhan, M. Fränzle, B. Xue : *Taming delays in dyn. syst. : Unbounded verif. of DDEs.* CAV '19.

Stability of General Nonlinear Dynamics by Linearization

For nonlinear DDEs :

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \boldsymbol{f}(\mathbf{x}(t), \mathbf{x}(t-r)) \\ &= A\mathbf{x} + B\mathbf{y} + \boldsymbol{g}(\mathbf{x}, \mathbf{y}), \text{ with } A = \boldsymbol{f}_{\mathbf{x}}(\mathbf{0}, \mathbf{0}), B = \boldsymbol{f}_{\mathbf{y}}(\mathbf{0}, \mathbf{0})\end{aligned}$$

Stability of General Nonlinear Dynamics by Linearization

For nonlinear DDEs :

$$\begin{aligned}\dot{\mathrm{x}}(t) &= \boldsymbol{f}(\mathrm{x}(t), \mathrm{x}(t-r)) \\ &= A\mathrm{x} + B\mathrm{y} + \boldsymbol{g}(\mathrm{x}, \mathrm{y}), \text{ with } A = \boldsymbol{f}_{\mathrm{x}}(\mathbf{0}, \mathbf{0}), B = \boldsymbol{f}_{\mathrm{y}}(\mathbf{0}, \mathbf{0})\end{aligned}$$

The linearization yields

$$\dot{\mathrm{x}}(t) = A\mathrm{x}(t) + B\mathrm{x}(t-r)$$

Stability of General Nonlinear Dynamics by Linearization

For nonlinear DDEs :

$$\begin{aligned} \dot{\mathrm{x}}(t) &= \boldsymbol{f}(\mathrm{x}(t), \mathrm{x}(t-r)) \\ &= A\mathrm{x} + B\mathrm{y} + \boldsymbol{g}(\mathrm{x}, \mathrm{y}), \text{ with } A = \boldsymbol{f}_{\mathrm{x}}(\mathbf{0}, \mathbf{0}), B = \boldsymbol{f}_{\mathrm{y}}(\mathbf{0}, \mathbf{0}) \end{aligned}$$

The linearization yields

$$\dot{\mathrm{x}}(t) = A\mathrm{x}(t) + B\mathrm{x}(t-r)$$

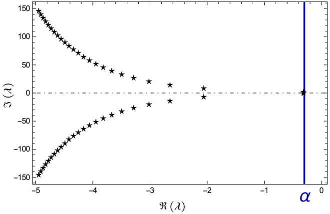
Locally exponentially stable if $\forall \lambda \colon \Re(\lambda) < 0$, i.e.,

$$\exists \delta > 0.\, \exists K > 0.\, \exists \alpha < 0 \colon \; \|\phi\| \leq \delta \implies \left\|\boldsymbol{\xi}_{\phi}(t)\right\| \leq K\|\phi\|\, \mathrm{e}^{\alpha t/2}, \quad \forall t \geq 0$$

Reduction to Bounded Verification

[Population Dynamics, G. Hutchinson, 1948]

1. Identify the **rightmost eigenvalue** (and hence α), then construct K and δ.
2. Compute T^*, as well as T' (by bounded verifiers) s.t. $\|\Omega\| < \delta$ within T'.
3. Reduce to **bounded verifi.**, i.e., $\forall T > T' + T^*$, ∞-safe $\iff$ T-safe.



$$\delta = \min \left\{ \delta_\epsilon, \delta_\epsilon / \left(\hat{K} e^{-r\alpha} \left(1 + \|B\| \int_0^r e^{-\alpha\tau}\, d\tau\right) \right) \right\}$$

$$\delta_\epsilon = \hat{K} e^{-r\alpha} \left(1 + \|B\| \int_0^r e^{-\alpha\tau}\, d\tau\right) \|\phi\|\, e^{\epsilon \hat{K} e^{-r\alpha} t + \alpha t}$$

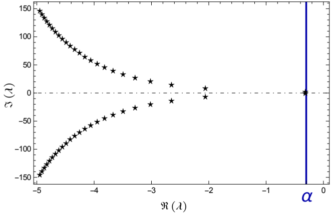
$$\epsilon \leq -\alpha / (2\hat{K} e^{-r\alpha})$$

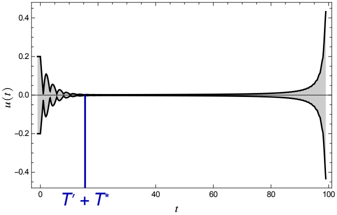
⇒ S. Feng, M. Chen, N. Zhan, M. Fränzle, B. Xue : *Taming delays in dyn. syst. : Unbounded verif. of DDEs.* CAV'19.

Reduction to Bounded Verification

[Population Dynamics, G. Hutchinson, 1948]

1. Identify the **rightmost eigenvalue** (and hence α), then construct K and δ.
2. Compute T^*, as well as T' (by bounded verifiers) s.t. $\|\Omega\| < \delta$ within T'.
3. Reduce to **bounded verifi.**, i.e., $\forall T > T' + T^*$, ∞-safe $\iff$ T-safe.



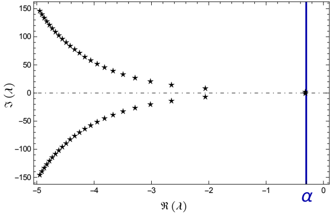


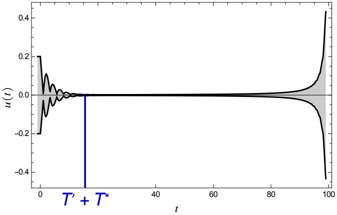
⇒ S. Feng, M. Chen, N. Zhan, M. Fränzle, B. Xue : *Taming delays in dyn. syst. : Unbounded verif. of DDEs*. CAV'19.

Reduction to Bounded Verification

[Population Dynamics, G. Hutchinson, 1948]

1. Identify the **rightmost eigenvalue** (and hence α), then construct K and δ.
2. Compute T^*, as well as T' (by bounded verifiers) s.t. $\|\Omega\| < \delta$ within T'.
3. Reduce to **bounded verifi.**, i.e., $\forall T > T' + T^*$, ∞-safe $\iff$ T-safe.





⇒ S. Feng, M. Chen, N. Zhan, M. Fränzle, B. Xue : *Taming delays in dyn. syst. : Unbounded verif. of DDEs*. CAV'19.

Non-Polynomial Dynamics : Disease Pathology

[M. C. Mackey and L. Glass, 1977]

$$\dot{p}(t) = \frac{\beta\theta^n p(t-r)}{\theta^n + p^n(t-r)} - \gamma p(t)$$

#mature blood cells in circulation   delay btw. cell production and maturation

Non-Polynomial Dynamics : Disease Pathology

[M. C. Mackey and L. Glass, 1977]

$$\dot{p}(t) = \frac{\beta\theta^n p(t-r)}{\theta^n + p^n(t-r)} - \gamma p(t)$$

#mature blood cells in circulation   delay btw. cell production and maturation

Parameters : $\theta = n = 1, \beta = 0.5, \gamma = 0.6, r = 0.5$.

∞-safety configuration : $\mathcal{X}_0 = [0, 0.2], \mathcal{U} = \{p \mid |p| > 0.3\}$.

Non-Polynomial Dynamics : Disease Pathology

[M. C. Mackey and L. Glass, 1977]

$$\dot{p}(t) = \frac{\beta\theta^n p(t-r)}{\theta^n + p^n(t-r)} - \gamma p(t)$$

#mature blood cells in circulation delay btw. cell production and maturation

Parameters : $\theta = n = 1, \beta = 0.5, \gamma = 0.6, r = 0.5$.

∞-safety configuration : $\mathcal{X}_0 = [0, 0.2], \mathcal{U} = \{p \mid |p| > 0.3\}$.

Linearization yields

$$\dot{p}(t) = -0.6p(t) + 0.5p(t - 0.5).$$

Critical values : $\alpha = -0.07, K = 1.75081, \delta = 0.0163426, T^* = 0$.

Non-Polynomial Dynamics : Disease Pathology

[M. C. Mackey and L. Glass, 1977]

$$\dot{p}(t) = \frac{\beta\theta^n p(t-r)}{\theta^n + p^n(t-r)} - \gamma p(t)$$

#mature blood cells in circulation   delay btw. cell production and maturation

Parameters : $\theta = n = 1, \beta = 0.5, \gamma = 0.6, r = 0.5$.

∞-safety configuration : $\mathcal{X}_0 = [0, 0.2], \mathcal{U} = \{p \mid |p| > 0.3\}$.

Linearization yields

$$\dot{p}(t) = -0.6p(t) + 0.5p(t - 0.5).$$

Critical values : $\alpha = -0.07, K = 1.75081, \delta = 0.0163426, T^* = 0$.

By bounded verification [E. Goubault et al., CAV'18], with Taylor models of the order 5 :

$$\left\| \Omega|_{[25.45, 25.95]} \right\| < \delta \quad \text{and} \quad \Omega|_{[-0.5, 25.95+0]} \cap \mathcal{U} = \emptyset.$$

Non-Polynomial Dynamics : Disease Pathology

[M. C. Mackey and L. Glass, 1977]

$$\underbrace{\dot{p}(t)}_{\text{\#mature blood cells in circulation}} = \frac{\beta\theta^n p(t-\underbrace{r}_{\text{delay btw. cell production and maturation}})}{\theta^n + p^n(t-r)} - \gamma p(t)$$

Parameters : $\theta = n = 1, \beta = 0.5, \gamma = 0.6, r = 0.5.$

∞-safety configuration : $\mathcal{X}_0 = [0, 0.2], \mathcal{U} = \{p \mid |p| > 0.3\}.$

Linearization yields

$$\dot{p}(t) = -0.6p(t) + 0.5p(t - 0.5).$$

Critical values : $\alpha = -0.07, K = 1.75081, \delta = 0.0163426, T^* = 0.$

By bounded verification [E. Goubault et al., CAV'18], with Taylor models of the order 5 :

$$\left\| \Omega|_{[25.45, 25.95]} \right\| < \delta \quad \text{and} \quad \Omega|_{[-0.5, 25.95+0]} \cap \mathcal{U} = \emptyset.$$

$$\Downarrow$$

∞-safety

Comparison with Existing Methods for Unbounded Verification

- ☺ Allow immediate feedback, i.e, $\mathrm{x}(t)$, as well as multiple delays in the dynamics, to which the technique in [L. Zou et al., CAV'15] does not generalize immediately.
- ☺ No polynomial template needs to be specified, yet necessarily for the *interval Taylor models* in [L. Zou et al., CAV'15] and [P. N. Mosaad et al., ICTAC'16], for *Lyapunov functionals* in [M. Peet and S. Lall, NOLCOS'04], or for *barrier certificates* in [S. Prajna and A. Jadbabaie, CDC'05].
- ☺ Delay-dependent stability certificate, other than the *absolute stability* exploited in [M. Peet and S. Lall, NOLCOS'04], i.e., a criterion requiring stability for arbitrarily large delays.
- ☹ Confined to differential dynamics featuring exponential stability. Investigation of more permissive forms of stability, e.g., asymptotical stability, that may admit a similar reduction-based idea, is subject to future work.

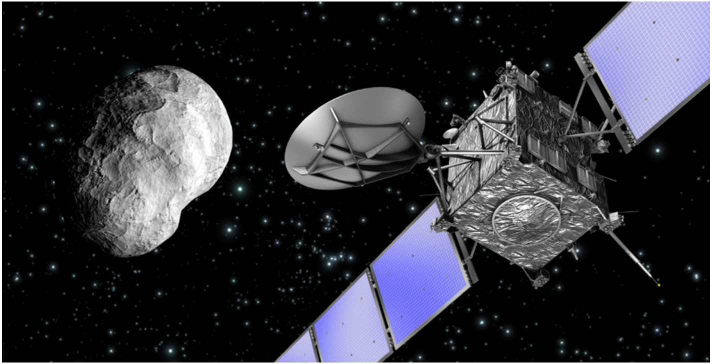
Synthesizing Safe Control Resilient to Delayed Interaction

Staying safe and reaching an objective
when observation & actuation are confined by delays

—Joint work w/ M. Fränzle, Y. Li, P. Mosaad, Y. Bai, T. Gan, L. Jiao, B. Xia, B. Xue—

Staying Safe

When Observation & Actuation Suffer from Serious Delays



©ESA

- You could move slowly. (Well, can you?)
- You could trust autonomy.
- Or you have to anticipate and issue actions early.

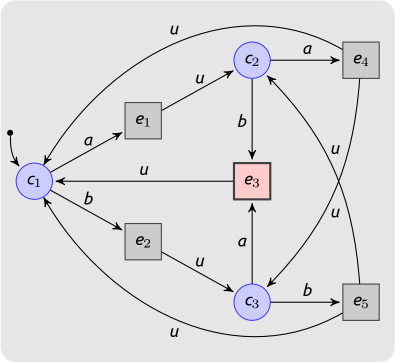
Synthesizing Delay-Resilient Control in Safety Games

2.1 Safety Games
Incremental Synthesis → Delay-Resilient Control
Diff. Delay Patterns → Equivalent Controllability

Invariance →

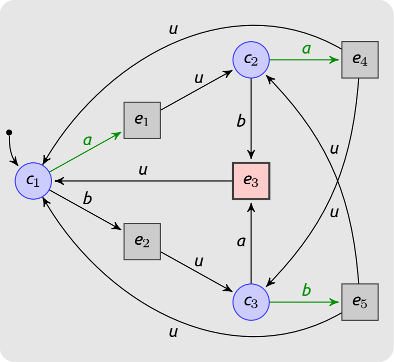
2.2 Delay Hybrid Automata
Fixed-Point Comput. → Invariant Generation
Global Invariants → Safe Switching Logic

A Trivial Safety Game



Goal : Avoid e_3 by appropriate actions of player c.

A Trivial Safety Game

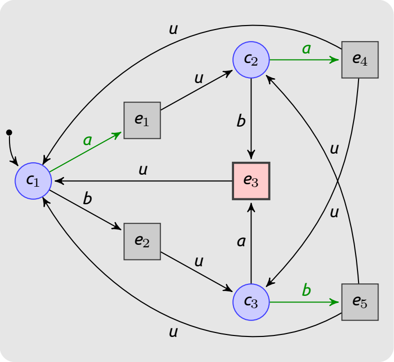


Goal : Avoid e_3 by appropriate actions of player c.

Strategy : May always play a except in c_3 :

$$c_1, c_2 \mapsto a$$
$$c_3 \mapsto b$$

A Trivial Safety Game



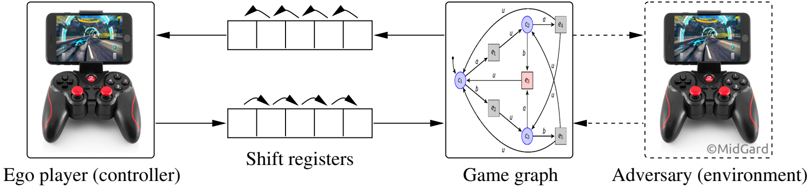
Goal : Avoid $\boxed{e_3}$ by appropriate actions of player c.

Strategy : May always play a except in c_3 :

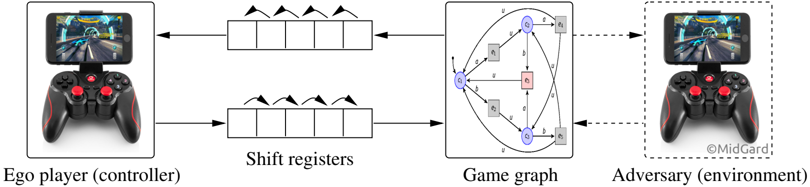
$$c_1, c_2 \mapsto a$$
$$c_3 \mapsto b$$

Properties : Determinacy and memoryless.

Playing Safety Games under Discrete Delay



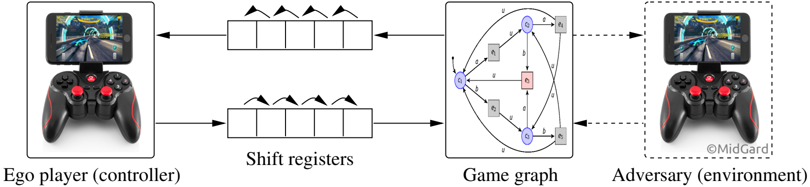
Playing Safety Games under Discrete Delay



Observation : It doesn't make an observable difference for the joint dynamics whether delay occurs in *perception*, *actuation*, or *both*.

1. In fact, two different ones : To mimic opacity of the shift registers, delay has to be moved to actuation/sensing for ego/adversary, resp. *The two thus play different games!*

Playing Safety Games under Discrete Delay



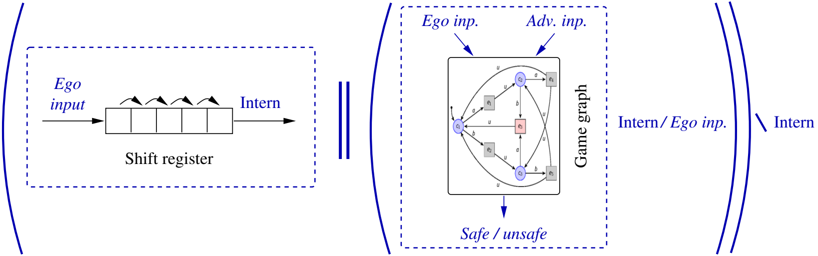
Observation : It doesn't make an observable difference for the joint dynamics whether delay occurs in *perception*, *actuation*, or *both*.

Consequence : An obvious *reduction* to a safety game of *perfect information*.

1. In fact, two different ones : To mimic opacity of the shift registers, delay has to be moved to actuation/sensing for ego/adversary, resp. *The two thus play different games!*

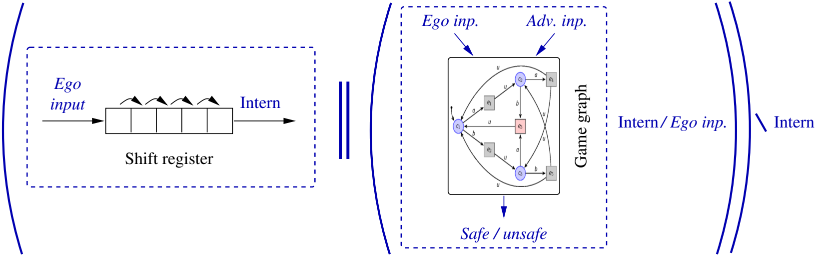
Reduction to Delay-Free Games

from Ego-Player Perspective



Reduction to Delay-Free Games

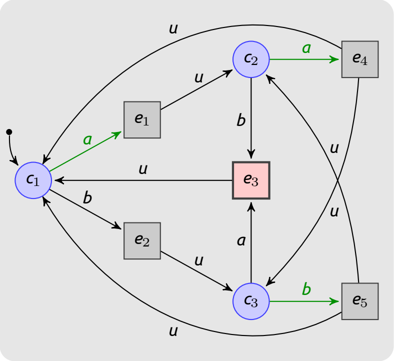
from Ego-Player Perspective



- ☺ Safety games under delay can be solved algorithmically.
- ☹ Game graph incurs blow-up by factor $|\text{Alphabet(ego)}|^{\text{delay}}$.

The Simple Safety Game

... but with Delay

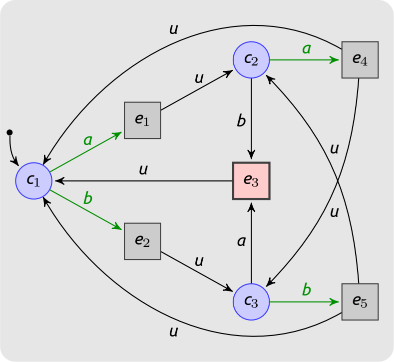


No delay :

$$\begin{aligned} c_1, c_2 &\mapsto a \\ c_3 &\mapsto b \end{aligned}$$

The Simple Safety Game

... but with Delay



No delay :

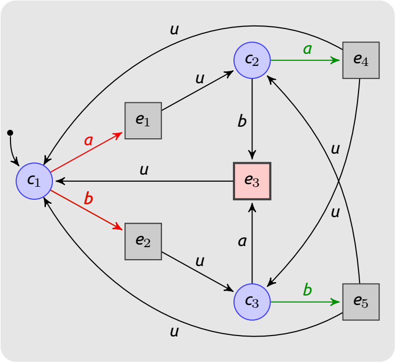
$$c_1, c_2 \mapsto a$$
$$c_3 \mapsto b$$

1 step delay :

$$e_1, e_5 \mapsto a$$
$$e_2, e_4 \mapsto b$$

The Simple Safety Game

... but with Delay



No delay :

$$\begin{aligned} c_1, c_2 &\mapsto a \\ c_3 &\mapsto b \end{aligned}$$

1 step delay :

$$\begin{aligned} e_1, e_5 &\mapsto a \\ e_2, e_4 &\mapsto b \end{aligned}$$

2 steps delay :

$$c_1 \mapsto \begin{cases} a & \text{if 2 steps back an } a \text{ was issued,} \\ b & \text{if 2 steps back a } b \text{ was issued.} \end{cases}$$

$$\begin{aligned} c_2 &\mapsto b \\ c_3 &\mapsto a \end{aligned}$$

Need memory!

Incremental Synthesis in a Nutshell

Observation : A winning strategy for delay $k' > k$ can always be utilized for a safe win under delay k.

Consequence : A position is winning for delay k is a necessary condition for it being winning under delay $k' > k$.

⇒ M. Chen, M. Fränzle, Y. Li, P. N. Mosaad, N. Zhan : *What's to come is still unsure : Synthesizing controllers resilient to delayed interaction*. ATVA '18. [Distinguished Paper Award].

Incremental Synthesis in a Nutshell

Observation : A winning strategy for delay $k' > k$ can always be utilized for a safe win under delay k.

Consequence : A position is winning for delay k is a necessary condition for it being winning under delay $k' > k$.

Idea : Incrementally filter out loss states &
incrementally synthesize winning strategy for the remaining :

1. synthesize winning strategy for the *delay-free* counterpart;
2. for each winning state, *lift strategy from delay k to k* $+ 1$;
3. *remove states* where this does not succeed;
4. repeat from 2 until either delay-resilience suffices (*winning*) or initial state turns lossy (*losing*).

⇒ M. Chen, M. Fränzle, Y. Li, P. N. Mosaad, N. Zhan : *What's to come is still unsure : Synthesizing controllers resilient to delayed interaction.* ATVA '18. [Distinguished Paper Award].

Incremental Synthesis in a Nutshell

Observation : A winning strategy for delay $k' > k$ can always be utilized for a safe win under delay k.

Consequence : A position is winning for delay k is a necessary condition for it being winning under delay $k' > k$.

Idea : Incrementally filter out loss states &
incrementally synthesize winning strategy for the remaining :

1. synthesize winning strategy for the *delay-free* counterpart;
2. for each winning state, *lift strategy from delay k to $k + 1$*;
3. *remove states* where this does not succeed;
4. repeat from 2 until either delay-resilience suffices (*winning*) or initial state turns lossy (*losing*).

⇒ M. Chen, M. Fränzle, Y. Li, P. N. Mosaad, N. Zhan : *What's to come is still unsure : Synthesizing controllers resilient to delayed interaction.* ATVA '18. [Distinguished Paper Award].

Incremental Synthesis in a Nutshell

Observation : A winning strategy for delay $k' > k$ can always be utilized for a safe win under delay k.

Consequence : A position is winning for delay k is a necessary condition for it being winning under delay $k' > k$.

Idea : Incrementally filter out loss states &
incrementally synthesize winning strategy for the remaining :

1. synthesize winning strategy for the *delay-free* counterpart;
2. for each winning state, *lift strategy from delay k to k* $+ 1$;
3. *remove states* where this does not succeed;
4. repeat from 2 until either delay-resilience suffices (*winning*) or initial state turns lossy (*losing*).

⇒ M. Chen, M. Fränzle, Y. Li, P. N. Mosaad, N. Zhan : *What's to come is still unsure : Synthesizing controllers resilient to delayed interaction.* ATVA '18. [Distinguished Paper Award].

Incremental Synthesis in a Nutshell

Observation : A winning strategy for delay $k' > k$ can always be utilized for a safe win under delay k.

Consequence : A position is winning for delay k is a necessary condition for it being winning under delay $k' > k$.

Idea : Incrementally filter out loss states &
incrementally synthesize winning strategy for the remaining :

1. synthesize winning strategy for the *delay-free* counterpart;
2. for each winning state, *lift strategy from delay k to k* $+ 1$;
3. *remove states* where this does not succeed;
4. repeat from 2 until either delay-resilience suffices (*winning*) or initial state turns lossy (*losing*).

⇒ M. Chen, M. Fränzle, Y. Li, P. N. Mosaad, N. Zhan : *What's to come is still unsure : Synthesizing controllers resilient to delayed interaction*. ATVA '18. [Distinguished Paper Award].

Incremental Synthesis in a Nutshell

Observation : A winning strategy for delay $k' > k$ can always be utilized for a safe win under delay k.

Consequence : A position is winning for delay k is a necessary condition for it being winning under delay $k' > k$.

Idea : Incrementally filter out loss states &
incrementally synthesize winning strategy for the remaining :

1. synthesize winning strategy for the *delay-free* counterpart ;
2. for each winning state, *lift strategy from delay k to k* $+1$;
3. *remove states* where this does not succeed ;
4. repeat from 2 until either delay-resilience suffices (*winning*) or initial state turns lossy (*losing*).

⇒ M. Chen, M. Fränzle, Y. Li, P. N. Mosaad, N. Zhan : *What's to come is still unsure : Synthesizing controllers resilient to delayed interaction.* ATVA '18. [Distinguished Paper Award].

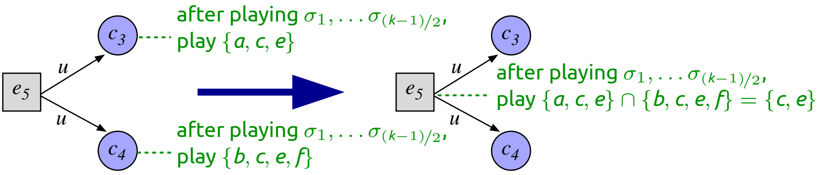
Incremental Synthesis of Delay-Tolerant Strategies

1. Generate a *maximally permissive* strategy for delay $k = 0$.

Incremental Synthesis of Delay-Tolerant Strategies

1. Generate a *maximally permissive* strategy for delay $k = 0$.
2. Advance to delay $k + 1$:

   If k odd : For each (ego-)winning adversarial state define strategy as

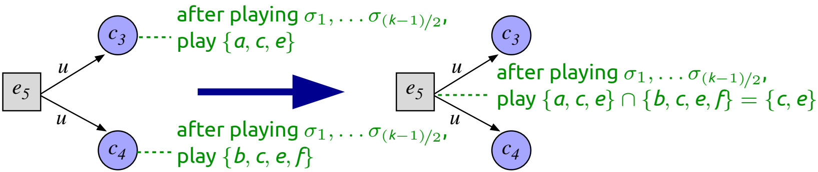


... and eliminate any dead ends by bwd. traversal.

Incremental Synthesis of Delay-Tolerant Strategies

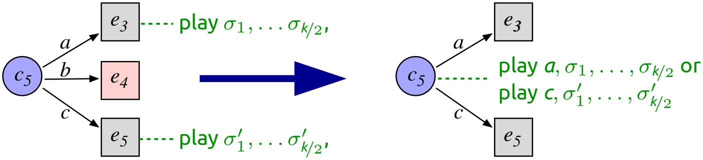
1. Generate a *maximally permissive* strategy for delay $k = 0$.
2. Advance to delay $k + 1$:

   If k odd : For each (ego-)winning adversarial state define strategy as



   … and eliminate any dead ends by bwd. traversal.

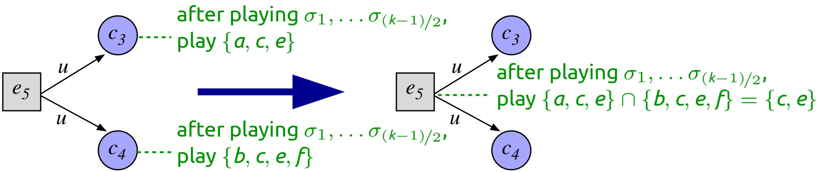
   If k even : For each winning ego state define strategy as



Incremental Synthesis of Delay-Tolerant Strategies

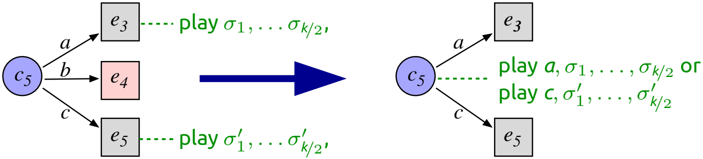
1. Generate a *maximally permissive* strategy for delay $k = 0$.
2. Advance to delay $k + 1$:

   If k odd : For each (ego-)winning adversarial state define strategy as



   … and eliminate any dead ends by bwd. traversal.

   If k even : For each winning ego state define strategy as



3. Repeat from 2 until either delay-resilience suffices or initial state turns lossy.

Incremental- vs. Reduction-Based Synthesis

Benchmark				Reduction + Explicit-State Synthesis						Incremental Explicit-State Synthesis						
name	$\|S\|$	$\|\to\|$	$\|\mathcal{U}\|$	$\delta_{\max}$	$\delta=0$	$\delta=1$	$\delta=2$	$\delta=3$	$\delta=4$	$\delta_{\max}$	$\delta=0$	$\delta=1$	$\delta=2$	$\delta=3$	$\delta=4$	%
`Exmp.trv1`	14	20	4	≥ 22	0.00	0.00	0.01	0.02	0.02	$\boldsymbol{\geq 30}$	0.00	0.00	**0.00**	**0.01**	**0.01**	
`Exmp.trv2`	14	22	4	$=2$	0.00	0.01	0.01	0.02	–	$=2$	0.00	**0.00**	**0.00**	**0.01**	–	81.97
`Escp.4×4`	224	738	16	$=2$	0.08	11.66	11.73	1059.23	–	$=2$	0.08	**0.13**	**0.22**	**0.25**	–	99.02
`Escp.4×5`	360	1326	20	$=2$	0.18	34.09	33.80	3084.58	–	$=2$	0.18	**0.27**	**0.46**	**0.63**	–	99.02
`Escp.5×5`	598	2301	26	≥ 2	0.46	96.24	97.10	?	?	$\boldsymbol{=2}$	0.46	**0.68**	**1.16**	**1.71**	–	98.98
`Escp.5×6`	840	3516	30	≥ 2	1.01	217.63	216.83	?	?	$\boldsymbol{=2}$	**1.00**	**1.42**	**2.40**	**4.30**	–	99.00
`Escp.6×6`	1224	5424	36	≥ 2	2.13	516.92	511.41	?	?	$\boldsymbol{=2}$	**2.06**	**2.90**	**5.12**	**10.30**	–	98.97
`Escp.7×7`	2350	11097	50	≥ 2	7.81	2167.86	2183.01	?	?	$\boldsymbol{=2}$	**7.71**	**10.67**	**19.04**	**52.47**	–	98.99
`Escp.7×8`	3024	14820	56	≥ 0	**13.07**	?	?	?	?	$\boldsymbol{=2}$	13.44	**18.25**	**32.69**	**108.60**	–	99.01

Benchmark		Reduction + Yosys + SafetySynth (symbolic)							Incremental Synthesis (explicit-state implementation)							
name	$\delta_{\max}$	$\delta=0$	$\delta=1$	$\delta=2$	$\delta=3$	$\delta=4$	$\delta=5$	$\delta=6$	$\delta=0$	$\delta=1$	$\delta=2$	$\delta=3$	$\delta=4$	$\delta=5$	$\delta=6$	%
`Stub.4×4`	$=2$	1.07	1.24	1.24	1.80	–	–	–	**0.04**	**0.07**	**0.12**	**0.18**	–	–	–	98.98
`Stub.4×5`	$=2$	1.16	1.49	1.49	2.83	–	–	–	**0.08**	**0.14**	**0.25**	**0.44**	–	–	–	98.97
`Stub.5×5`	$=2$	1.19	2.61	2.50	13.67	–	–	–	**0.21**	**0.37**	**0.63**	**1.17**	–	–	–	98.97
`Stub.5×6`	$=2$	1.18	2.60	2.59	23.30	–	–	–	**0.42**	**0.69**	**1.20**	**2.49**	–	–	–	98.96
`Stub.6×6`	$=4$	1.17	2.76	2.74	19.96	19.69	655.24	–	**0.93**	**1.47**	**2.60**	**5.79**	**7.54**	**7.60**	–	99.89
`Stub.7×7`	$=4$	**1.23**	**2.50**	**2.48**	24.57	**23.01**	2224.62	–	3.60	5.52	10.08	**22.75**	31.18	**32.98**	–	99.88

Incremental- vs. Reduction-Based Synthesis

Benchmark				Reduction + Explicit-State Synthesis						Incremental Explicit-State Synthesis						
name	$\|S\|$	$\|\to\|$	$\|\mathcal{U}\|$	$\delta_{\max}$	$\delta=0$	$\delta=1$	$\delta=2$	$\delta=3$	$\delta=4$	$\delta_{\max}$	$\delta=0$	$\delta=1$	$\delta=2$	$\delta=3$	$\delta=4$	%
`Exmp.trv1`	14	20	4	≥ 22	0.00	0.00	0.01	0.02	0.02	**$\geq$ 30**	0.00	0.00	**0.00**	**0.01**	**0.01**	
`Exmp.trv2`	14	22	4	$=2$	0.00	0.01	0.01	0.02	–	$=2$	0.00	**0.00**	**0.00**	**0.01**	–	81.97
`Escp.4×4`	224	738	16	$=2$	0.08	11.66	11.73	1059.23	–	$=2$	0.08	**0.13**	**0.22**	**0.25**	–	99.02
`Escp.4×5`	360	1326	20	$=2$	0.18	34.09	33.80	3084.58	–	$=2$	0.18	**0.27**	**0.46**	**0.63**	–	99.02
`Escp.5×5`	598	2301	26	≥ 2	0.46	96.24	97.10	?	?	**= 2**	0.46	**0.68**	**1.16**	**1.71**	–	98.98
`Escp.5×6`	840	3516	30	≥ 2	1.01	217.63	216.83	?	?	**= 2**	**1.00**	**1.42**	**2.40**	**4.30**	–	99.00
`Escp.6×6`	1224	5424	36	≥ 2	2.13	516.92	511.41	?	?	**= 2**	**2.06**	**2.90**	**5.12**	**10.30**	–	98.97
`Escp.7×7`	2350	11097	50	≥ 2	7.81	2167.86	2183.01	?	?	**= 2**	**7.71**	**10.67**	**19.04**	**52.47**	–	98.99
`Escp.7×8`	3024	14820	56	≥ 0	**13.07**	?	?	?	?	**= 2**	13.44	**18.25**	**32.69**	**108.60**	–	99.01

Benchmark		Reduction + Yosys + SafetySynth (symbolic)							Incremental Synthesis (explicit-state implementation)							
name	$\delta_{\max}$	$\delta=0$	$\delta=1$	$\delta=2$	$\delta=3$	$\delta=4$	$\delta=5$	$\delta=6$	$\delta=0$	$\delta=1$	$\delta=2$	$\delta=3$	$\delta=4$	$\delta=5$	$\delta=6$	%
`Stub.4×4`	$=2$	1.07	1.24	1.24	1.80	–	–	–	**0.04**	**0.07**	**0.12**	**0.18**	–	–	–	98.98
`Stub.4×5`	$=2$	1.16	1.49	1.49	2.83	–	–	–	**0.08**	**0.14**	**0.25**	**0.44**	–	–	–	98.97
`Stub.5×5`	$=2$	1.19	2.61	2.50	13.67	–	–	–	**0.21**	**0.37**	**0.63**	**1.17**	–	–	–	98.97
`Stub.5×6`	$=2$	1.18	2.60	2.59	23.30	–	–	–	**0.42**	**0.69**	**1.20**	**2.49**	–	–	–	98.96
`Stub.6×6`	$=4$	1.17	2.76	2.74	19.96	19.69	655.24	–	**0.93**	**1.47**	**2.60**	**5.79**	**7.54**	**7.60**	–	99.89
`Stub.7×7`	$=4$	**1.23**	**2.50**	**2.48**	24.57	**23.01**	2224.62	–	3.60	5.52	10.08	**22.75**	31.18	**32.98**	–	99.88

Incremental- vs. Reduction-Based Synthesis

Benchmark				Reduction + Explicit-State Synthesis						Incremental Explicit-State Synthesis						
name	$\|S\|$	$\|\to\|$	$\|\mathcal{U}\|$	$\delta_{\max}$	$\delta=0$	$\delta=1$	$\delta=2$	$\delta=3$	$\delta=4$	$\delta_{\max}$	$\delta=0$	$\delta=1$	$\delta=2$	$\delta=3$	$\delta=4$	%
`Exmp.trv1`	14	20	4	≥ 22	0.00	0.00	0.01	0.02	0.02	**$\geq$ 30**	0.00	0.00	**0.00**	**0.01**	**0.01**	
`Exmp.trv2`	14	22	4	$= 2$	0.00	0.01	0.01	0.02	–	$= 2$	0.00	**0.00**	**0.00**	**0.01**	–	81.97
`Escp.4×4`	224	738	16	$= 2$	0.08	11.66	11.73	1059.23	–	$= 2$	0.08	**0.13**	**0.22**	**0.25**	–	99.02
`Escp.4×5`	360	1326	20	$= 2$	0.18	34.09	33.80	3084.58	–	$= 2$	0.18	**0.27**	**0.46**	**0.63**	–	99.02
`Escp.5×5`	598	2301	26	≥ 2	0.46	96.24	97.10	?	?	**= 2**	0.46	**0.68**	**1.16**	**1.71**	–	98.98
`Escp.5×6`	840	3516	30	≥ 2	1.01	217.63	216.83	?	?	**= 2**	**1.00**	**1.42**	**2.40**	**4.30**	–	99.00
`Escp.6×6`	1224	5424	36	≥ 2	2.13	516.92	511.41	?	?	**= 2**	**2.06**	**2.90**	**5.12**	**10.30**	–	98.97
`Escp.7×7`	2350	11097	50	≥ 2	7.81	2167.86	2183.01	?	?	**= 2**	**7.71**	**10.67**	**19.04**	**52.47**	–	98.99
`Escp.7×8`	3024	14820	56	≥ 0	**13.07**	?	?	?	?	**= 2**	13.44	**18.25**	**32.69**	**108.60**	–	99.01

Benchmark		Reduction + Yosys + SafetySynth (symbolic)							Incremental Synthesis (explicit-state implementation)							
name	$\delta_{\max}$	$\delta=0$	$\delta=1$	$\delta=2$	$\delta=3$	$\delta=4$	$\delta=5$	$\delta=6$	$\delta=0$	$\delta=1$	$\delta=2$	$\delta=3$	$\delta=4$	$\delta=5$	$\delta=6$	%
`Stub.4×4`	$= 2$	1.07	1.24	1.24	1.80	–	–	–	**0.04**	**0.07**	**0.12**	**0.18**	–	–	–	98.98
`Stub.4×5`	$= 2$	1.16	1.49	1.49	2.83	–	–	–	**0.08**	**0.14**	**0.25**	**0.44**	–	–	–	98.97
`Stub.5×5`	$= 2$	1.19	2.61	2.50	13.67	–	–	–	**0.21**	**0.37**	**0.63**	**1.17**	–	–	–	98.97
`Stub.5×6`	$= 2$	1.18	2.60	2.59	23.30	–	–	–	**0.42**	**0.69**	**1.20**	**2.49**	–	–	–	98.96
`Stub.6×6`	$= 4$	1.17	2.76	2.74	19.96	19.69	655.24	–	**0.93**	**1.47**	**2.60**	**5.79**	**7.54**	**7.60**	–	99.89
`Stub.7×7`	$= 4$	**1.23**	**2.50**	**2.48**	24.57	**23.01**	2224.62	–	3.60	5.52	10.08	**22.75**	31.18	**32.98**	–	99.88

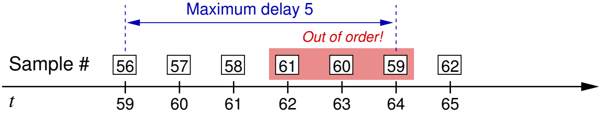
Incremental- vs. Reduction-Based Synthesis

Benchmark				Reduction + Explicit-State Synthesis						Incremental Explicit-State Synthesis						
name	$\|S\|$	$\|\rightarrow\|$	$\|\mathcal{U}\|$	$\delta_{\max}$	$\delta=0$	$\delta=1$	$\delta=2$	$\delta=3$	$\delta=4$	$\delta_{\max}$	$\delta=0$	$\delta=1$	$\delta=2$	$\delta=3$	$\delta=4$	%
`Exmp.trv1`	14	20	4	≥ 22	0.00	0.00	0.01	0.02	0.02	**$\geq$ 30**	0.00	0.00	**0.00**	**0.01**	**0.01**	
`Exmp.trv2`	14	22	4	$= 2$	0.00	0.01	0.01	0.02	–	$= 2$	0.00	**0.00**	**0.00**	**0.01**	–	81.97
`Escp.4×4`	224	738	16	$= 2$	0.08	11.66	11.73	1059.23	–	$= 2$	0.08	**0.13**	**0.22**	**0.25**	–	99.02
`Escp.4×5`	360	1326	20	$= 2$	0.18	34.09	33.80	3084.58	–	$= 2$	0.18	**0.27**	**0.46**	**0.63**	–	99.02
`Escp.5×5`	598	2301	26	≥ 2	0.46	96.24	97.10	?	?	**= 2**	0.46	**0.68**	**1.16**	**1.71**	–	98.98
`Escp.5×6`	840	3516	30	≥ 2	1.01	217.63	216.83	?	?	**= 2**	**1.00**	**1.42**	**2.40**	**4.30**	–	99.00
`Escp.6×6`	1224	5424	36	≥ 2	2.13	516.92	511.41	?	?	**= 2**	**2.06**	**2.90**	**5.12**	**10.30**	–	98.97
`Escp.7×7`	2350	11097	50	≥ 2	7.81	2167.86	2183.01	?	?	**= 2**	**7.71**	**10.67**	**19.04**	**52.47**	–	98.99
`Escp.7×8`	3024	14820	56	≥ 0	**13.07**	?	?	?	?	**= 2**	13.44	**18.25**	**32.69**	**108.60**	–	99.01

Benchmark		Reduction + Yosys + SafetySynth (symbolic)							Incremental Synthesis (explicit-state implementation)							
name	$\delta_{\max}$	$\delta=0$	$\delta=1$	$\delta=2$	$\delta=3$	$\delta=4$	$\delta=5$	$\delta=6$	$\delta=0$	$\delta=1$	$\delta=2$	$\delta=3$	$\delta=4$	$\delta=5$	$\delta=6$	%
`Stub.4×4`	$= 2$	1.07	1.24	1.24	1.80	–	–	–	**0.04**	**0.07**	**0.12**	**0.18**	–	–	–	98.98
`Stub.4×5`	$= 2$	1.16	1.49	1.49	2.83	–	–	–	**0.08**	**0.14**	**0.25**	**0.44**	–	–	–	98.97
`Stub.5×5`	$= 2$	1.19	2.61	2.50	13.67	–	–	–	**0.21**	**0.37**	**0.63**	**1.17**	–	–	–	98.97
`Stub.5×6`	$= 2$	1.18	2.60	2.59	23.30	–	–	–	**0.42**	**0.69**	**1.20**	**2.49**	–	–	–	98.96
`Stub.6×6`	$= 4$	1.17	2.76	2.74	19.96	19.69	655.24	–	**0.93**	**1.47**	**2.60**	**5.79**	**7.54**	**7.60**	–	99.89
`Stub.7×7`	$= 4$	**1.23**	**2.50**	**2.48**	24.57	**23.01**	2224.62	–	3.60	5.52	10.08	**22.75**	31.18	**32.98**	–	99.88

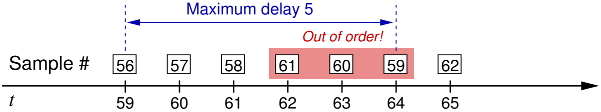
Out-of-Order Message Delivery

☹ Observations may arrive *out-of-order* :

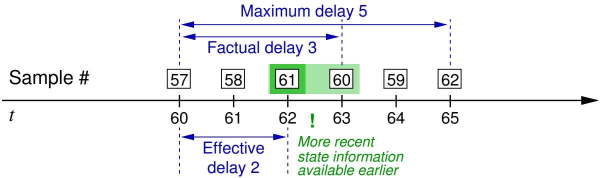


Out-of-Order Message Delivery

☹ Observations may arrive *out-of-order* :

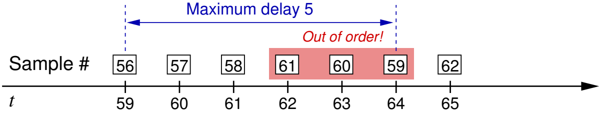


☺ But this may only reduce effective delay, improving controllability :

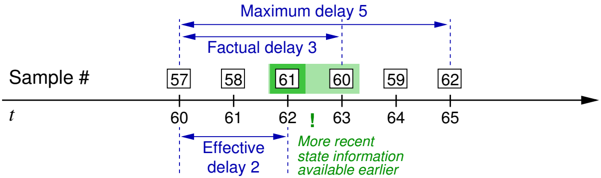


Out-of-Order Message Delivery

☹ Observations may arrive *out-of-order* :



☺ But this may only reduce effective delay, improving controllability :

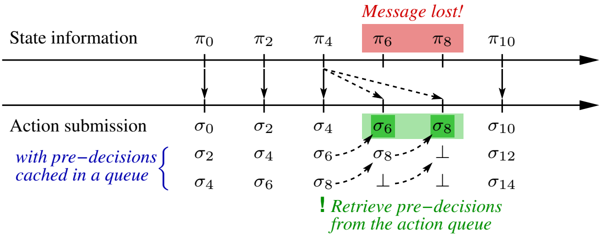


☺ W.r.t. qualitative controllability, the worst-case of out-of-order delivery is equivalent to order-preserving delay k.

☺ Stochastically expected controllability even better than for strict delay k.

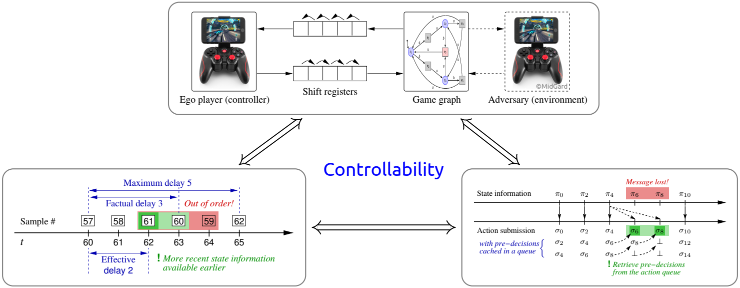
(Bounded) Message Loss

☹ Message carrying the state information may get *lost* :



☺ The controller can still win a safety game in the presence of bounded message loss leveraging delay-resilient strategies.

Equivalence of Qualitative Controllability



⇒ M. Chen, M. Fränzle, Y. Li, P. N. Mosaad, N. Zhan : *Indecision and delays are the parents of failure : Taming them algorithmically by synthesizing delay-resilient control*. Acta Informatica '21.

Synthesizing Safe Switching Logic for Hybrid Systems

2.1 Safety Games
Incremental Synthesis → Delay-Resilient Control
Diff. Delay Patterns → Equivalent Controllability

Invariance →

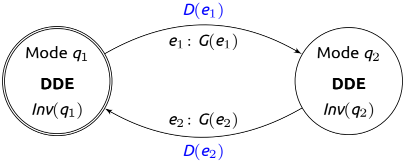
2.2 Delay Hybrid Automata
Fixed-Point Comput. → Invariant Generation
Global Invariants → Safe Switching Logic

Delay Hybrid Automata

Definition (Delay Hybrid Automaton, DHA)

A DHA is a tuple $\mathcal{H} \mathrel{\widehat{=}} (Q, X, U, Inv, X_0, F, E, D, G, R)$ where

- U: a set of continuous functionals,
- Inv: an invariant $Inv(q)$ for each mode $q \in Q$,
- R: $E \times X_D \to U$: reset functions,
- ...



Switching-Logic Synthesis Problem

Given : A DHA $\mathcal{H} = (Q, X, U, Inv, X_0, F, E, D, G, R)$ and a safety property P;

Goal : A new DHA $\mathcal{H}^\star = (Q, X, U^\star, Inv^\star, X_0^\star, F, E, D, G^\star, R)$ such that

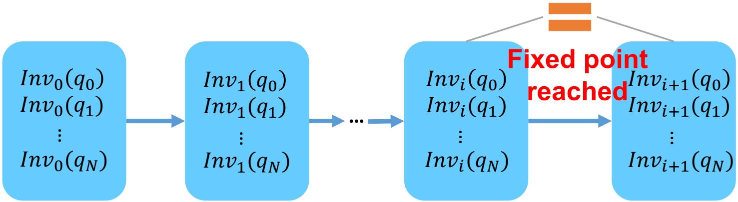
- $\mathcal{H}^\star$ is *safe* w.r.t. P,
- $\mathcal{H}^\star$ is a *refinement* of $\mathcal{H}$,
- $\mathcal{H}^\star$ is *non-blocking*.

⇒ Y. Bai, T. Gan, L. Jiao, B. Xia, B. Xue, N. Zhan : *Switching controller synthesis for time-delayed hybrid systems (under perturbation)*. HSCC'21 (Sci. China Math. '21).

Invariant Generation

Generate a global invariant for $\mathcal{H}$ by computing a fixed point :

1. generate a *strengthened differential invariant* for each mode,
2. generate a *strengthened guard* for each transition.



Generating Differential Invariants

$$\text{Linear DDE}: \quad \dot{\mathrm{x}}(t) = A\mathrm{x}(t) + B\mathrm{x}(t-r) + C\mathrm{w}(t)$$

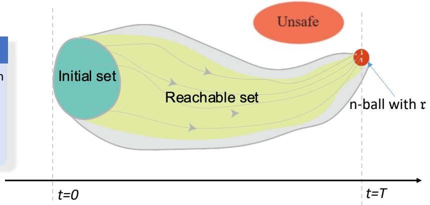
1. reduce to *T-invariant*, i.e., $\forall T > T^*$, ∞-invariant $\Longleftrightarrow$ T-invariant,
2. compute a *safe over-approximation* of the reachable set within T.

Exponentially convergent to a ball:

if there exist constant $\gamma > 0$ and non-decreasing function $\kappa(\cdot)$ s.t.

$$\left\|\xi_\phi^w(t)\right\|_\infty \le r + \kappa(\|\phi\|_\infty)e^{-\gamma t}, \qquad \forall t \ge 0$$

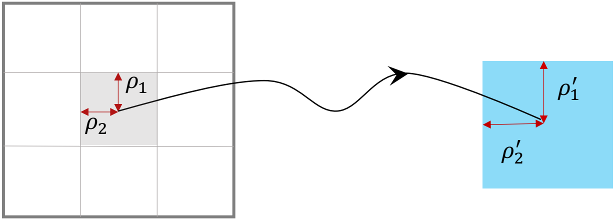
holds for all $\phi \in C$, $\|w(t)\|_\infty \le \bar{w}$, $\forall t \ge 0$.



Generating Differential Invariants

$$\text{Linear DDE}: \ \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{x}(t-r) + C\mathbf{w}(t)$$

1. reduce to *T-invariant*, i.e., $\forall T > T^*$, ∞-invariant $\iff$ T-invariant,
2. compute a *safe over-approximation* of the reachable set within T.



Generating Differential Invariants

$$\text{Nonlinear DDE}: \quad \dot{\mathrm{x}}(t) = \boldsymbol{f}(\mathrm{x}(t), \mathrm{x}(t-r), \mathrm{w}(t))$$

$$\Big\Downarrow \ \text{linearization}$$

$$\text{Linear DDE}: \quad \dot{\mathrm{x}}(t) = A\mathrm{x}(t) + B\mathrm{x}(t-r) + C\mathrm{w}(t) + \boldsymbol{g}(\mathrm{x}(t), \mathrm{x}(t-r))$$

Generating Differential Invariants

$$\text{Nonlinear DDE}: \ \dot{\mathrm{x}}(t) = \boldsymbol{f}(\mathrm{x}(t), \mathrm{x}(t-r), \mathrm{w}(t))$$

$$\Big\Downarrow \ \text{linearization}$$

$$\text{Linear DDE}: \ \dot{\mathrm{x}}(t) = A\mathrm{x}(t) + B\mathrm{x}(t-r) + C\mathrm{w}(t) + \boldsymbol{g}(\mathrm{x}(t), \mathrm{x}(t-r))$$

Reduce to *T-invariant*, i.e., $\forall T > T^*$, ∞-invariant $\Longleftrightarrow$ T-invariant.

Locally exponentially convergent to a ball:

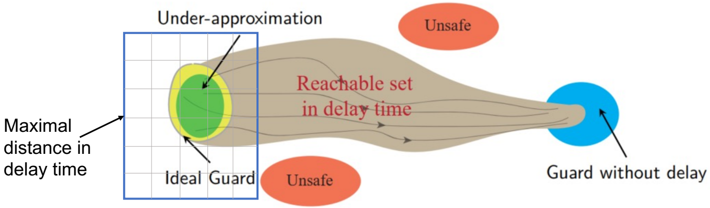
if there exist constants $\gamma > 0, l > 0$ and non-decreasing function $\kappa(\cdot)$ s.t.

$$\|\phi\ (t)\|_\infty \le l \ \Rightarrow \left\|\xi_\phi^w\ (t)\right\|_\infty \ \le \ r + \kappa(\|\phi\|_\infty)e^{-\gamma t}, \qquad \forall\, t \ge \ 0$$

holds for all $\phi \in C$, $\|w(t)\|_\infty \le \bar{w}$, $\forall\, t \ge \ 0$.

Generating Guard Conditions

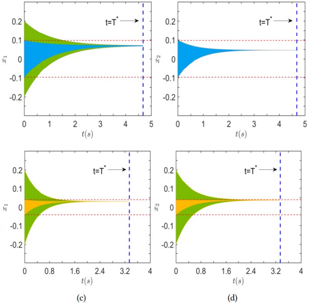
1. generate guards without delay via *invariants*,
2. generate guards under delay by *backward reachable-set computation*.



Example : Predator-Prey Populations

$$q_1 : \begin{cases} \begin{cases} \dot{x}_1(t) = -x_1(t)(1 - \frac{x_1(t)}{100}) + 0.2d_1 + w_{11}(t) \\ \dot{x}_2(t) = -1.5x_2(t)(1 - \frac{x_2(t)}{100}) + 0.1d_2 + w_{12}(t) \end{cases} \\ \Xi(q_1) = [-0.2, 0.2] \times [-0.1, 0.1] \\ I(q_1) = \mathbb{R}^2. \end{cases}$$

$$q_2 : \begin{cases} \begin{cases} \dot{x}_1(t) = -2.5x_1(t) + 0.2x_1(t - 0.01)(1 + x_2(t)) + w_{21}(t) \\ \dot{x}_2(t) = -2x_2(t) + 0.15x_2(t - 0.01)(1 + x_2(t)) + w_{22}(t) \end{cases} \\ \Xi(q_2) = [-0.2, 0.2] \times [-0.2, 0.2] \\ I(q_2) = \mathbb{R}^2. \end{cases}$$



Concluding Remarks

Problem : We face

- increasingly wide-spread use of networked distributed sensing and control,
- substantial feedback delays thus affecting hybrid control schemes,
- delays impact controllability and control performance in both the discrete and the continuous parts.

Status : We present

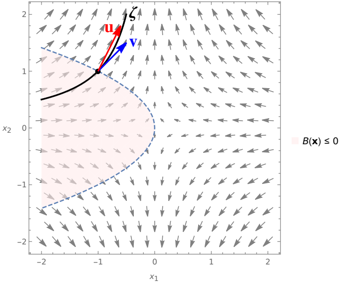
- bounded safety verification methods for delayed differential dynamics,
- extension to unbounded verification by leveraging stability criteria,
- safety games under delays and incremental algorithms for efficient control synthesis,
- delay hybrid automata and algorithms for switching-logic synthesis.

Future Work : We'd explore

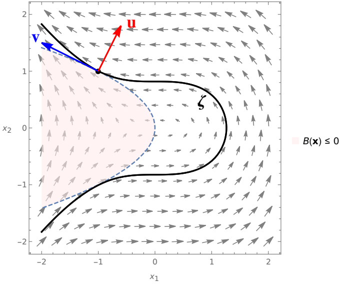
- DDE exhibiting state-dependent and/or stochastic delay,
- invariant generation for time-delayed systems (on-going) :
  - initial attempts : [Prajna & Jadbabaie : CDC'05], [Ames *et al.* : ACC'19, ACC'21], [Liu *et al.* : SCIS'21],
  - but general invariant generation for DDEs remains challenging.


HURRAY
FOR DELAY!
Brussels Dichterscollectief
Le Collectif de Poètes Bruxellois
Brussels Poetry Collective
2011
27.03

Lie Derivatives and Trajectory Tendency



(a) first-order Lie derivative and the gradient



(b) demand for the second-order Lie derivative

Figure 6: An illustration of how Lie derivatives capture the tendency of trajectories in terms of a polynomial function $B(\mathbf{x})$. ζ: the system trajectory passing through $(-1, 1)$; $\mathbf{v}$: the evolution direction per the vector field at $(-1, 1)$; $\mathbf{u}$: the gradient of $B(\mathbf{x})$ at $(-1, 1)$.

Equivalence of Qualitative Controllability

Theorem (Equivalence of qualitative controllability)

Given a two-player safety game, the following statements are equivalent if δ is even :

1. *There exists a winning strategy under an exact delay of δ, i.e., if at any point of time t the control strategy is computed based on a prefix of the game that has length $t - \delta$.*
2. *There exists a winning strategy under time-stamped out-of-order delivery with a maximum delay of δ, i.e., if at any point of time t the control strategy is computed based on the complete prefix of the game of length $t - \delta$ plus potentially available partial knowledge of the game states between $t - \delta$ and t.*
3. *There exists a winning strategy when at any time $t = 2n$, i.e., any player-0 move, information on the game state at some time $t' \in \{t - 2k, \ldots, t\}$ is available, i.e., under out-of-order delivery of messages with a maximum delay of δ and a maximum number of consecutively lost upstream or downstream messages of $\delta/2$.*

The first two equivalences do also hold for odd δ.

⇒ M. Chen, M. Fränzle, Y. Li, P. N. Mosaad, N. Zhan : *Indecision and delays are the parents of failure : Taming them algorithmically by synthesizing delay-resilient control.* Acta Informatica '20.