Verification & Synthesis of Time-Delayed Dynamics

[Doctoral Dissertation Defence]

Mingshuai Chen

✉ chenms@ios.ac.cn 🌍 lcs.ios.ac.cn/~chenms/

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

Beijing · May 2019

ISCAS

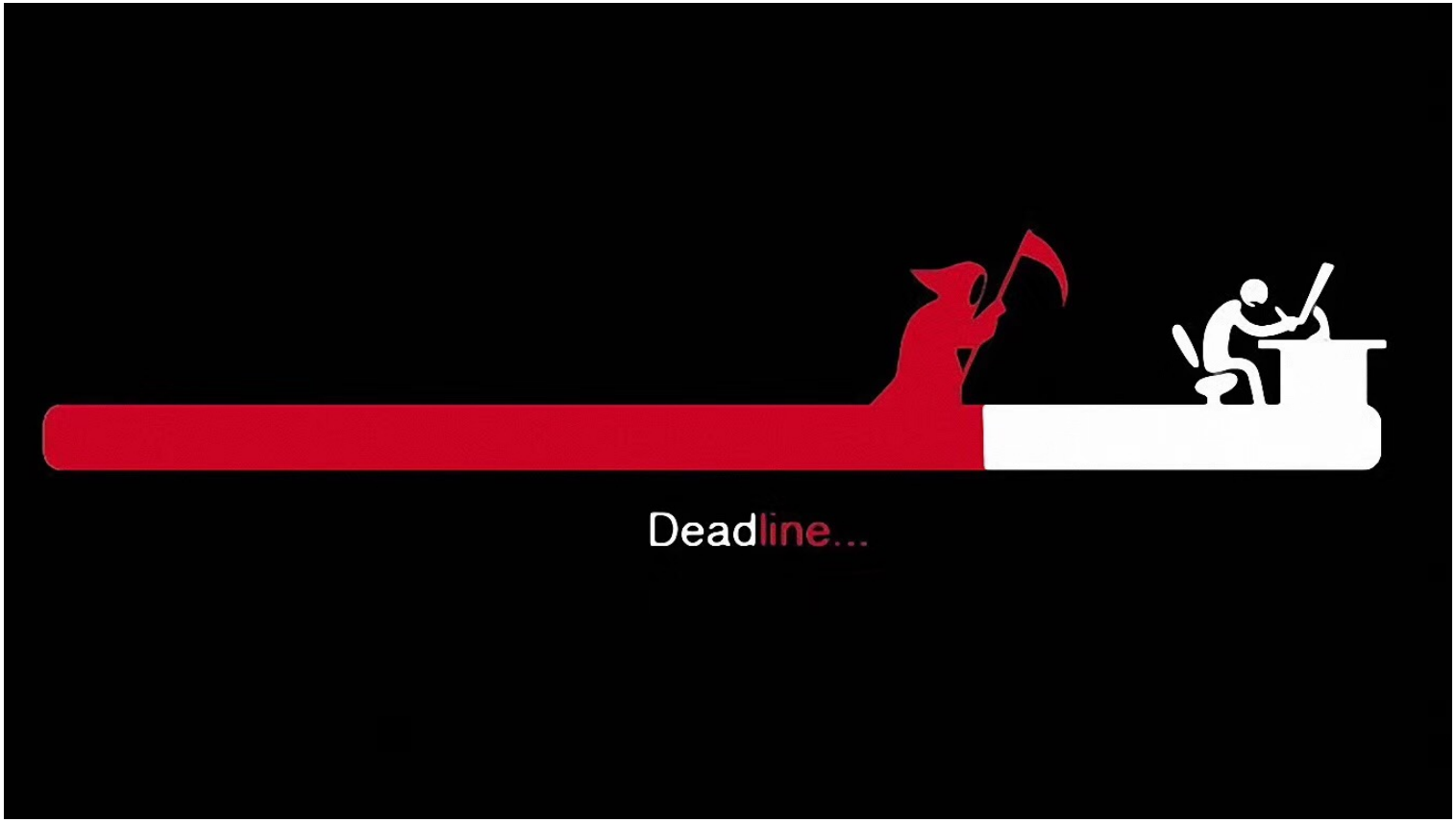
Every Time Being Asked to Give a Self-Intro. …

About Me …

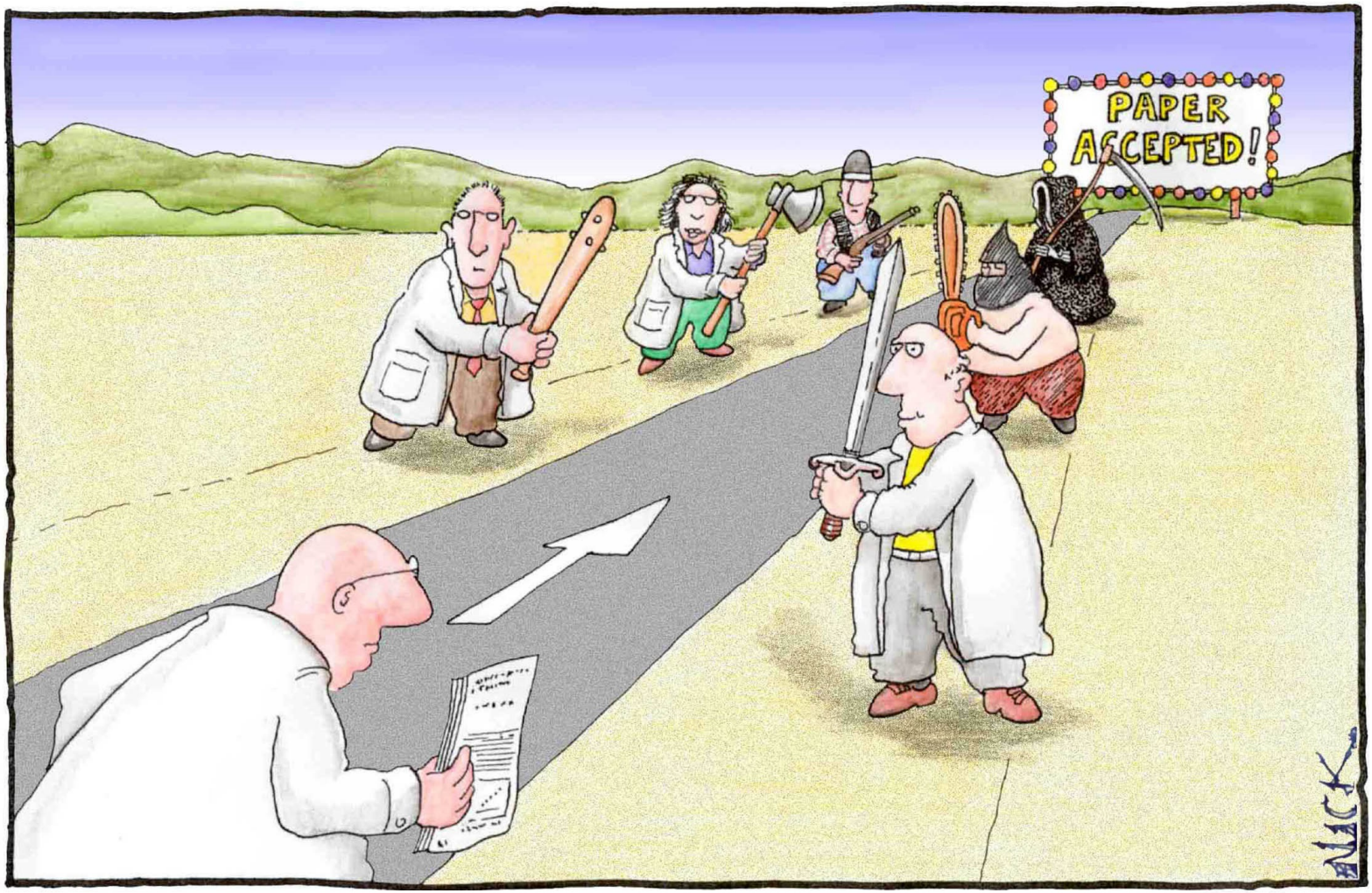
About Me …

About Me …

About Me …



About Me …

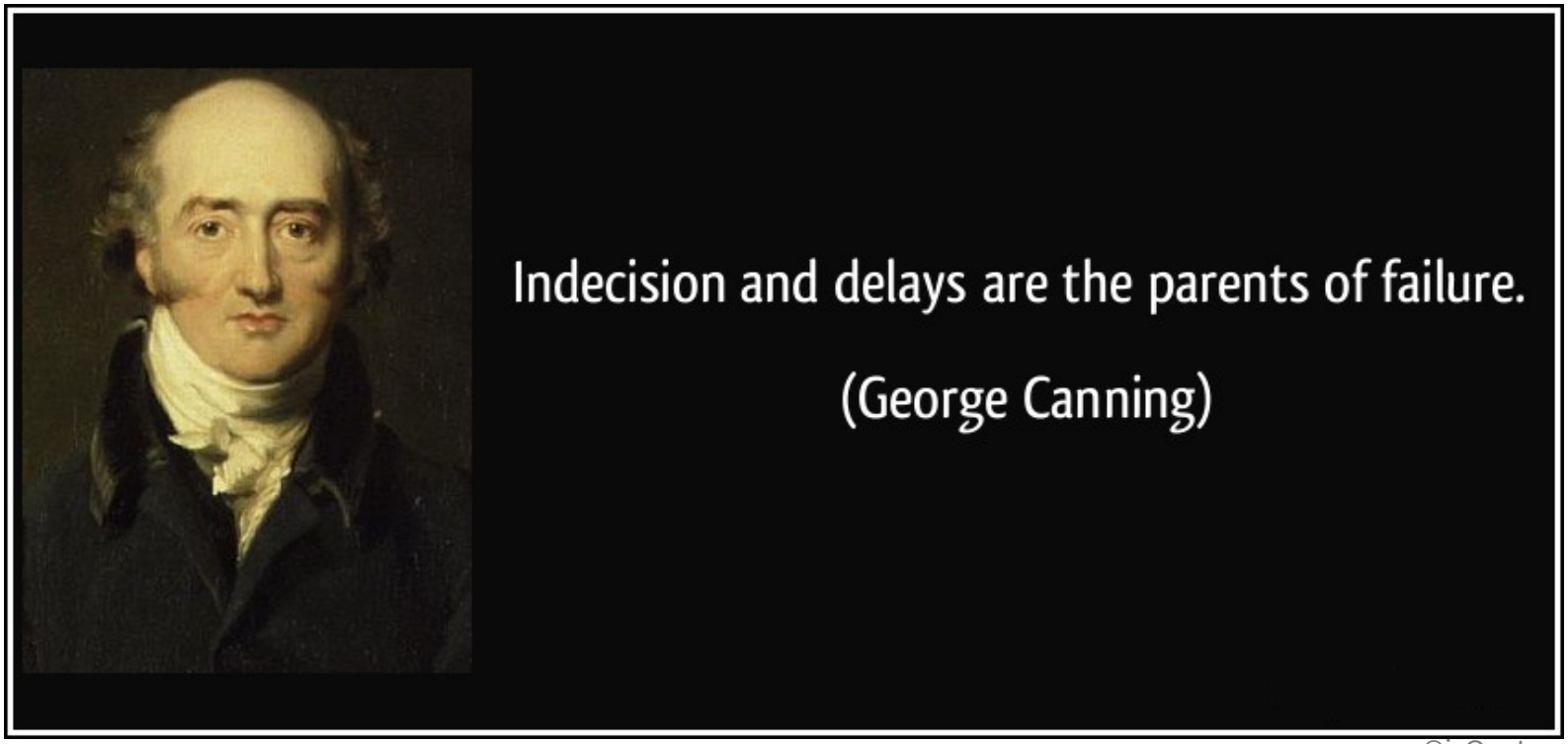


About Me …

Outline

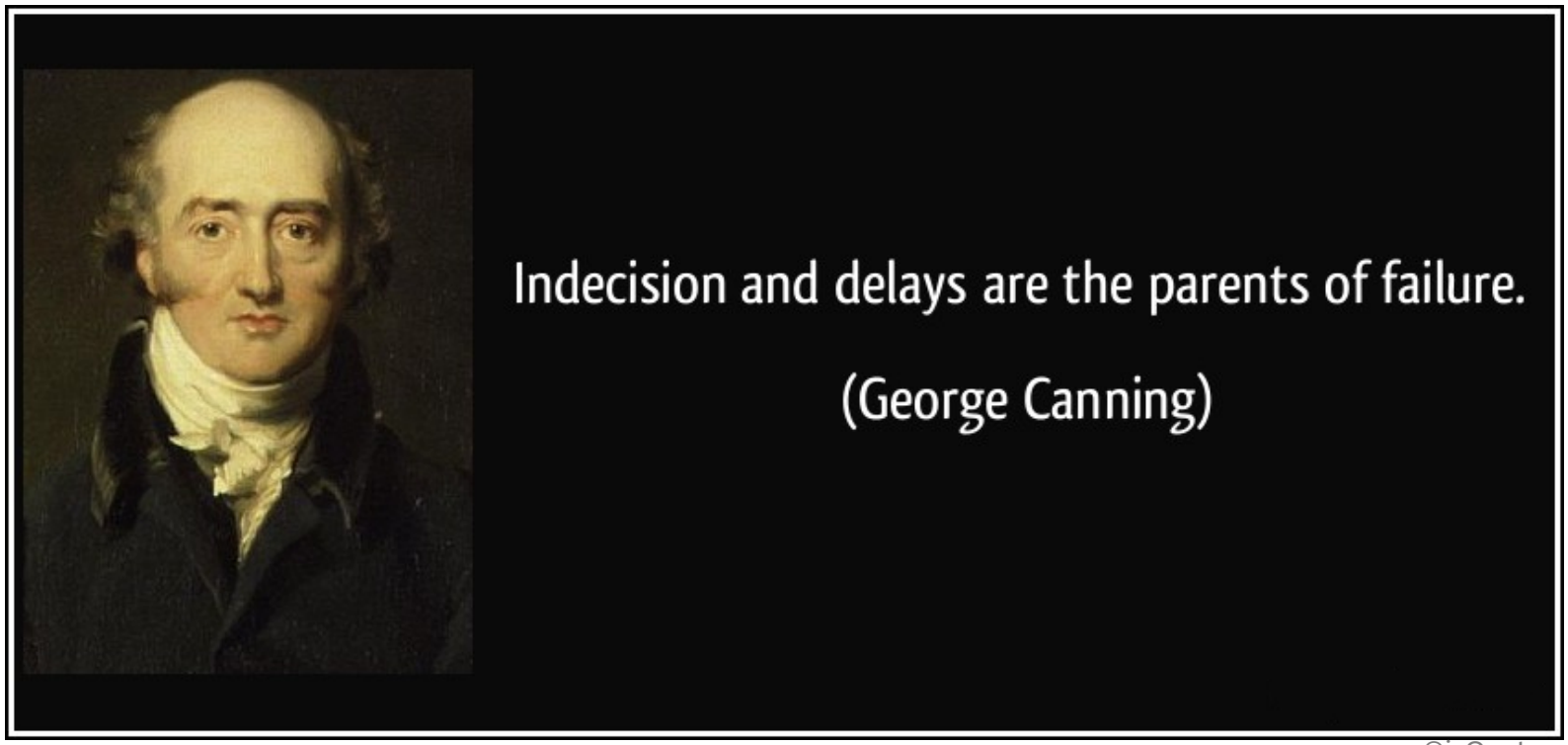
Outline

Advice by a Wise Man



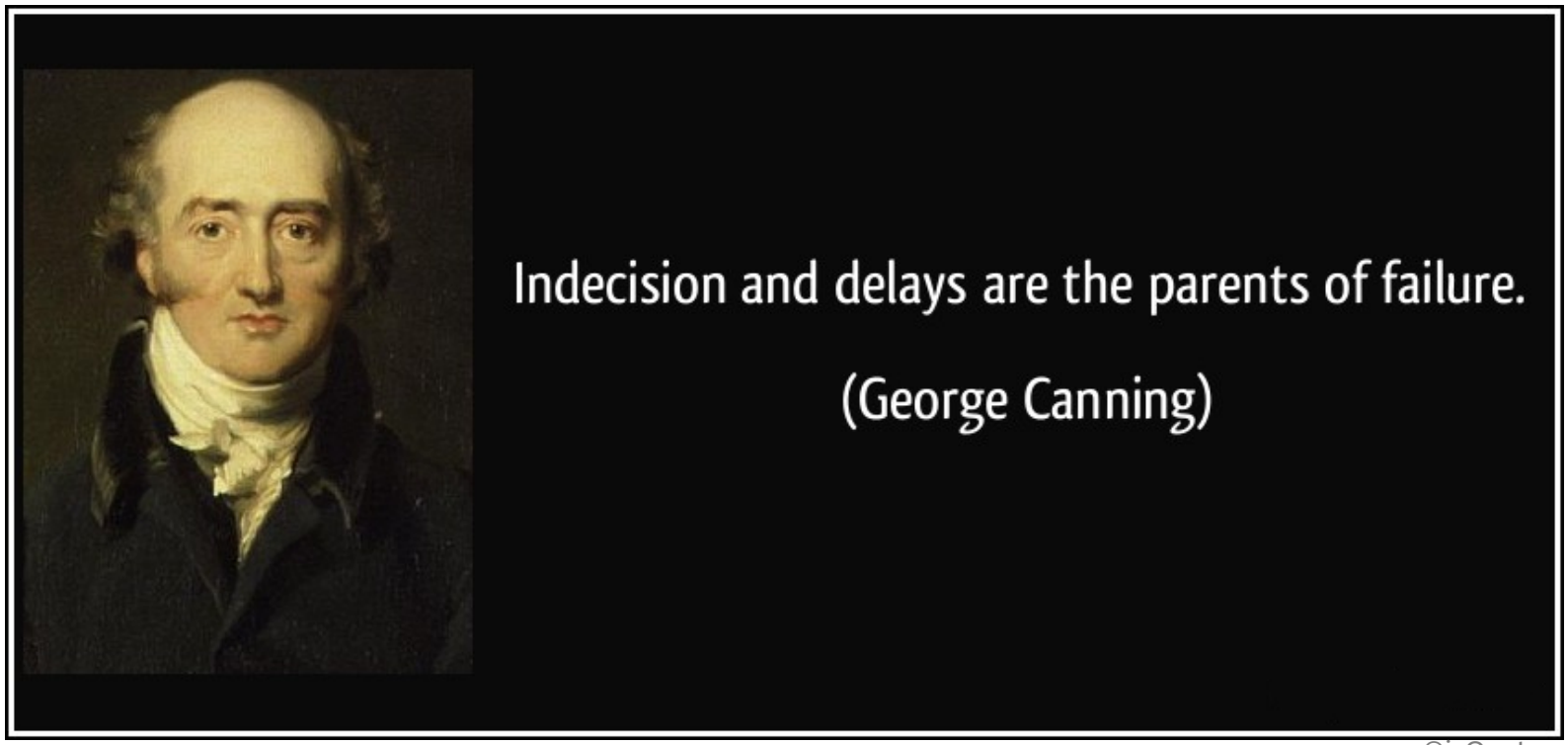
©izQuotes

Advice by a Wise Man



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

Advice by a Wise Man



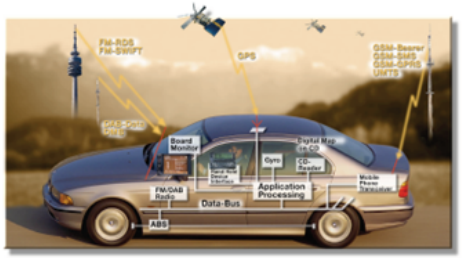
- Only relevant to ordinary people's life?
- Or to scientists, in particular comp. sci. and control folks, too?

Remember that Canning briefly controlled Great Britain!

Cyber-Physical Systems (CPS)

An open, interconnected form of embedded systems that integrates capabilities of computing, communication and control, among which many are safety-critical.

Automobiles



Medical

Entertainment

Handheld

Airplanes

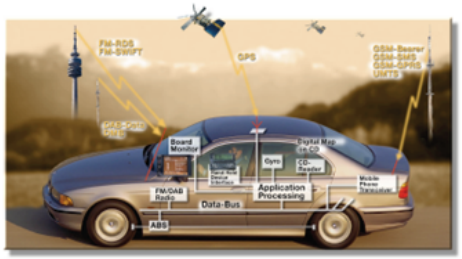
Military

Environmental Monitoring

Cyber-Physical Systems (CPS)

An open, interconnected form of embedded systems that integrates capabilities of computing, communication and control, among which many are safety-critical.

Automobiles



Medical

Entertainment

Handheld

Airplanes

Military

Environmental Monitoring

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

[Jeannette Wing]

Hybrid Behaviours

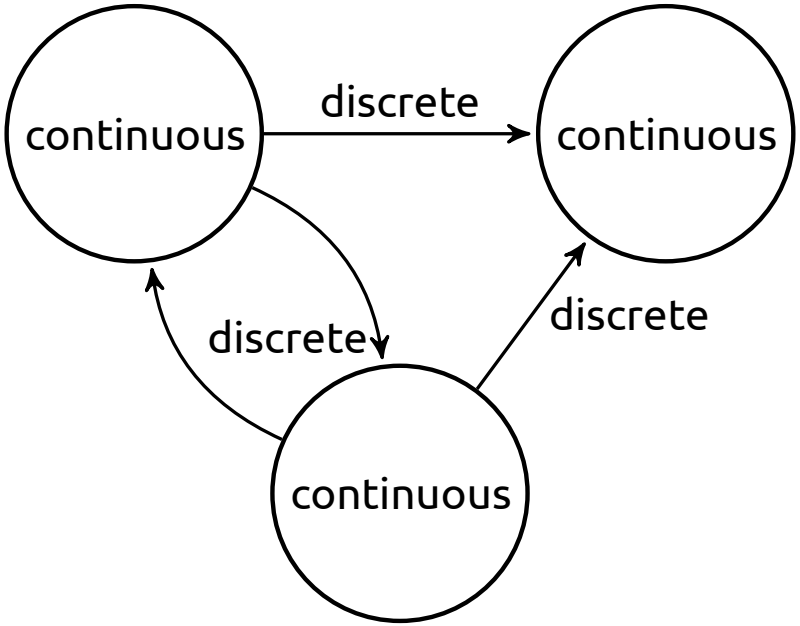


Figure – Macro : switching modes

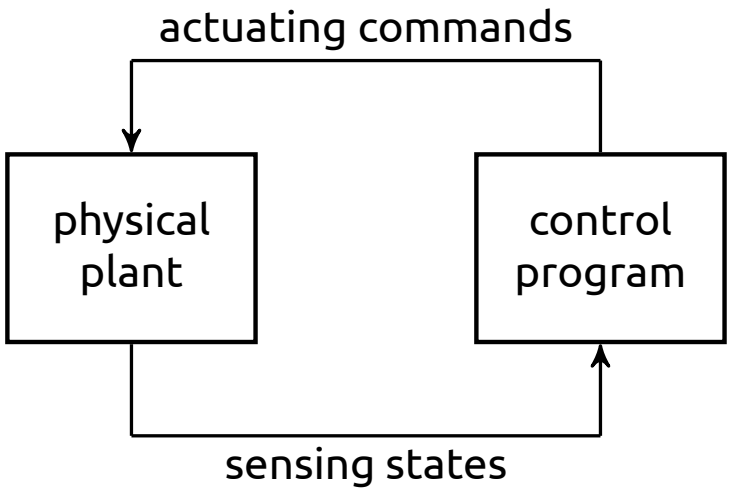
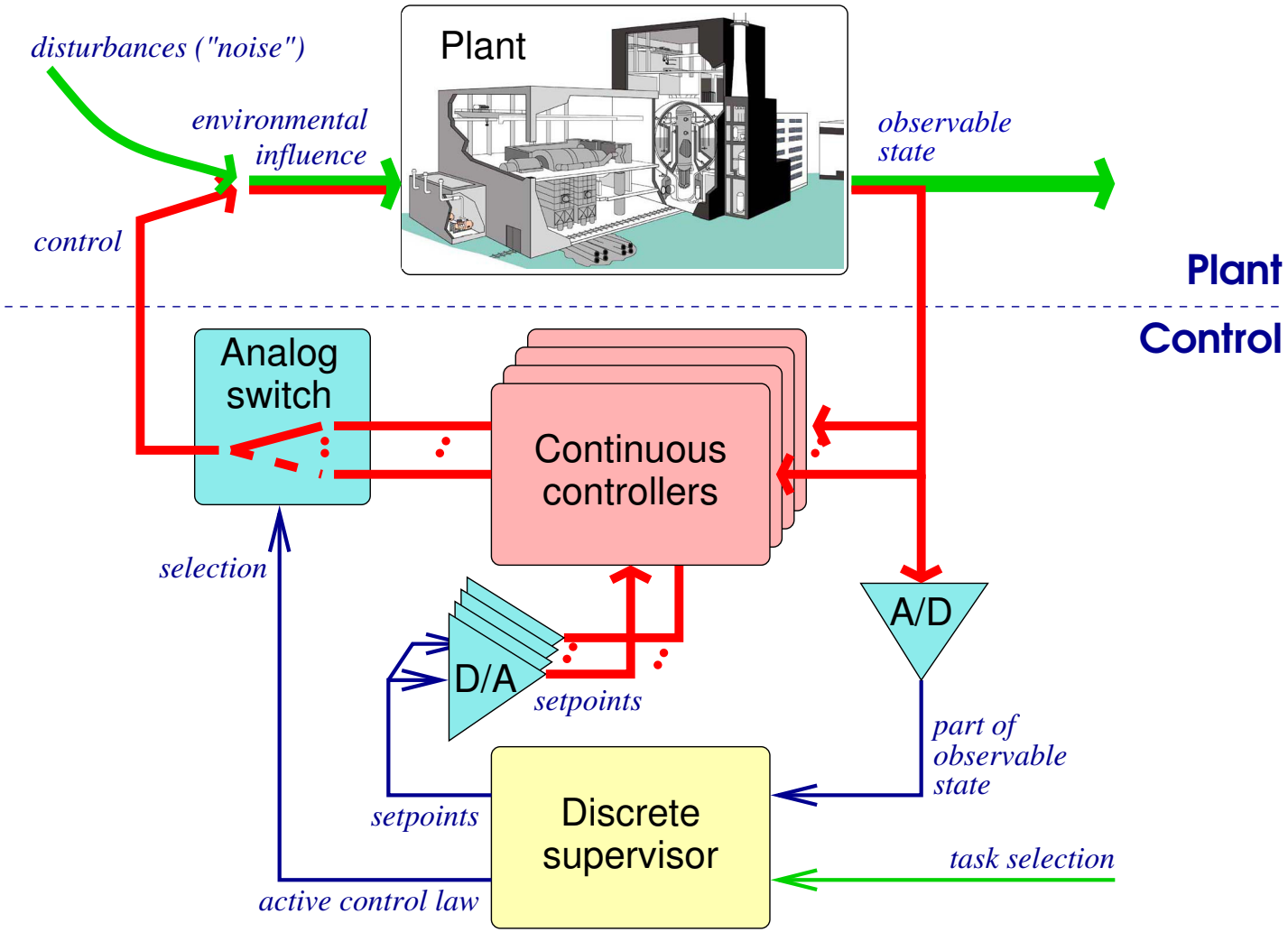
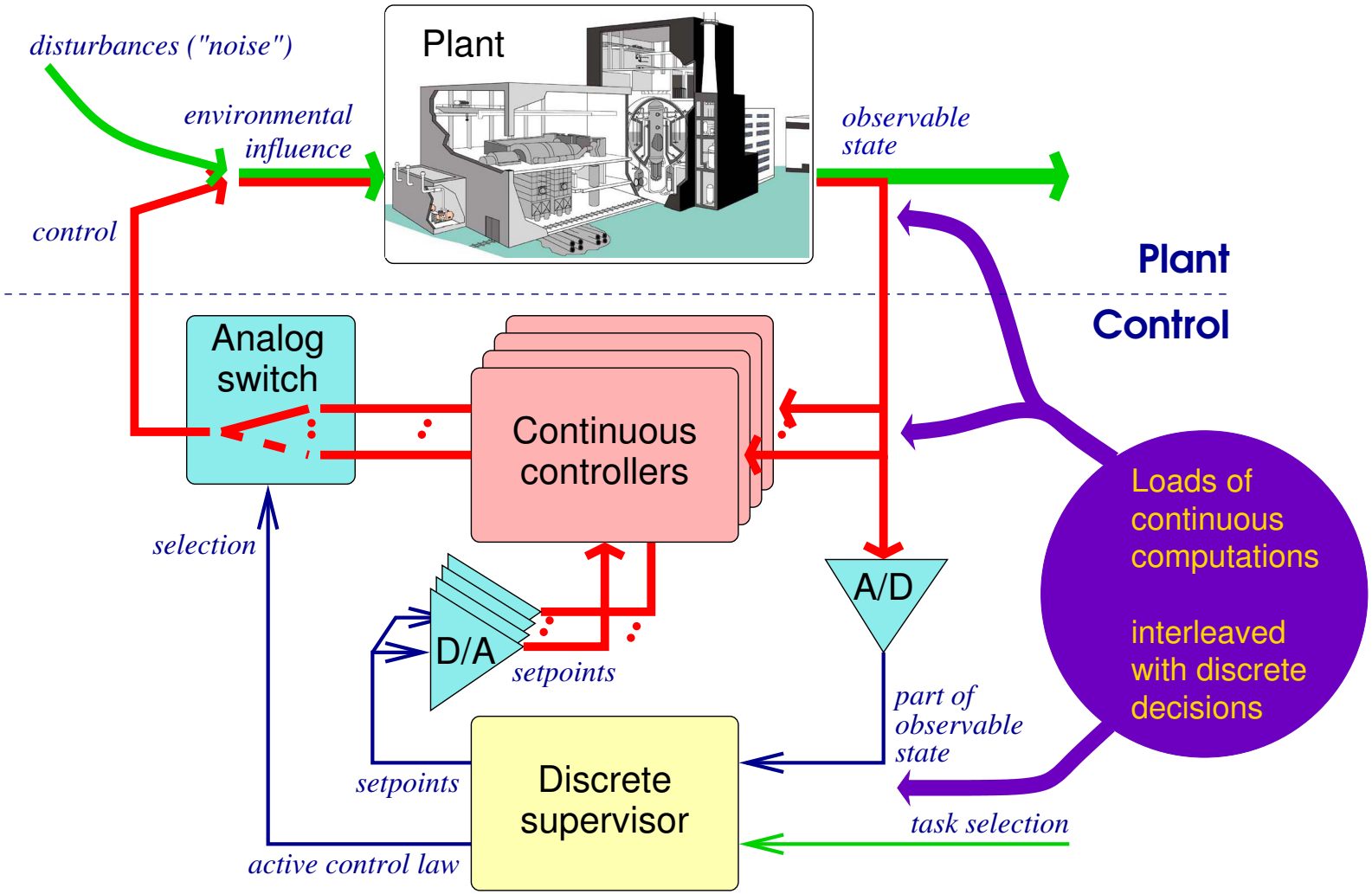


Figure – Micro : closed-loop feedback

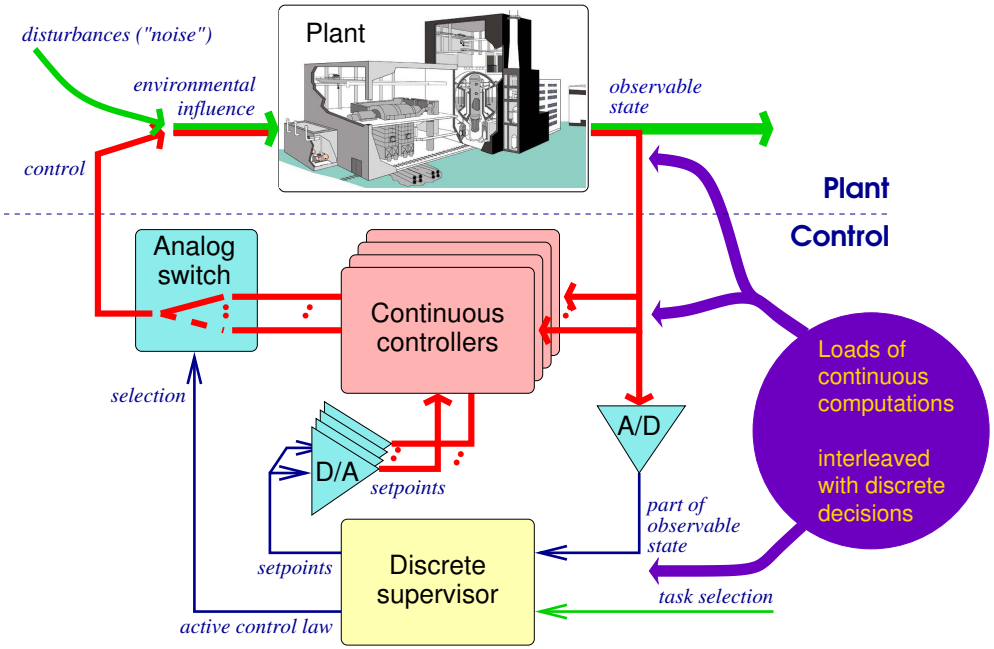
Hybrid Systems



Hybrid Systems



Hybrid Systems



Crucial question :

- How do the controller and the plant interact?

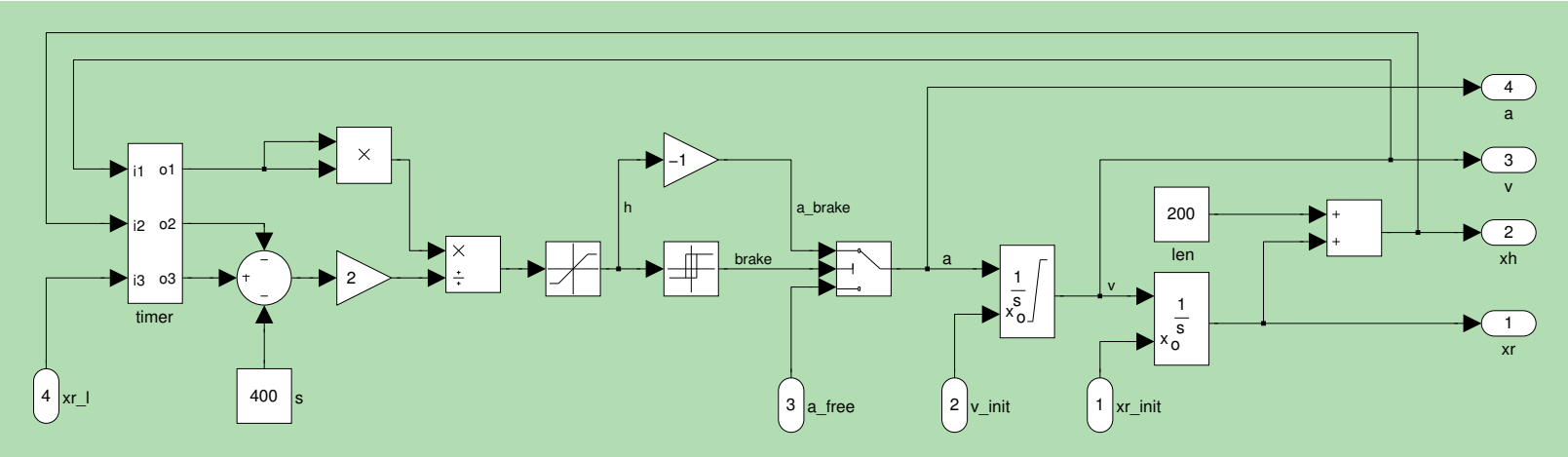
Traditional answer :

- Coupling assumed to be (or at least modelled as) delay-free.

⇒ Mode dynamics is covered by the conjunction of the individual ODEs.

⇒ Switching btw. modes is an immediate reaction to environmental conditions.

Instantaneous Coupling



©ETCS-3

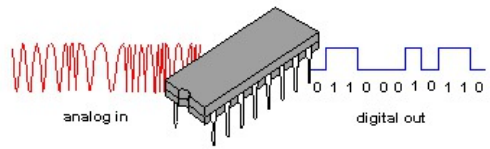
Following the tradition, above (rather typical) Simulink model assumes

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

Central questions :

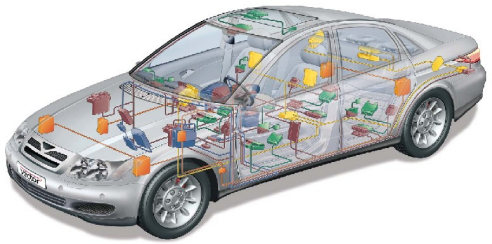
1. Is this realistic?
2. If not, does it have observable effect on control performance?
3. May that effect be detrimental or even harmful?

Q1 : Is Instantaneous Coupling Realistic?



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

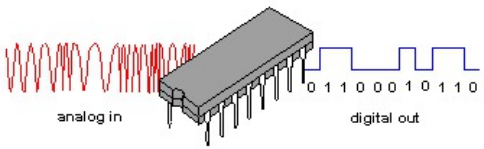
Digital signal processing, especially in complex sensors like CV, needs processing time, adding signal delays.



Networked control introduces communication latency into the feedback control loop.

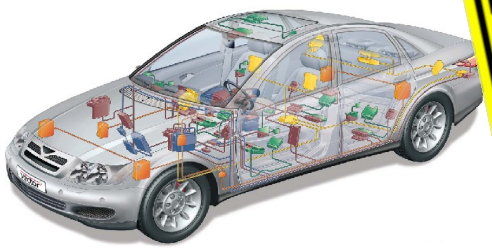
Harvesting, fusing, and forwarding data through sensor networks enlarge the latter by orders of magnitude.

Q1 : Is Instantaneous Coupling Realistic? – No.



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

Digital signal processing, especially in complex sensors like CV, ... time, adding signal de-...



... communication la-... loop.

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

Q1a : Resultant Forms of Delay

Delayed reaction : Reaction to a stimulus is not immediate.

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

- Thus amenable to the pertinent analysis tools.

⇒ Not of interest today.

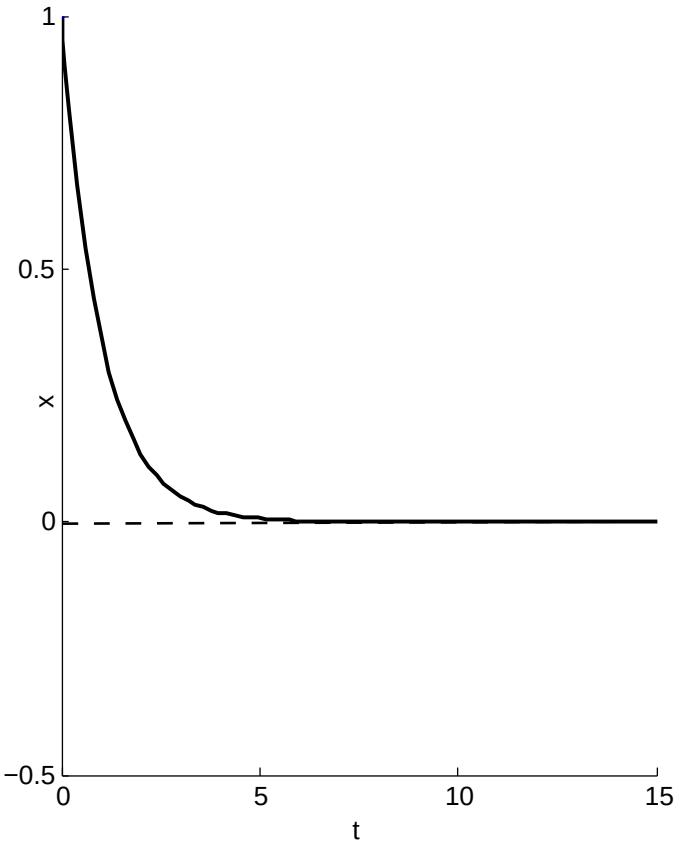
Network delay : Information of different age coexists and is queuing in the network when piped towards target.

- End-to-end latency may exceed sampling intervals etc. by orders of magnitude
- Not (continuous-time pipelined delay) or not efficiently (discrete-time pipelined delay) expressible in our std. models.

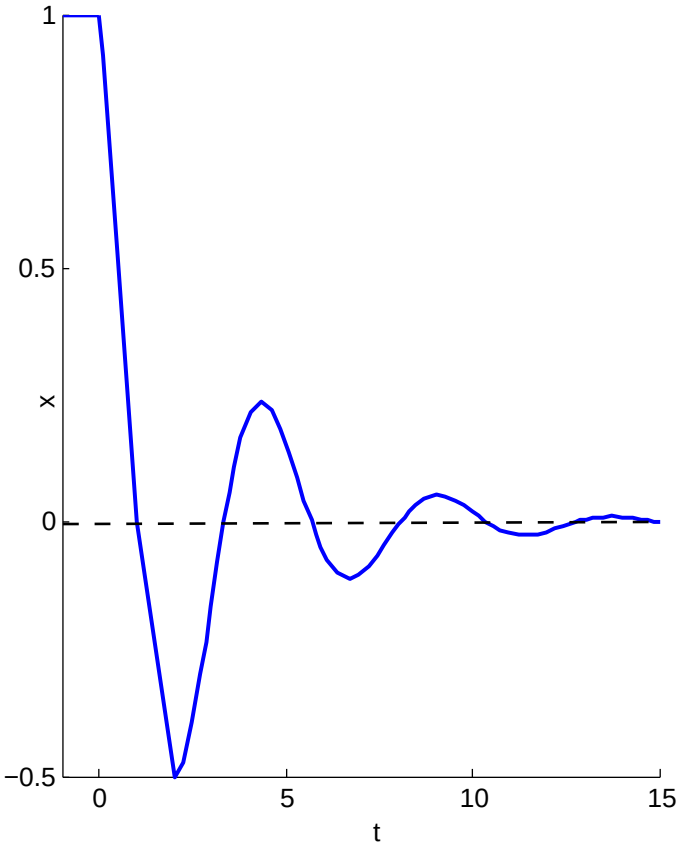
⇒ Our theme today.

Q2 : Do Delays Have Observable Effect?

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

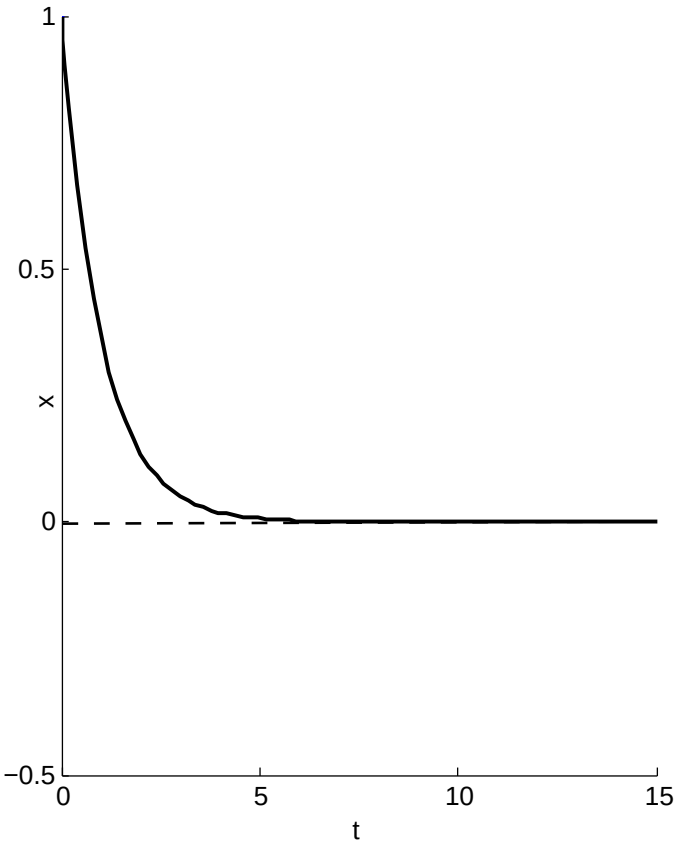


$$\begin{cases} \dot{x}(t) = -x(t-1) \\ x([-1,0]) \equiv 1 \end{cases}$$

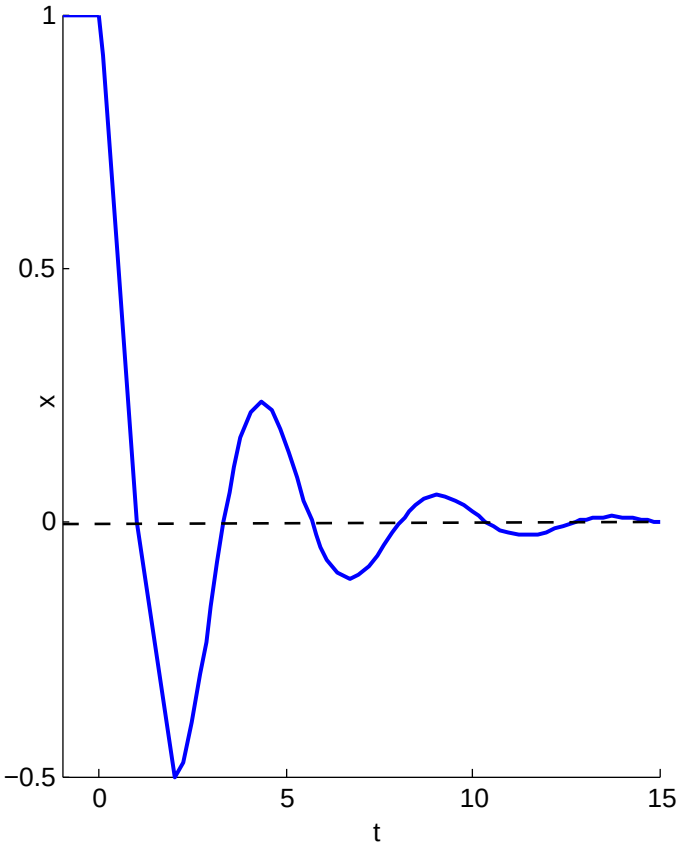


Q2 : Do Delays Have Observable Effect? – Yes, they have.

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



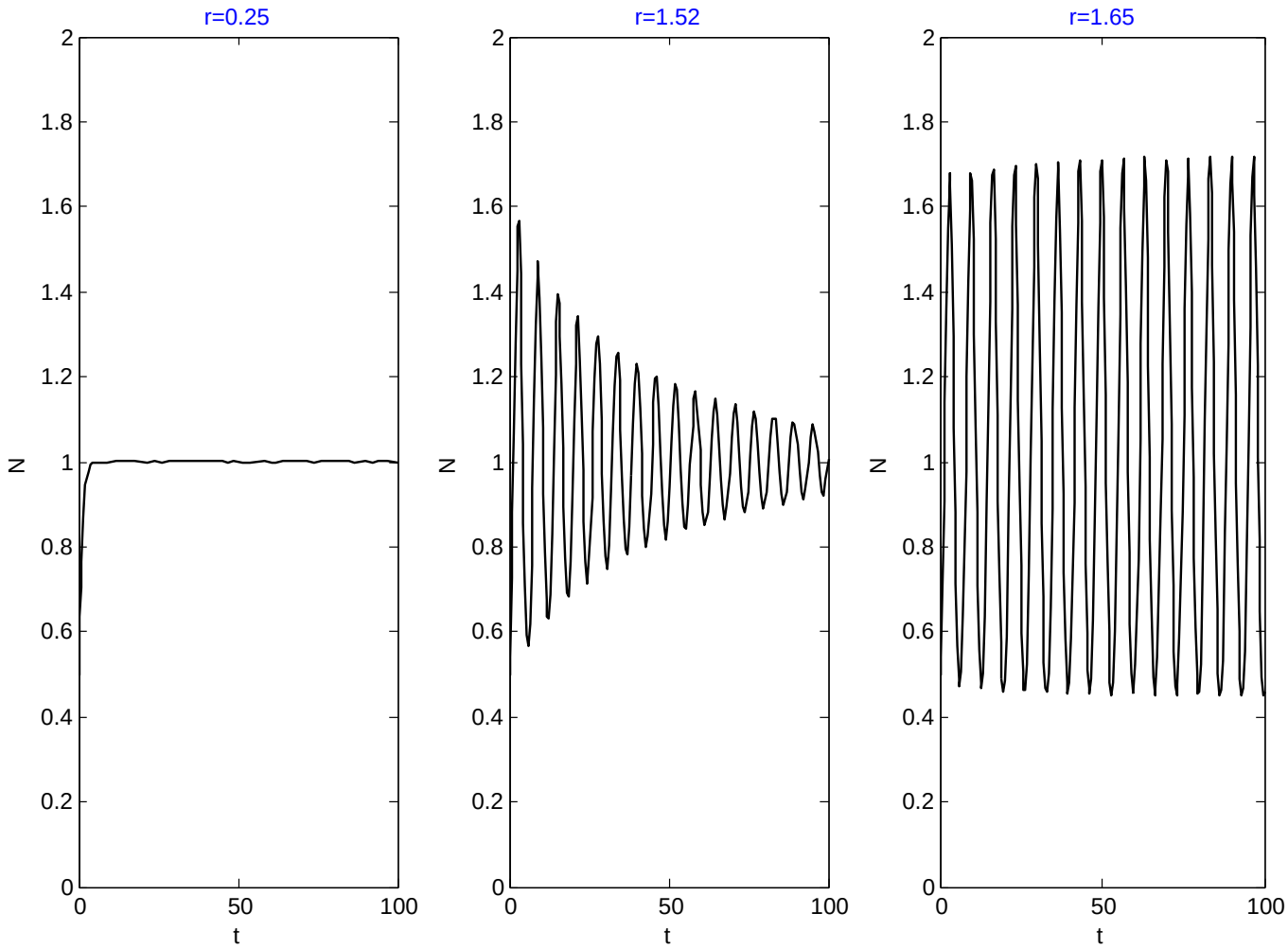
$$\begin{cases} \dot{x}(t) = -x(t-1) \\ x([-1,0]) \equiv 1 \end{cases}$$



Q3 : May the Effects be Harmful?

- Delayed logistic equation [G. Hutchinson, 1948] :

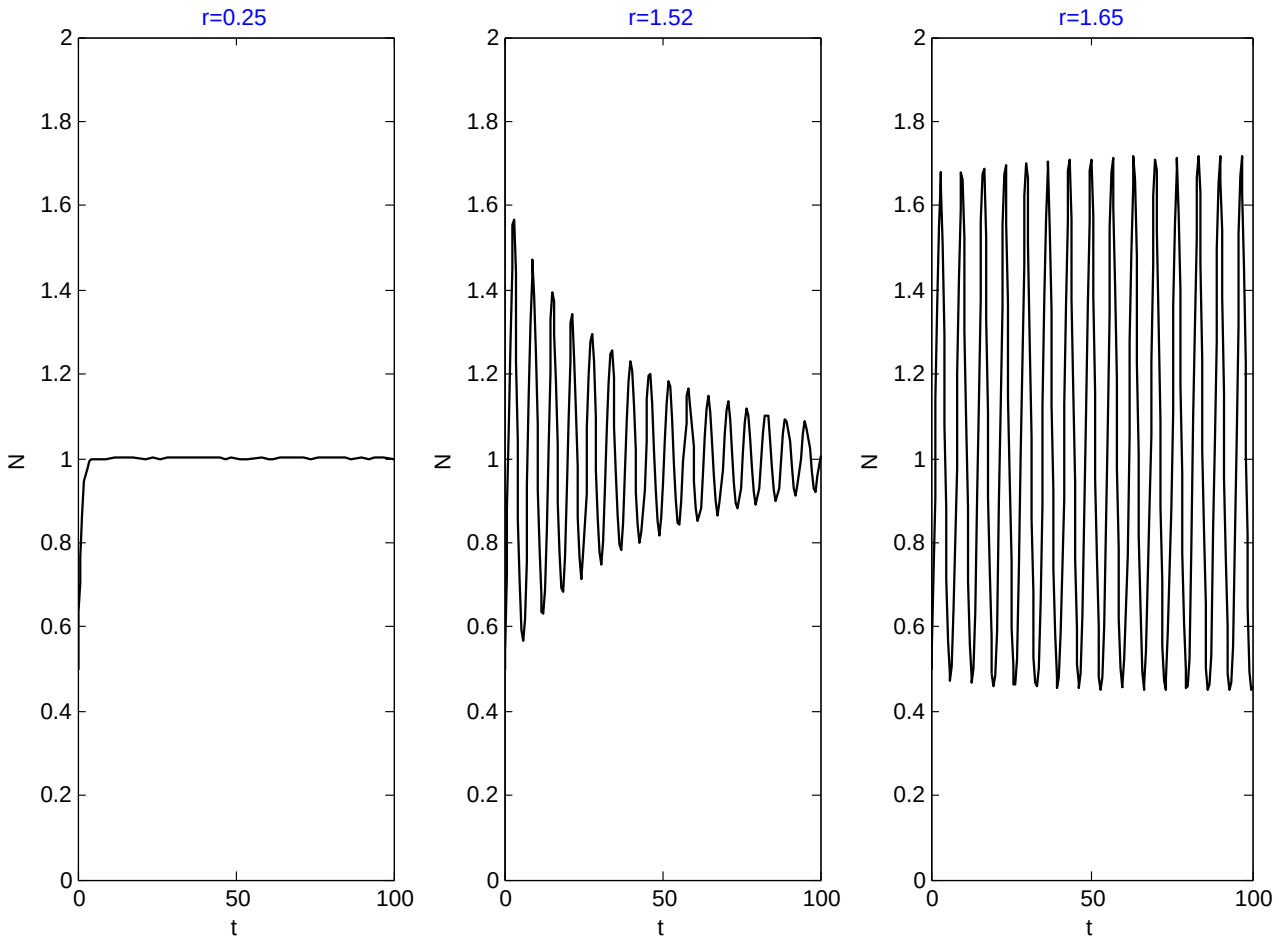
$$\dot{N}(t) = N(t)[1 - N(t - r)]$$



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

- Delayed logistic equation [G. Hutchinson, 1948] :

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



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!

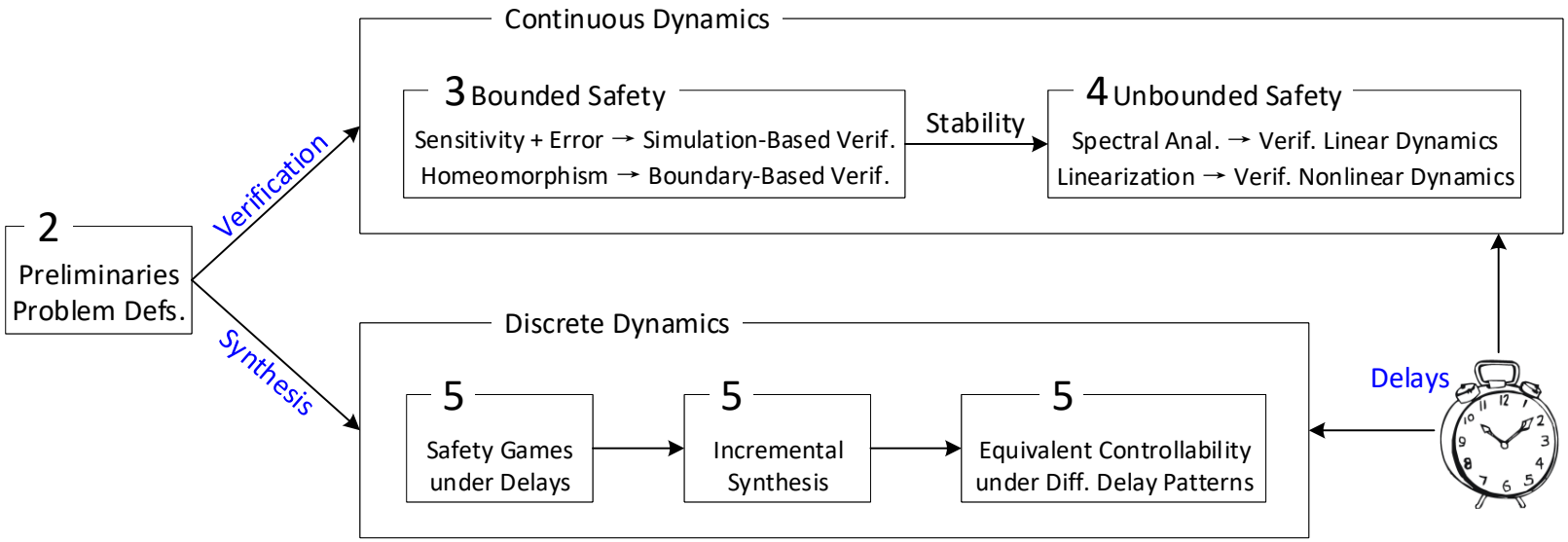
Surprisingly, they don't :

1. S. Prajna, A. Jadbabaie : *Meth. f. safety verification of time-delay syst.* (CDC'05)
2. L. Zou, M. Fränzle, N. Zhan, P.N. Mosaad : *Autom. verific. of stabil. and safety* (CAV'15)
3. 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)
4. Z. Huang, C. Fan, S. Mitra : *Bounded invariant verification for time-delayed nonlinear networked dynamical systems* (NAHS'16)
5. P.N. Mosaad, M. Fränzle, B. Xue : *Temporal logic verification for DDEs* (ICTAC'16)
6. E. Goubault, S. Putot, L. Sahlman : *Approximating flowpipes for DDEs* (CAV'18)
7. [M. Zimmermann. LICS'18, GandALF'17], [F. Klein & M. Zimmermann. ICALP'15, CSL'15]

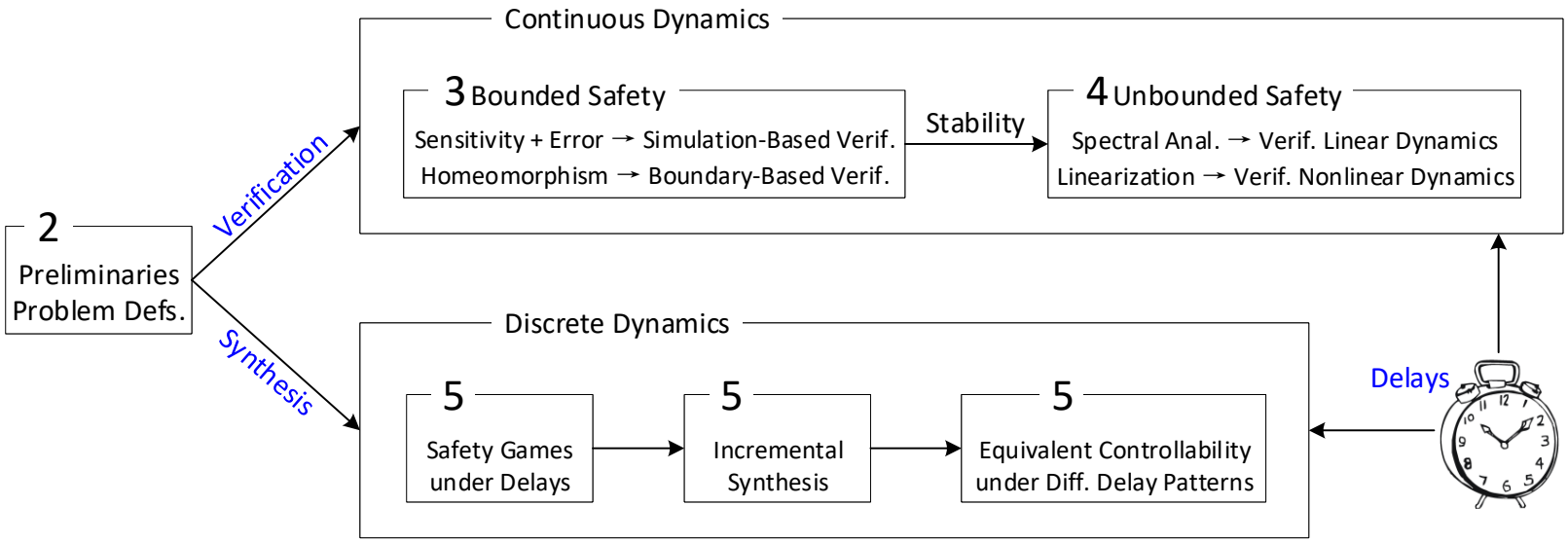
(plus a handful of related versions)

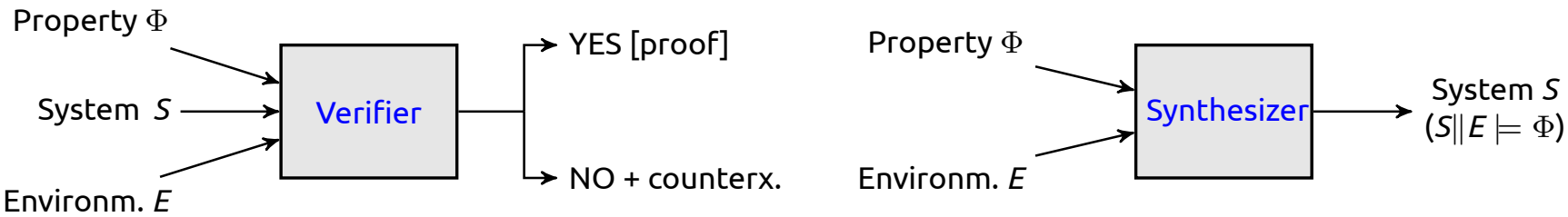
Outline

Structural Overview



Structural Overview





Solving Delay Differential Equations (DDEs)

A formal model of delayed feedback control

—Joint work with M. Fränzle, Y. Li, S. Feng, P. N. Mosaad, B. Xue and N. Zhan—

Delayed Differential Dynamics (a.k.a., DDEs)

Historical motivation :

> *"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]

Delayed Differential Dynamics (a.k.a., DDEs)

Historical motivation :

> *"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."*

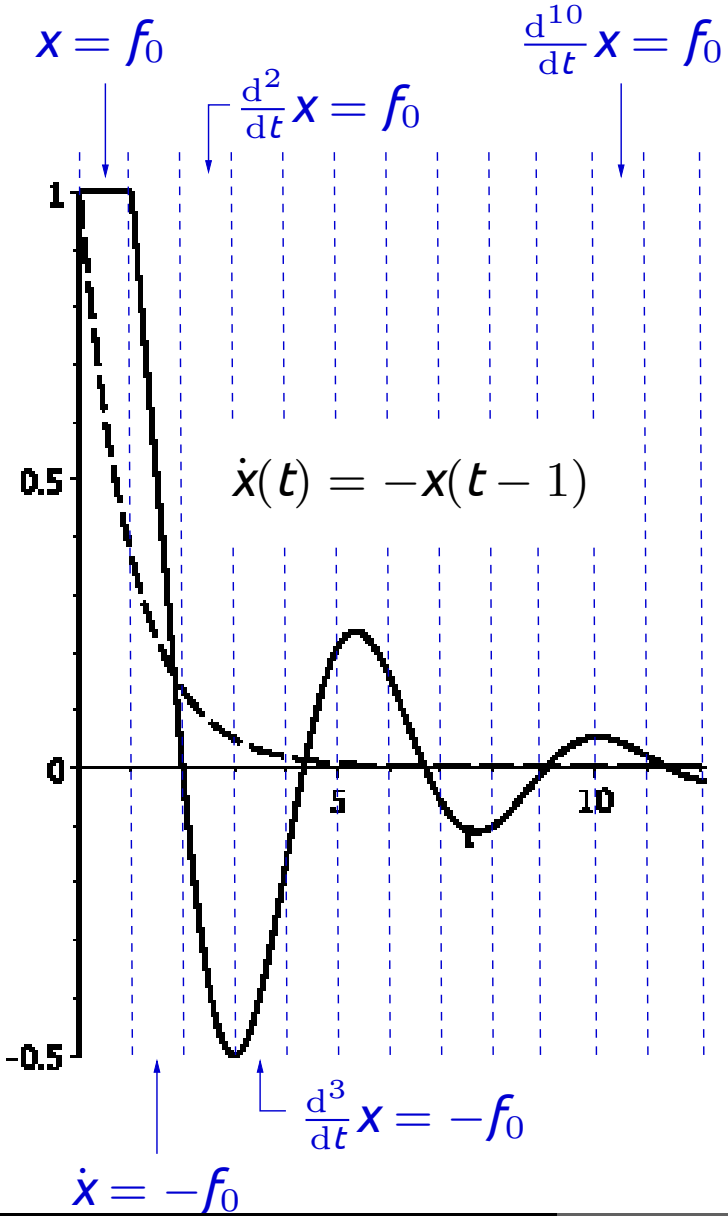
[Richard Bellman and Kenneth L. Cooke, 1963]

Delay Differential Equations (DDEs)

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

The unique *solution* (*trajectory*) : $\xi_{\mathbf{x}_0}(t) : [-r_{\max}, \infty) \mapsto \mathbb{R}^n$.

Why DDEs are Hard(er)

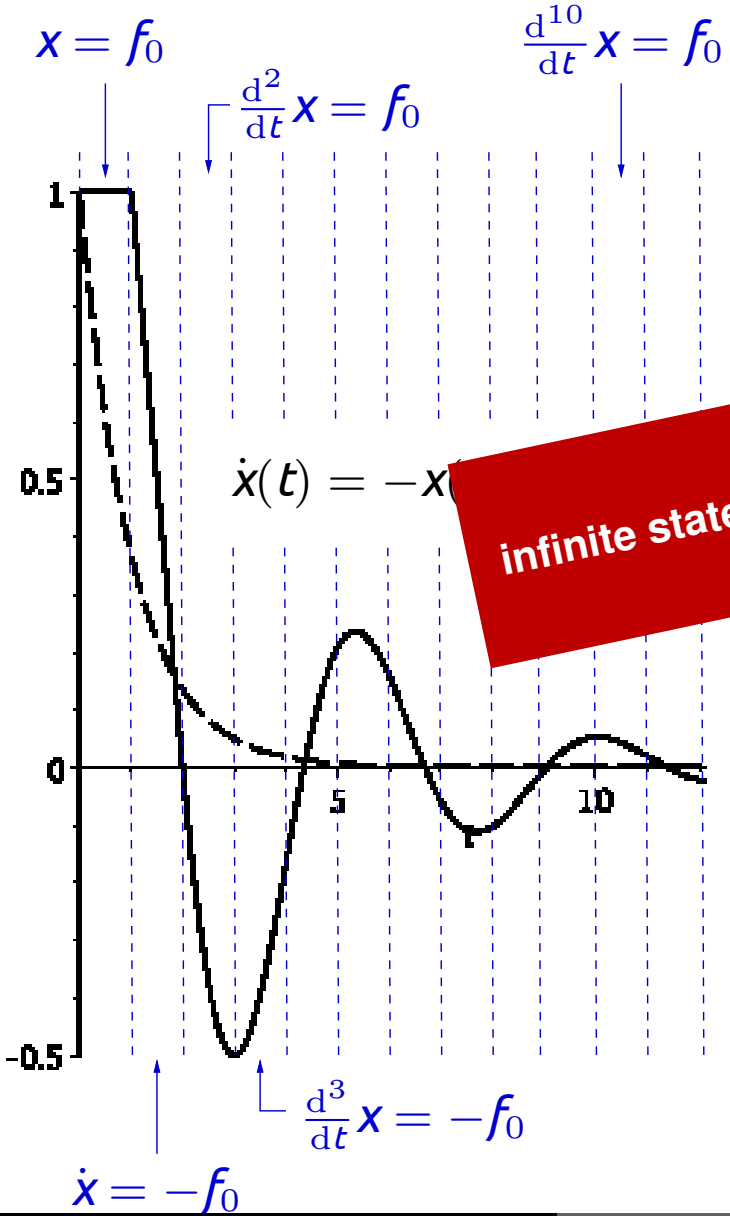


DDEs constitute a model of system dynamics beyond "state snapshots" :

- 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 con… model of system "state snapshots" :

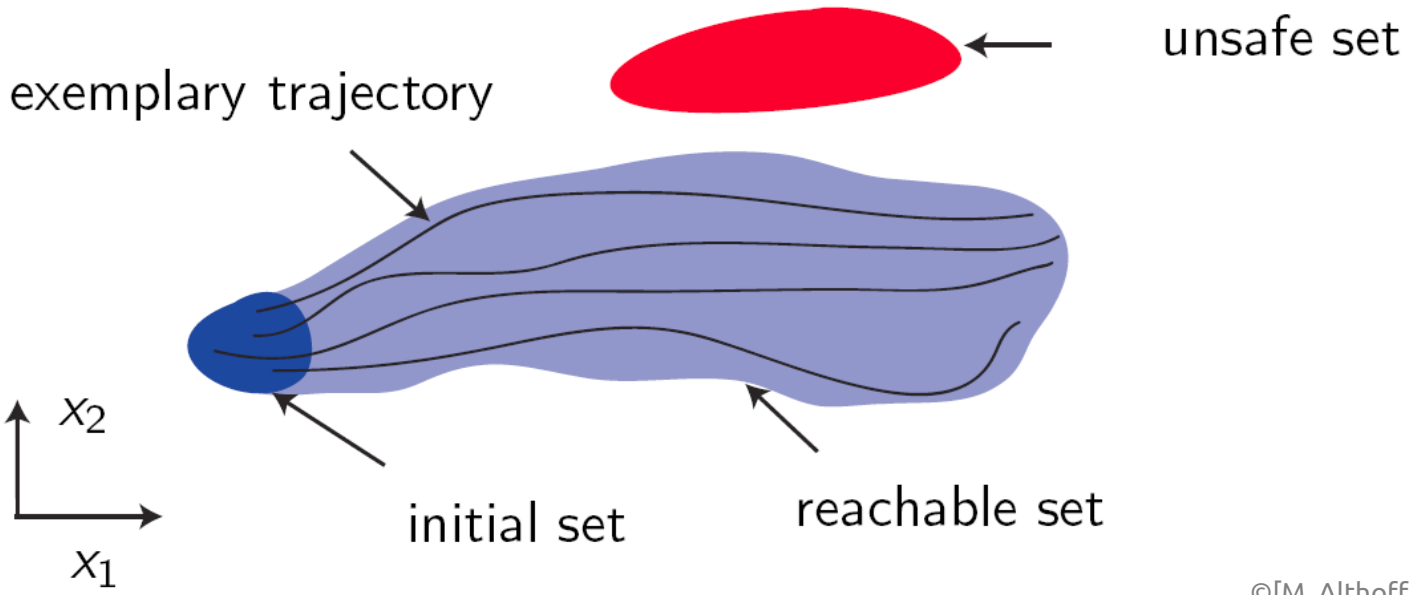
- …functional state" …state in the $\mathbb{R}^n$.
- Thus providing rather infallible, infinite-dimensional memory of the past.

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

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



©[M. Althoff, 2010]

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

Method I : Simulation-Based Verification

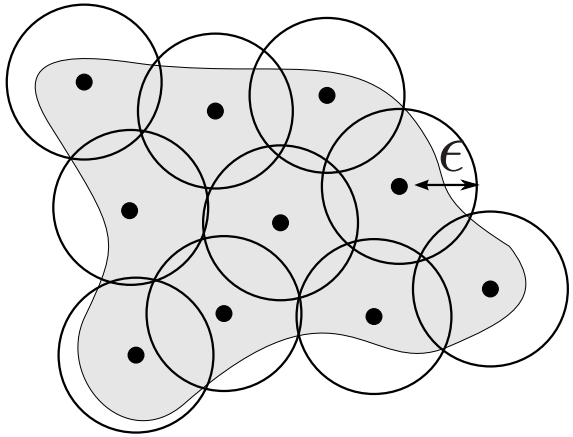


Figure – A finite ϵ-cover of the initial set of states.

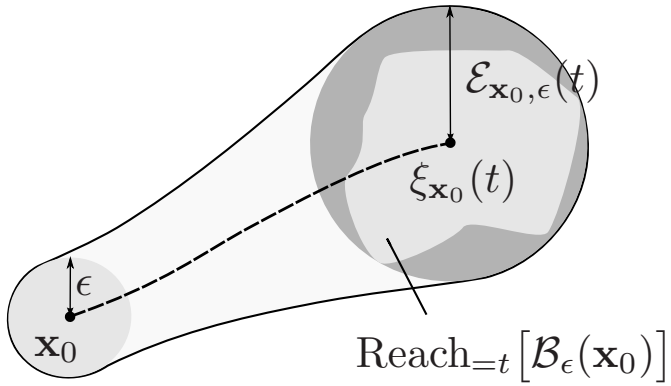
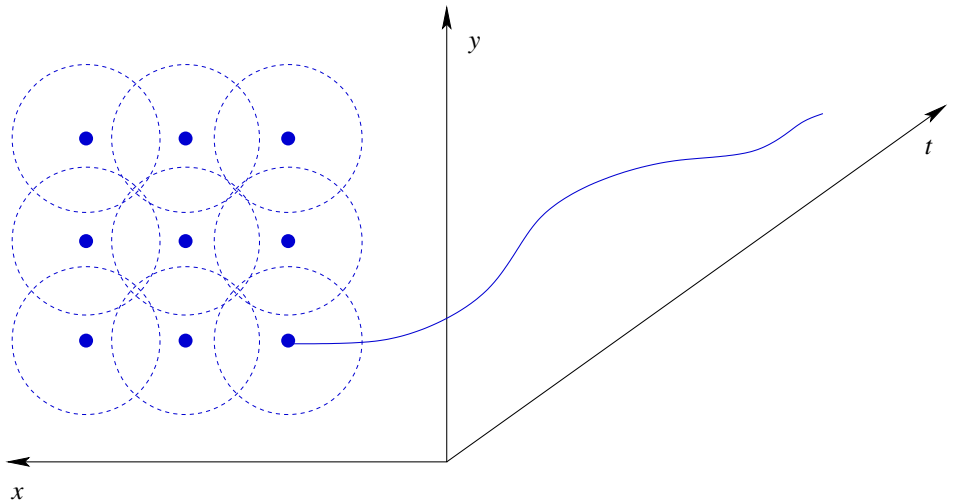


Figure – An Over-approximation of the reachable set by bloating the simulation.

©A. Donzé & O. Maler, 2007

Method I : Simulation-Based Verification

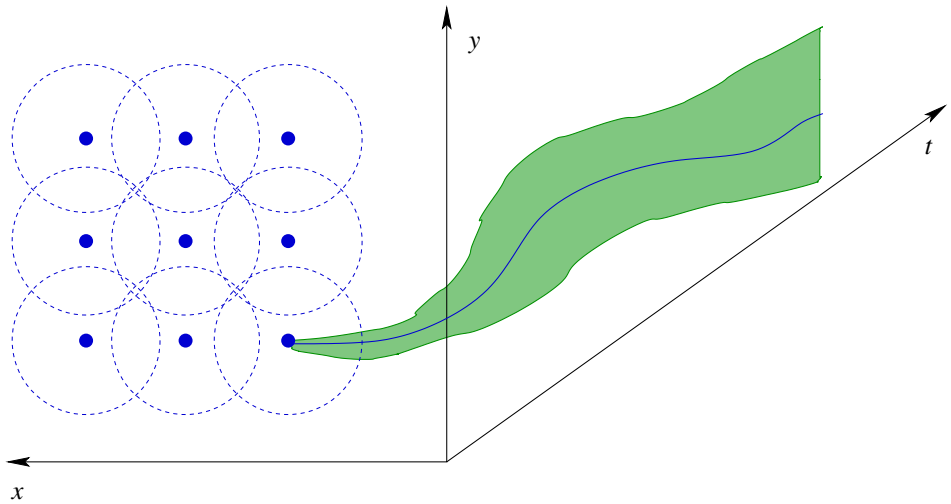
1. Do numerical simulation on a (sufficiently dense) sample of initial states.
2. Add (pessimistic) local-error analysis.
3. "Bloat" the resulting trajectories by sensitivity analysis.



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

Method I : Simulation-Based Verification

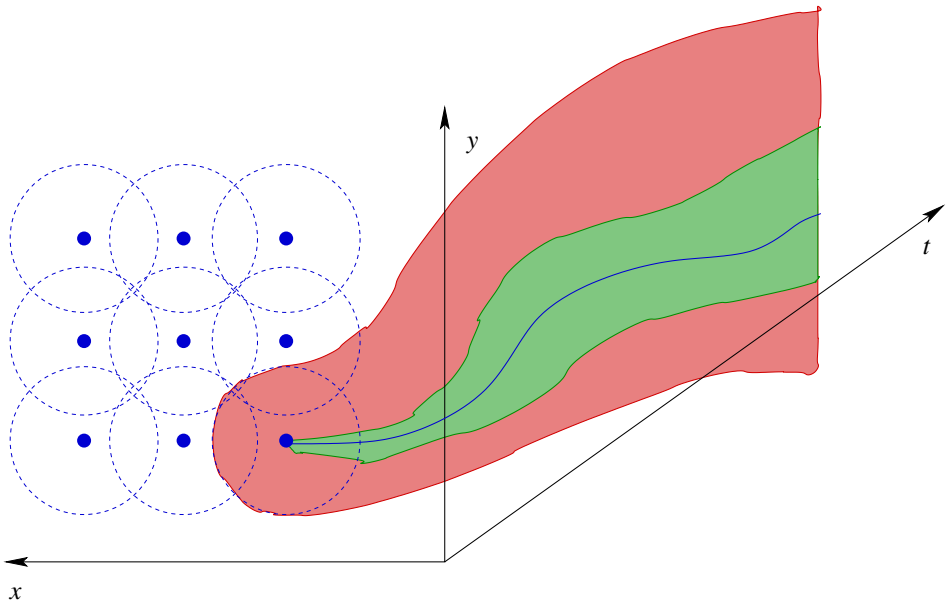
1. Do numerical simulation on a (sufficiently dense) sample of initial states.
2. Add (pessimistic) local-error analysis.
3. "Bloat" the resulting trajectories by sensitivity analysis.



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

Method I : Simulation-Based Verification

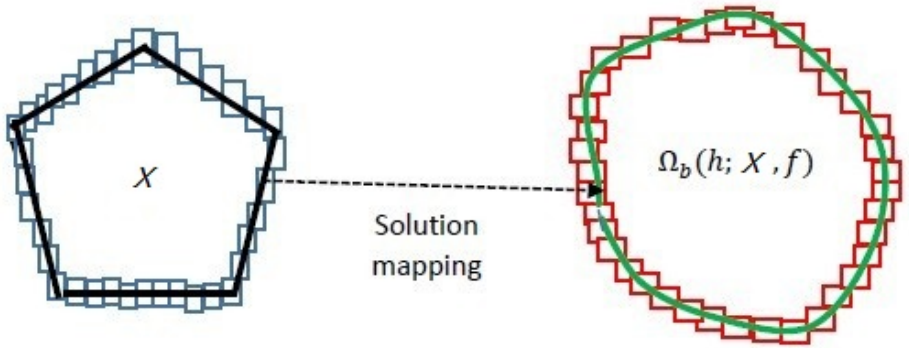
1. Do numerical simulation on a (sufficiently dense) sample of initial states.
2. Add (pessimistic) local-error analysis.
3. "Bloat" the resulting trajectories by sensitivity analysis.



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

Method II : Boundary-Based Approximation

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.

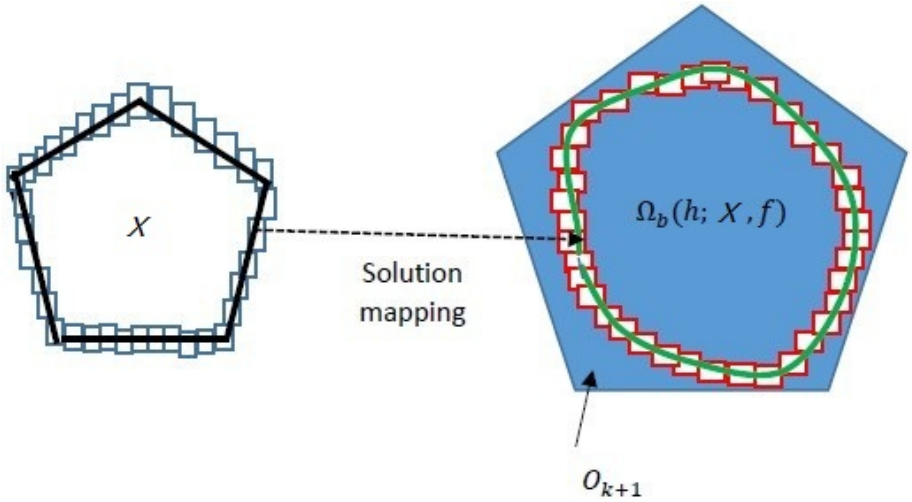


$$r \leq \min \left\{ \frac{\epsilon - 1}{\epsilon n^2 M' R}, \frac{\ln R}{2\sqrt{n} n M'}, \frac{\epsilon - 1}{\epsilon (n^2 MR + n^2 NR\epsilon)}, \frac{\ln \frac{R^2+1}{2}}{\sqrt{n}(2nM + n^2 NR\epsilon)} \right\}$$

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

Method II : Boundary-Based Approximation

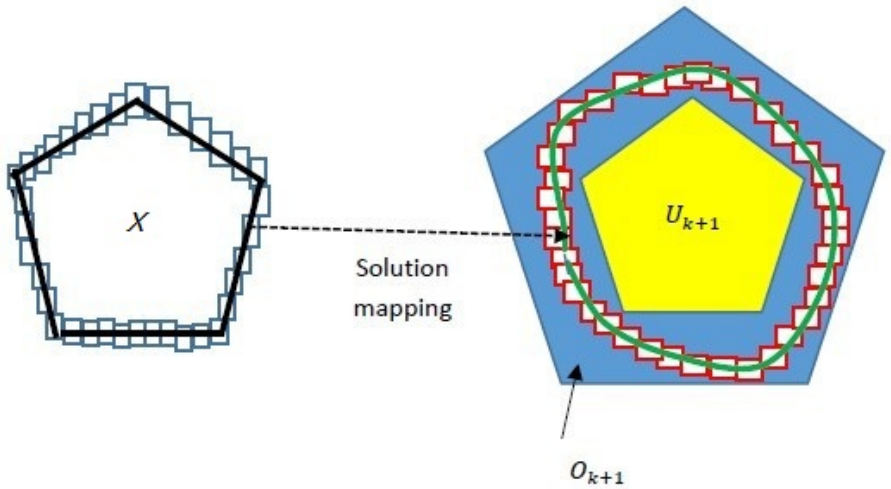
1. Impose a homeomorphism by bounding the time-lag through sensitivity analysis.
2. Compute an enclosure of the reachable set's boundary.
3. Over- (under-)approximate the reachable set by incl. (excl.) the enclosure.



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

Method II : Boundary-Based Approximation

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.

Method III : Unbounded Verification Leveraging Stability Criteria

For linear dynamics

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

Method III : Unbounded Verification Leveraging Stability Criteria

For linear dynamics

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

The characteristic equation :

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

Method III : Unbounded Verification Leveraging Stability Criteria

For linear dynamics

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

The characteristic equation :

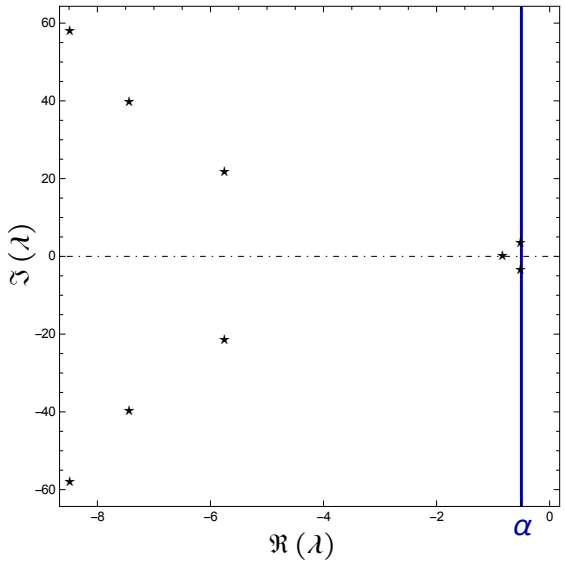
$$\det\left(\lambda I - A - Be^{-r\lambda}\right) = 0$$

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

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

Method III : Unbounded Verification Leveraging Stability Criteria

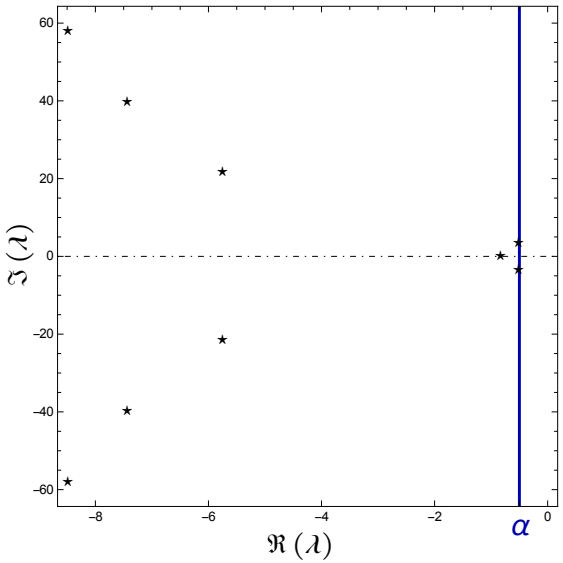
1. Identify the rightmost eigenvalue (and hence α) and construct K.
2. Compute T^* based on the exponential estimation spanned by α and K.
3. Reduce to bounded verifi., i.e., $\forall T > T^*$, ∞-safe $\iff$ T-safe.

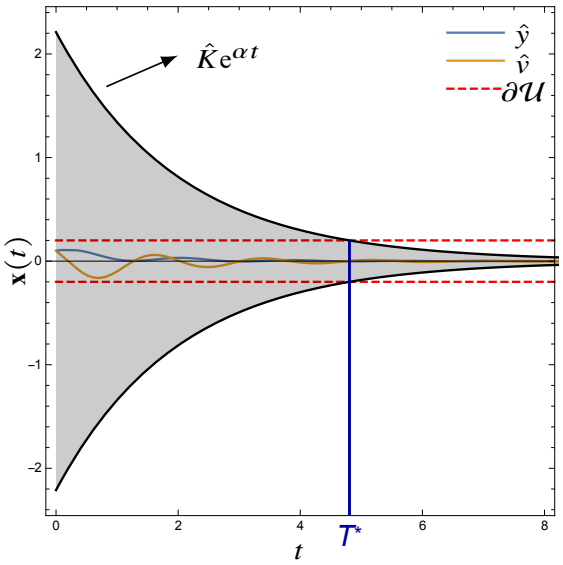


⇒ S. Feng, M. Chen, N. Zhan, M. Fränzle, B. Xue : *Taming delays in dyn. syst. : Unbounded verif. of DDEs*. CAV'19.

Method III : Unbounded Verification Leveraging Stability Criteria

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.

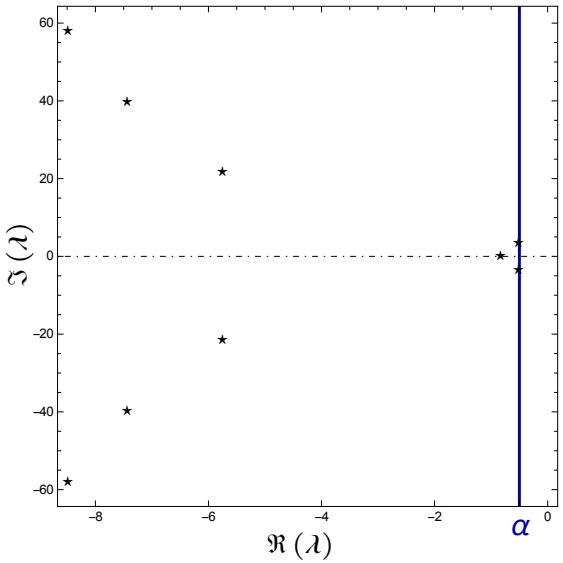


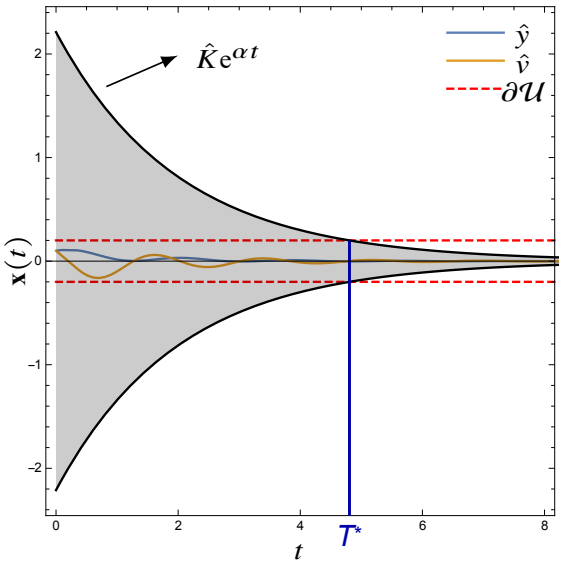


⇒ S. Feng, M. Chen, N. Zhan, M. Fränzle, B. Xue : *Taming delays in dyn. syst. : Unbounded verif. of DDEs.* CAV'19.

Method III : Unbounded Verification Leveraging Stability Criteria

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.

Method III : Unbounded Verification Leveraging Stability Criteria

For nonlinear dynamics

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

Method III : Unbounded Verification Leveraging Stability Criteria

For nonlinear dynamics

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

The linearization yields

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

Method III : Unbounded Verification Leveraging Stability Criteria

For nonlinear dynamics

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

The linearization yields

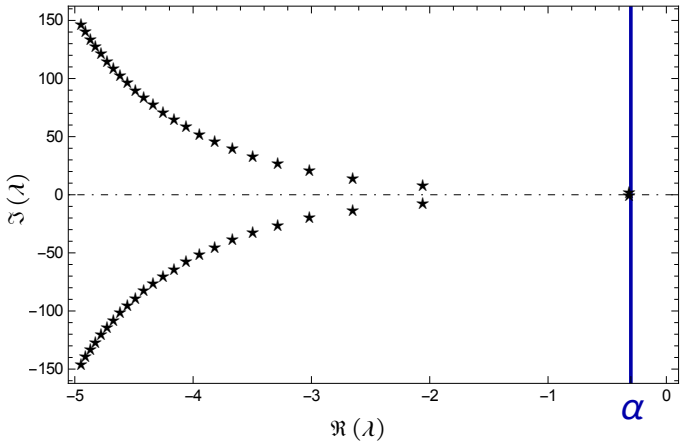
$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{x}(t-r)$$

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

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

Method III : Unbounded Verification Leveraging Stability Criteria

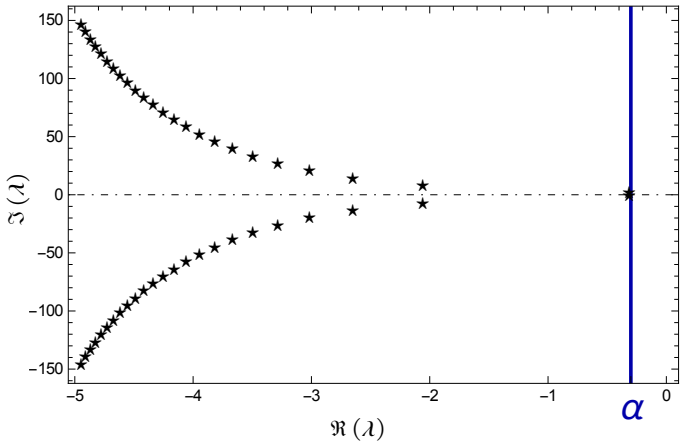
1. Identify the rightmost eigenvalue (and hence α) and construct K.
2. Compute T^* and δ, and hence T' (by bounded verifiers) that $\|\Omega\| < \delta$ within T'.
3. Reduce to bounded verifi., i.e., $\forall T > T' + T^*$, ∞-safe $\Longleftrightarrow$ T-safe.

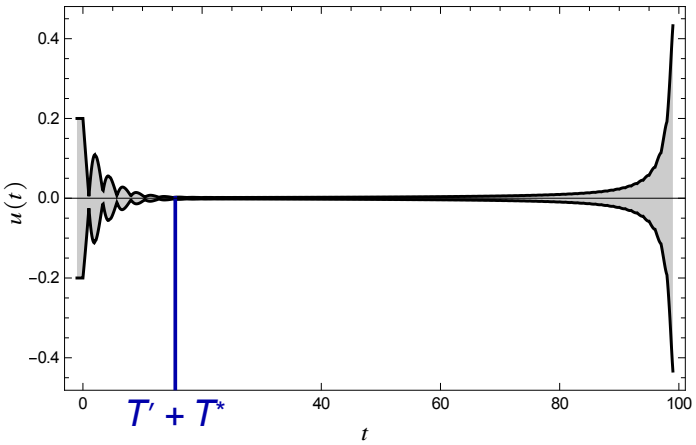


⇒ S. Feng, M. Chen, N. Zhan, M. Fränzle, B. Xue : *Taming delays in dyn. syst. : Unbounded verif. of DDEs*. CAV'19.

Method III : Unbounded Verification Leveraging Stability Criteria

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

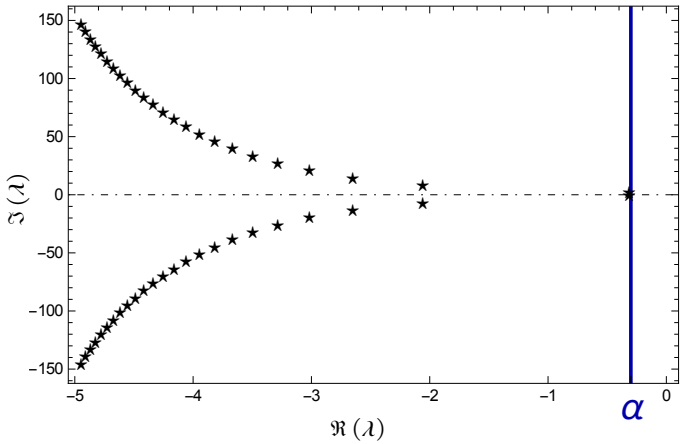


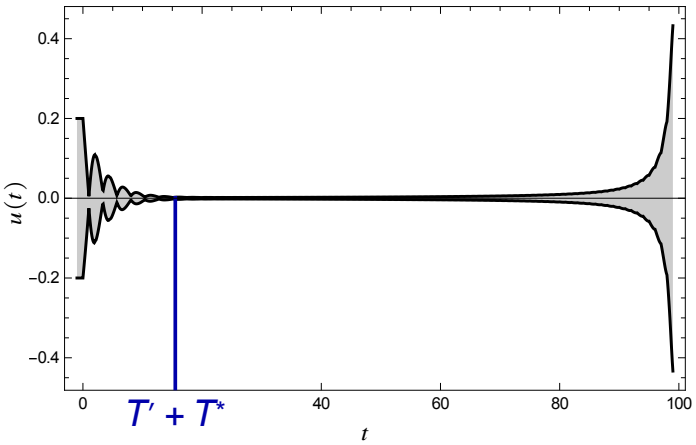


⇒ S. Feng, M. Chen, N. Zhan, M. Fränzle, B. Xue : *Taming delays in dyn. syst. : Unbounded verif. of DDEs*. CAV'19.

Method III : Unbounded Verification Leveraging Stability Criteria

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

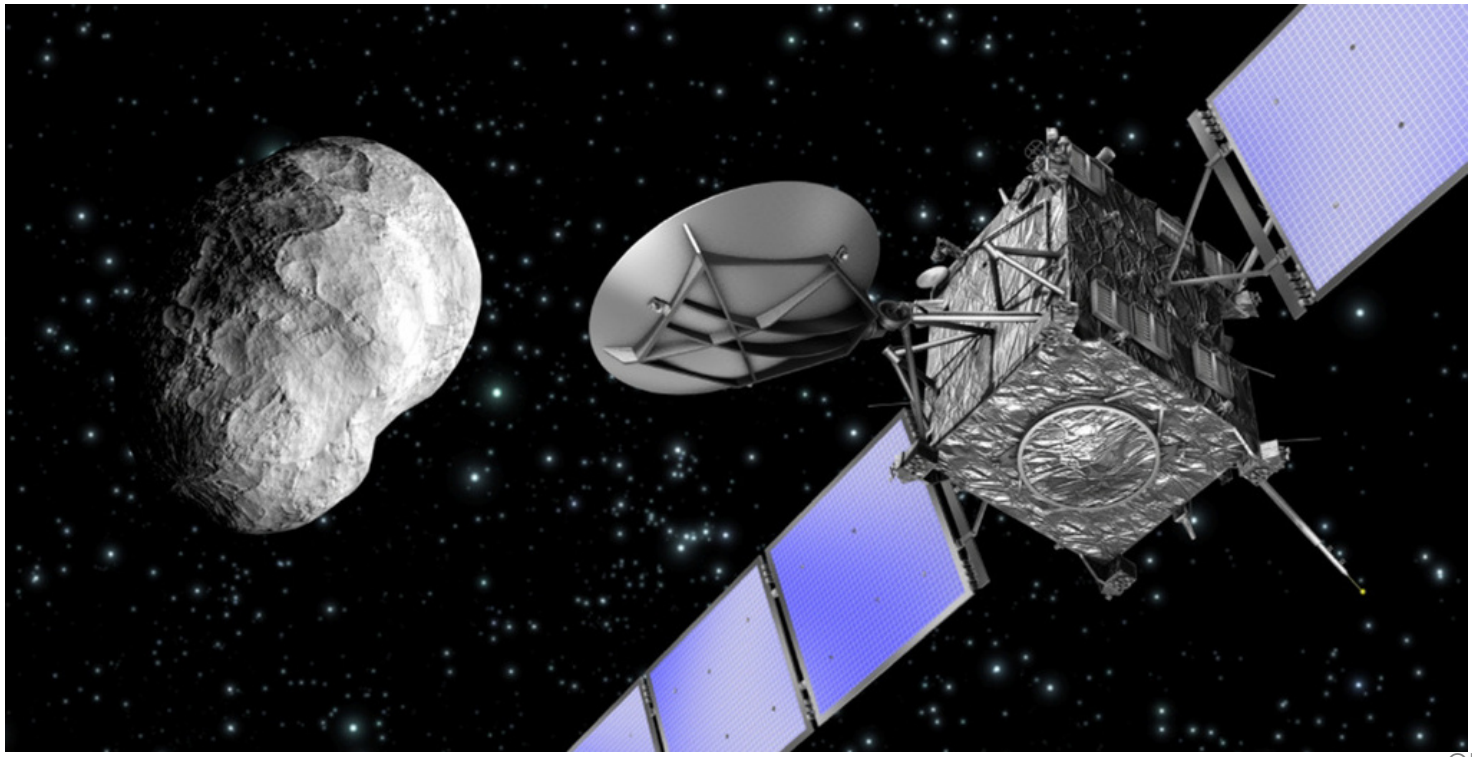
Discrete Safety Games

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

—Joint work with M. Fränzle, Y. Li, P. N. Mosaad and N. Zhan—

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.

A Robot-Escaping Game

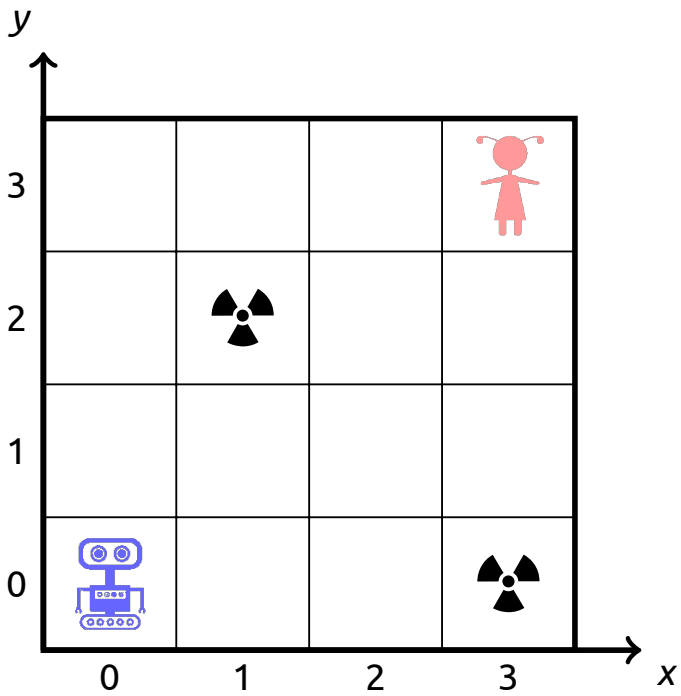


Figure – A robot escape game in a 4×4 room, with $\Sigma_r = \{\mathtt{RU}, \mathtt{UR}, \mathtt{LU}, \mathtt{UL}, \mathtt{RD}, \mathtt{DR}, \mathtt{LD}, \mathtt{DL}, \epsilon\}$, $\Sigma_k = \{\mathtt{R}, \mathtt{L}, \mathtt{U}, \mathtt{D}\}$.

A Robot-Escaping Game

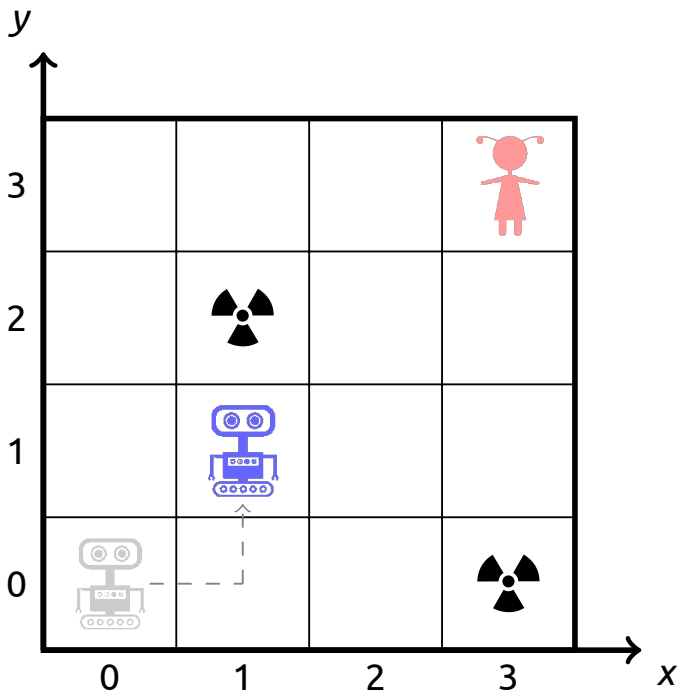


Figure – A robot escape game in a 4×4 room, with $\Sigma_r = \{\mathtt{RU}, \mathtt{UR}, \mathtt{LU}, \mathtt{UL}, \mathtt{RD}, \mathtt{DR}, \mathtt{LD}, \mathtt{DL}, \epsilon\}$, $\Sigma_k = \{\mathtt{R}, \mathtt{L}, \mathtt{U}, \mathtt{D}\}$.

A Robot-Escaping Game

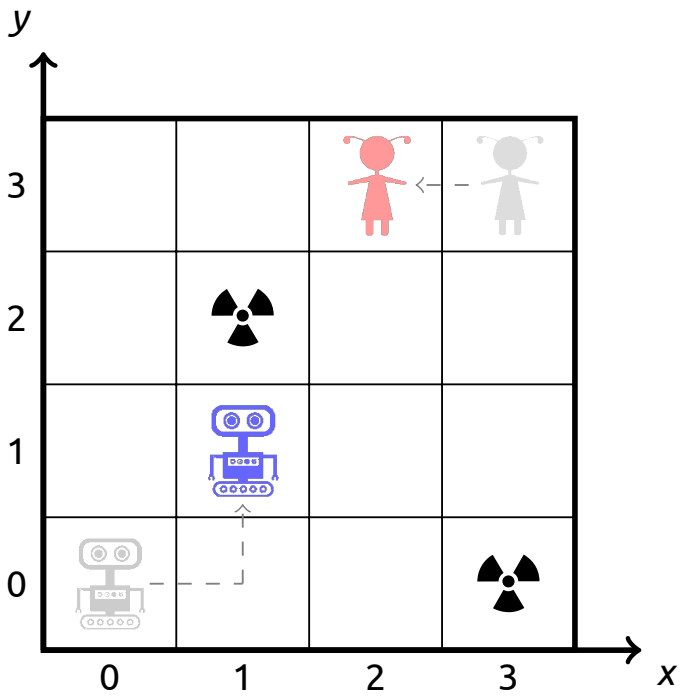


Figure – A robot escape game in a 4×4 room, with $\Sigma_r = \{\mathtt{RU}, \mathtt{UR}, \mathtt{LU}, \mathtt{UL}, \mathtt{RD}, \mathtt{DR}, \mathtt{LD}, \mathtt{DL}, \epsilon\}$, $\Sigma_k = \{\mathtt{R}, \mathtt{L}, \mathtt{U}, \mathtt{D}\}$.

A Robot-Escaping Game

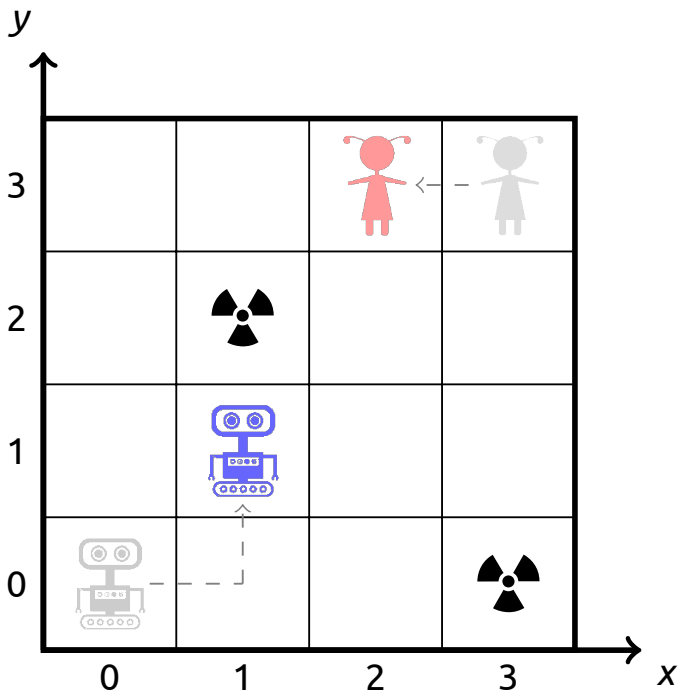


Figure – A robot escape game in a 4×4 room, with $\Sigma_r = \{\mathtt{RU}, \mathtt{UR}, \mathtt{LU}, \mathtt{UL}, \mathtt{RD}, \mathtt{DR}, \mathtt{LD}, \mathtt{DL}, \epsilon\}$, $\Sigma_k = \{\mathtt{R}, \mathtt{L}, \mathtt{U}, \mathtt{D}\}$.

No delay :

A Robot-Escaping Game

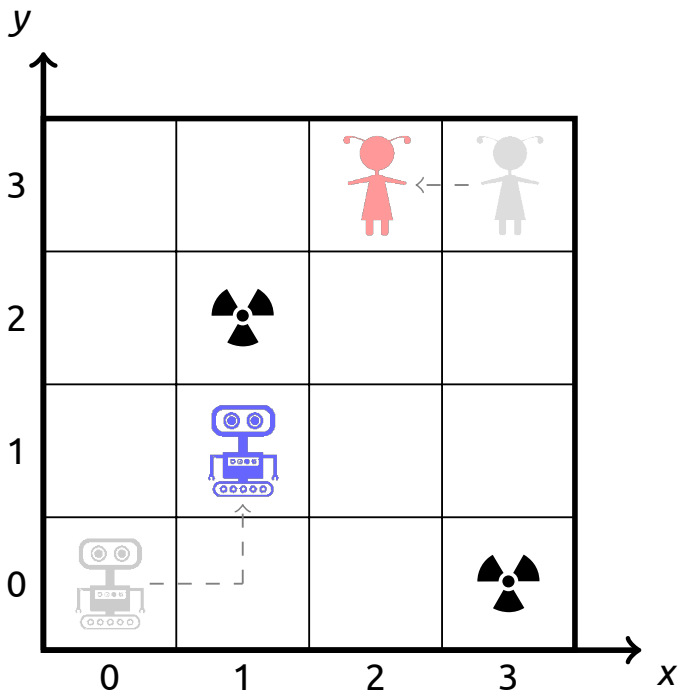


Figure – A robot escape game in a 4×4 room, with $\Sigma_r = \{\mathtt{RU}, \mathtt{UR}, \mathtt{LU}, \mathtt{UL}, \mathtt{RD}, \mathtt{DR}, \mathtt{LD}, \mathtt{DL}, \epsilon\}$, $\Sigma_k = \{\mathtt{R}, \mathtt{L}, \mathtt{U}, \mathtt{D}\}$.

No delay :

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

A Robot-Escaping Game

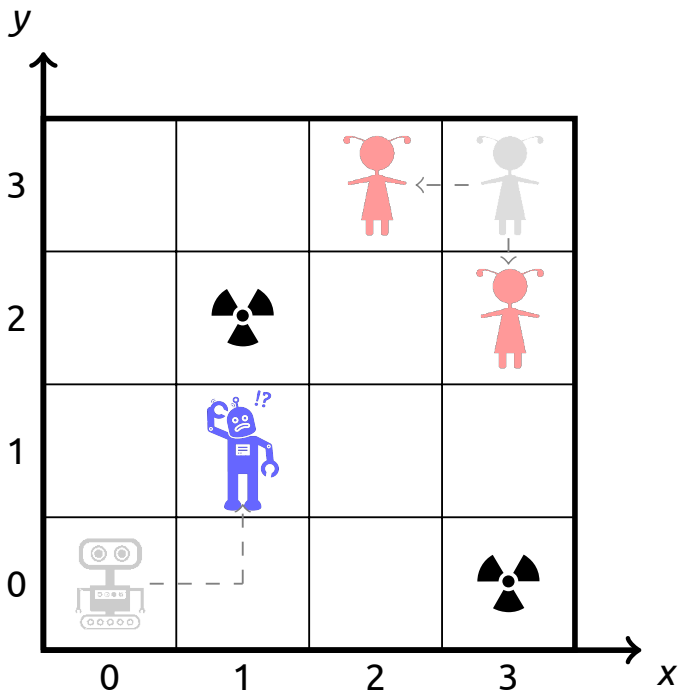


Figure – A robot escape game in a 4×4 room, with $\Sigma_r = \{\mathtt{RU}, \mathtt{UR}, \mathtt{LU}, \mathtt{UL}, \mathtt{RD}, \mathtt{DR}, \mathtt{LD}, \mathtt{DL}, \epsilon\}$, $\Sigma_k = \{\mathtt{R}, \mathtt{L}, \mathtt{U}, \mathtt{D}\}$.

No delay :

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

1 step delay :

A Robot-Escaping Game

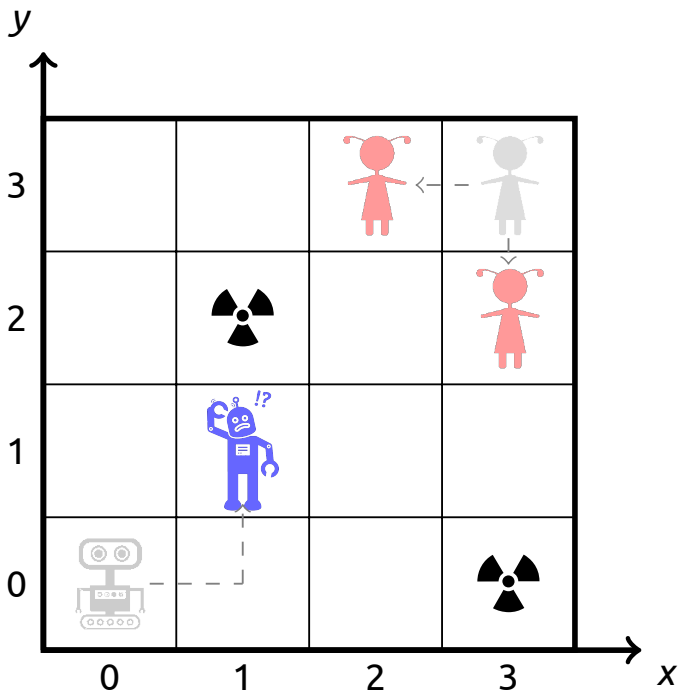


Figure – A robot escape game in a 4×4 room, with $\Sigma_r = \{\mathtt{RU}, \mathtt{UR}, \mathtt{LU}, \mathtt{UL}, \mathtt{RD}, \mathtt{DR}, \mathtt{LD}, \mathtt{DL}, \epsilon\}$, $\Sigma_k = \{\mathtt{R}, \mathtt{L}, \mathtt{U}, \mathtt{D}\}$.

No delay :

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

1 step delay :

Robot wins by 1-step pre-decision.

A Robot-Escaping Game

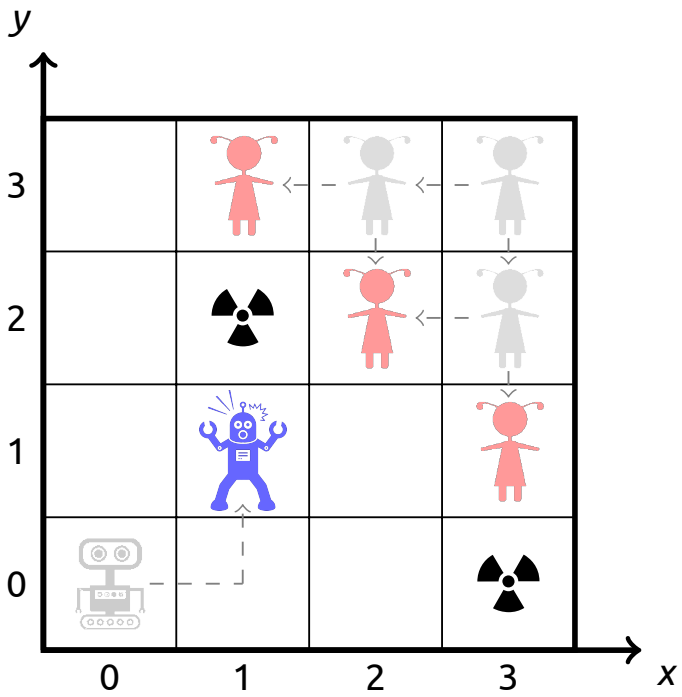


Figure – A robot escape game in a 4×4 room, with $\Sigma_r = \{\mathtt{RU}, \mathtt{UR}, \mathtt{LU}, \mathtt{UL}, \mathtt{RD}, \mathtt{DR}, \mathtt{LD}, \mathtt{DL}, \epsilon\}$, $\Sigma_k = \{\mathtt{R}, \mathtt{L}, \mathtt{U}, \mathtt{D}\}$.

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 :

A Robot-Escaping Game

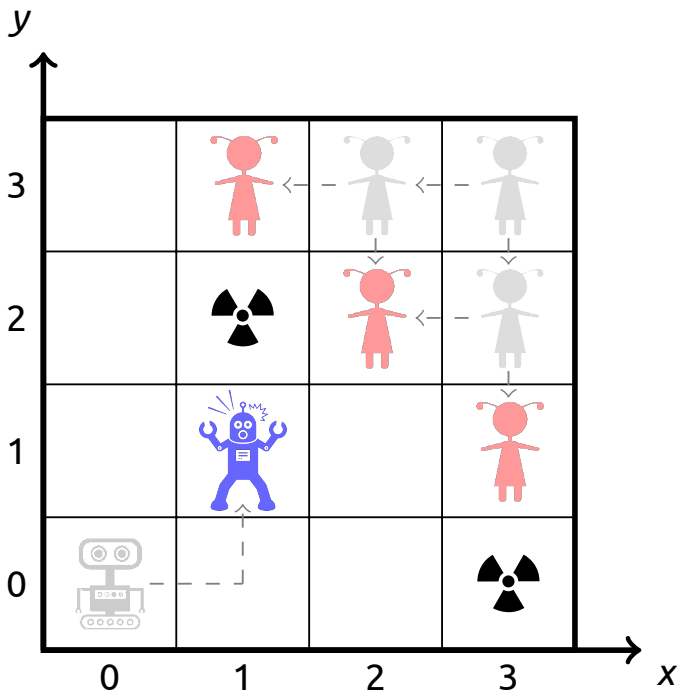


Figure – A robot escape game in a 4×4 room, with $\Sigma_r = \{\mathtt{RU}, \mathtt{UR}, \mathtt{LU}, \mathtt{UL}, \mathtt{RD}, \mathtt{DR}, \mathtt{LD}, \mathtt{DL}, \epsilon\}$, $\Sigma_k = \{\mathtt{R}, \mathtt{L}, \mathtt{U}, \mathtt{D}\}$.

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.

A Robot-Escaping Game

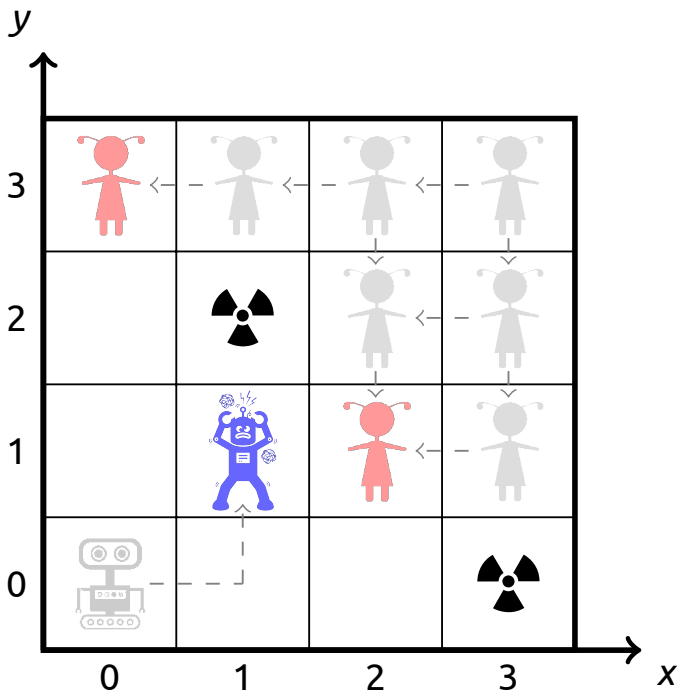


Figure – A robot escape game in a 4×4 room, with $\Sigma_r = \{\mathtt{RU}, \mathtt{UR}, \mathtt{LU}, \mathtt{UL}, \mathtt{RD}, \mathtt{DR}, \mathtt{LD}, \mathtt{DL}, \epsilon\}$, $\Sigma_k = \{\mathtt{R}, \mathtt{L}, \mathtt{U}, \mathtt{D}\}$.

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 :

A Robot-Escaping Game

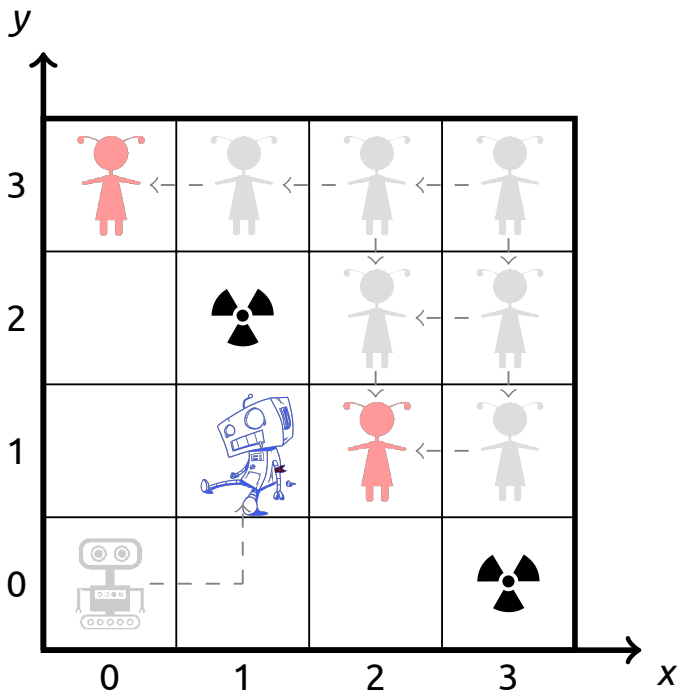


Figure – A robot escape game in a 4×4 room, with $\Sigma_r = \{\mathtt{RU}, \mathtt{UR}, \mathtt{LU}, \mathtt{UL}, \mathtt{RD}, \mathtt{DR}, \mathtt{LD}, \mathtt{DL}, \epsilon\}$, $\Sigma_k = \{\mathtt{R}, \mathtt{L}, \mathtt{U}, \mathtt{D}\}$.

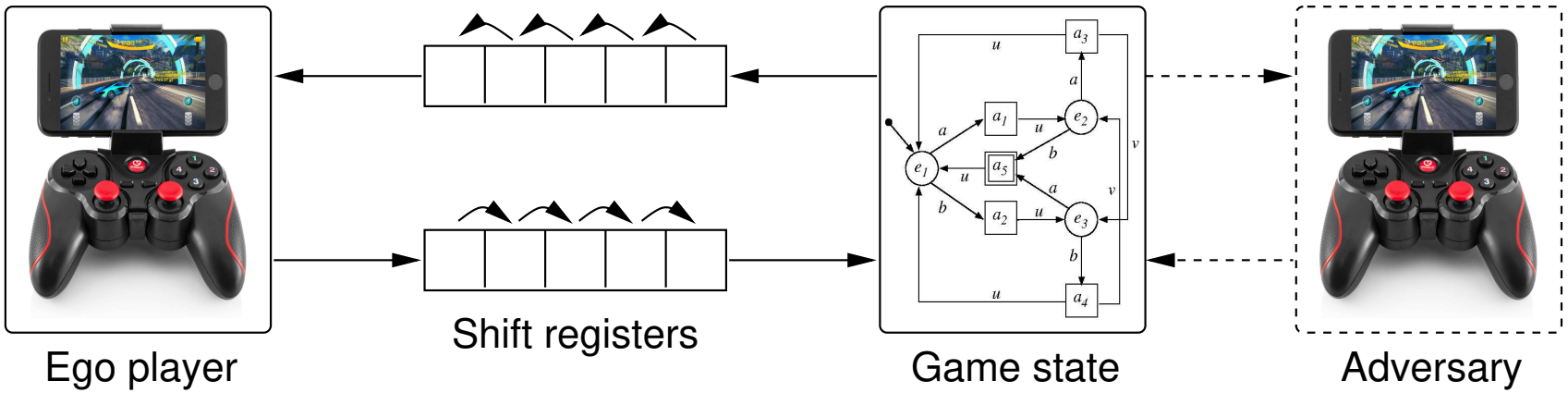
No delay :
Robot always wins by circling around the obstacle at (1,2).

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

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

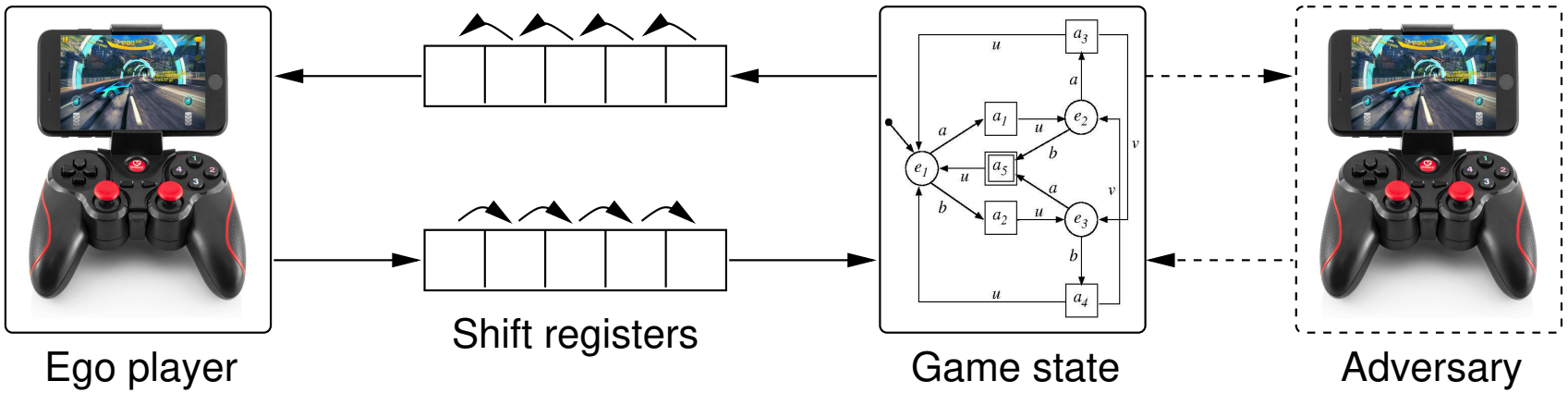
3 steps delay :
Robot is unwinnable (uncontrollable) anymore.

Playing Safety Game Subject to Discrete Delay



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

Playing Safety Game Subject to Discrete Delay



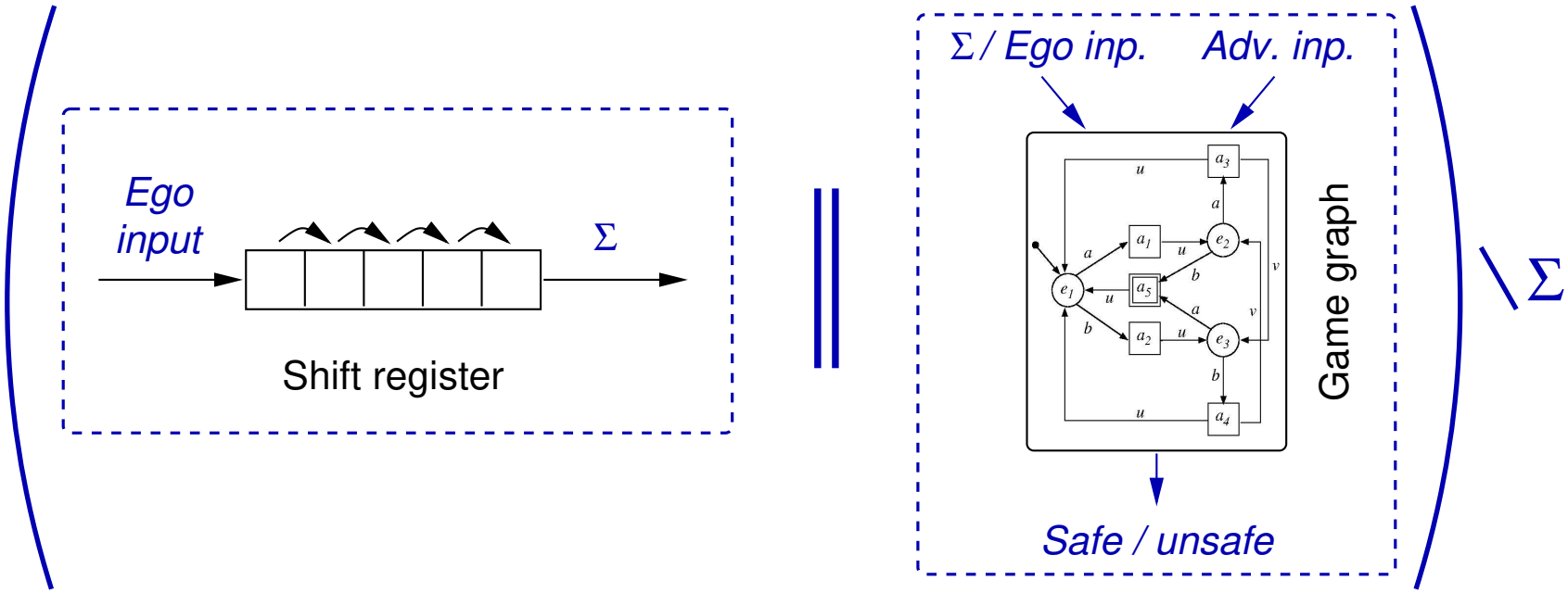
Observation : It doesn't make an observable difference for the joint dynamics whether delay occurs in perception, actuation, or both.

Consequence : There is an[1] 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!*

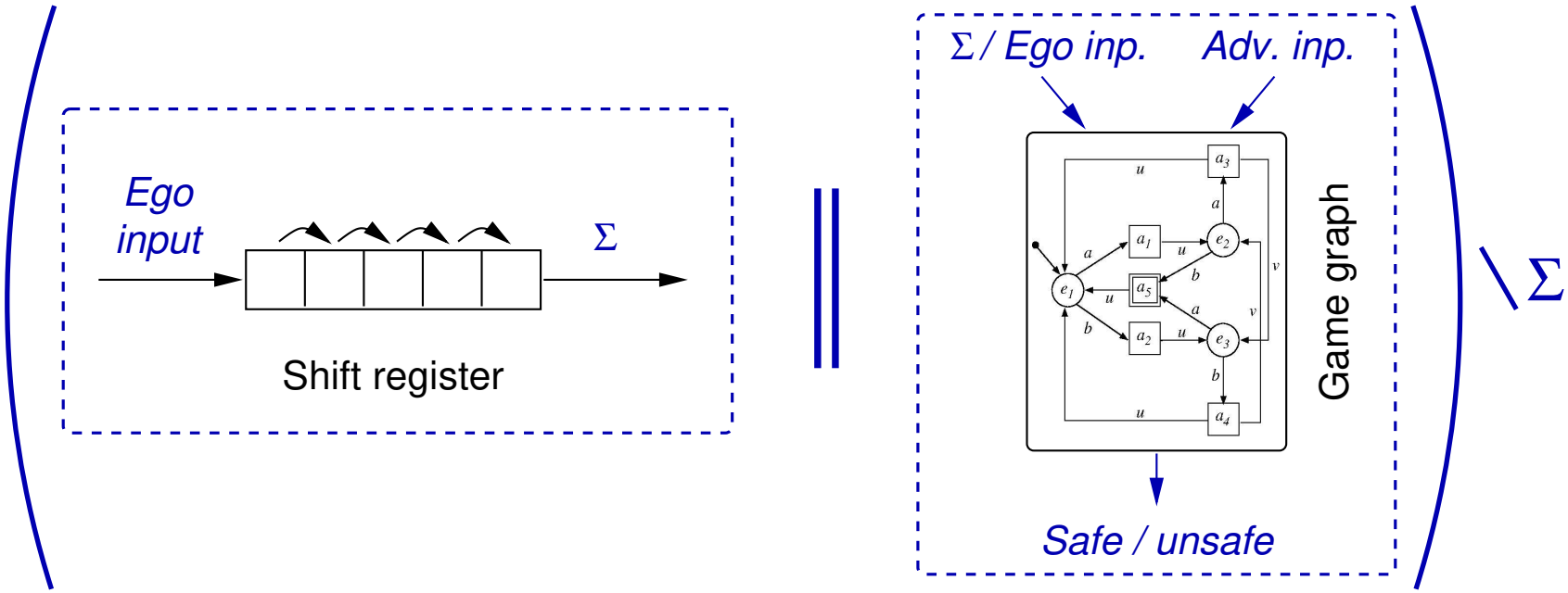
Reduction to Delay-Free Games

from Ego-Player Perspective



Reduction to Delay-Free Games

from Ego-Player Perspective



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

Incremental Synthesis

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

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

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 :

⇒ 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

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 underlying delay-free safety game ;

⇒ 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

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 underlying delay-free safety game ;
2. For each winning state, lift strategy from delay k to $k + 1$;

⇒ 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

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 underlying delay-free safety game;
2. For each winning state, lift strategy from delay k to $k+1$;
3. Remove states where this does not succeed;

⇒ 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

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 $k' > k$.

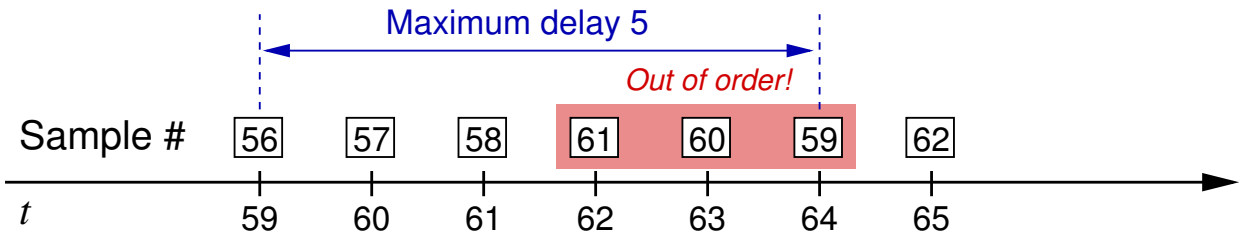
Idea : Incrementally filter out loss states &
incrementally synthesize winning strategy for the remaining :

1. Synthesize winning strategy for underlying delay-free safety game;
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].

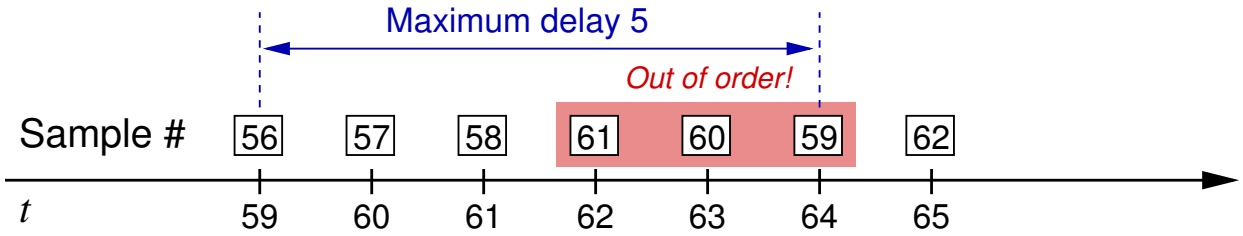
How about Non-Order-Preserving Delays?

☹ Observations may arrive out-of-order :

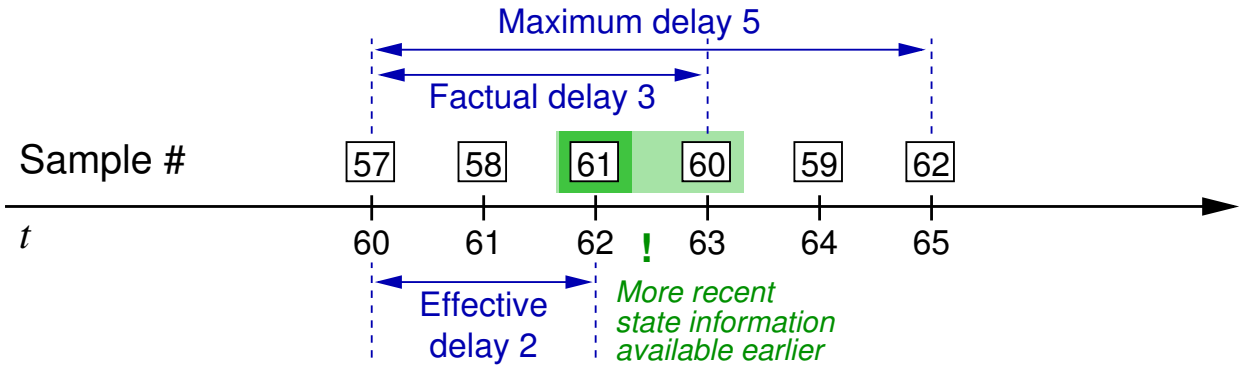


How about Non-Order-Preserving Delays?

☹ Observations may arrive out-of-order :

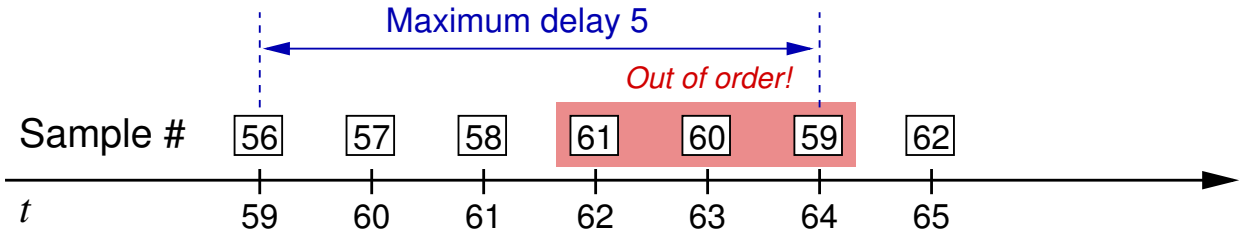


☺ But this may only reduce effective delay, improving controllability :

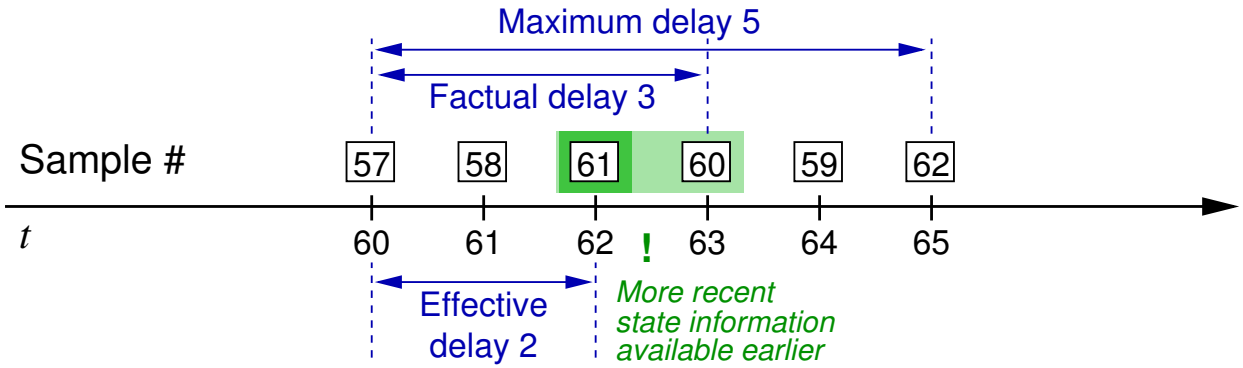


How about Non-Order-Preserving Delays?

☹ Observations may arrive out-of-order :



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

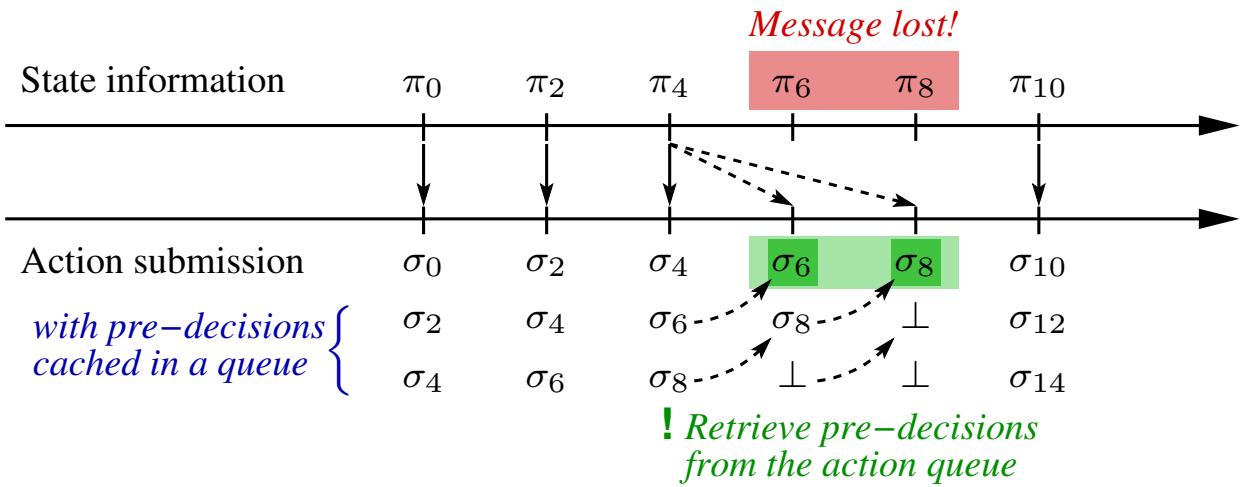


☺ W.r.t. qualitative controllability, the worst-case of out-of-order delivery is equivalent to order-preserving delay *k*.

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

How About (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

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 $\frac{\delta}{2}$.*

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.* Under review.

Publications

Verification & synthesis of timed-delayed dynamical systems :

1. Shenghua Feng, Mingshuai Chen, Naijun Zhan, Martin Fränzle, and Bai Xue. *Taming delays in dynamical systems : Unbounded verification of delay differential equations*. To appear in Proc. of **CAV 2019**.
2. Mingshuai Chen, Martin Fränzle, Yangjia Li, Peter N. Mosaad, and Naijun Zhan. *What's to come is still unsure : Synthesizing controllers resilient to delayed interaction*. In Proc. of **ATVA 2018**, LNCS 11138, pp.56-74.
3. Bai Xue, Peter N. Mosaad, Martin Fränzle, Mingshuai Chen, Yangjia Li, and Naijun Zhan. *Safe over- and under-approximation of reachable sets for delay differential equations*. In Proc. of **FORMATS 2017**, LNCS 10419, pp.281-299.
4. Mingshuai Chen, Martin Fränzle, Yangjia Li, Peter N. Mosaad, and Naijun Zhan. *Validated simulation-based verification of delayed differential dynamics*. In Proc. of **FM 2016**, LNCS 9995, pp.137-154.

Publications

Verification & synthesis of timed-delayed dynamical systems :

1. Shenghua Feng, Mingshuai Chen, Naijun Zhan, Martin Fränzle, and Bai Xue. *Taming delays in dynamical systems : Unbounded verification of delay differential equations*. To appear in Proc. of **CAV 2019**.
2. Mingshuai Chen, Martin Fränzle, Yangjia Li, Peter N. Mosaad, and Naijun Zhan. *What's to come is still unsure : Synthesizing controllers resilient to delayed interaction*. In Proc. of **ATVA 2018**, LNCS 11138, pp.56-74.
3. Bai Xue, Peter N. Mosaad, Martin Fränzle, Mingshuai Chen, Yangjia Li, and Naijun Zhan. *Safe over- and under-approximation of reachable sets for delay differential equations*. In Proc. of **FORMATS 2017**, LNCS 10419, pp.281-299.
4. Mingshuai Chen, Martin Fränzle, Yangjia Li, Peter N. Mosaad, and Naijun Zhan. *Validated simulation-based verification of delayed differential dynamics*. In Proc. of **FM 2016**, LNCS 9995, pp.137-154.

Decidability of the reachability for solvable dynamical systems :

5. Martin Fränzle, Mingshuai Chen, and Paul Kröger. *In memory of Oded Maler : Automatic reachability analysis of hybrid-state automata*. **ACM SIGLOG News**, 6(1) :19–39, 2019.
6. Ting Gan, Mingshuai Chen, Yangjia Li, Bican Xia, and Naijun Zhan. *Reachability analysis for solvable dynamical systems*. **IEEE Trans. Automat. Contr.**, 63(7) :2003–2018, 2018.
7. Ting Gan, Mingshuai Chen, Yangjia Li, Bican Xia, and Naijun Zhan. *Computing reachable sets of linear vector fields revisited*. In Proc. of **ECC 2016**, IEEE-Xplore, pp.419-426.
8. Ting Gan, Mingshuai Chen, Liyun Dai, Bican Xia, and Naijun Zhan. *Decidability of the reachability for a family of linear vector fields*. In Proc. of **ATVA 2015**, LNCS 9364, pp.482-499.

Publications

Discovering nonlinear interpolants :

9 Mingshuai Chen, Jian Wang, Jie An, Bohua Zhan, Deepak Kapur, and Naijun Zhan. *NIL : Learning nonlinear interpolants*. To appear in Proc. of **CADE 2019**.

10 Ting Gan, Liyun Dai, Naijun Zhan, Deepak Kapur, and Mingshuai Chen. *Interpolant synthesis for quadratic polynomial inequalities and combination with EUF*. In Proc. of **IJCAR 2016**, LNCS 9706, pp.195-212.

Publications

Discovering nonlinear interpolants :

9 Mingshuai Chen, Jian Wang, Jie An, Bohua Zhan, Deepak Kapur, and Naijun Zhan. *NIL : Learning nonlinear interpolants*. To appear in Proc. of **CADE 2019**.

10 Ting Gan, Liyun Dai, Naijun Zhan, Deepak Kapur, and Mingshuai Chen. *Interpolant synthesis for quadratic polynomial inequalities and combination with EUF*. In Proc. of **IJCAR 2016**, LNCS 9706, pp.195-212.

Modelling, analysis & verification of hybrid systems :

11 Mingshuai Chen, Anders P. Ravn, Shuling Wang, Mengfei Yang, and Naijun Zhan. *A two-way path between formal and informal design of embedded systems*. In Proc. of **UTP 2016**, LNCS 10134, pp.65-92.

12 Mingshuai Chen, Xiao Han, Tao Tang, Shuling Wang, Mengfei Yang, Naijun Zhan, Hengjun Zhao, and Liang Zou. *MARS : A toolchain for modelling, analysis and verification of hybrid systems*. In **ProCoS 2015**, NASA Monographs in Systems and Software Engineering, pp.39-58.

On-Going Work

13 Mingshuai Chen, Martin Fränzle, Yangjia Li, Peter N. Mosaad, and Naijun Zhan. *Indecision and delays are the parents of failure : Taming them algorithmically by synthesizing delay-resilient control*. Under review.

14 Jie An, Mingshuai Chen, Bohua Zhan, Naijun Zhan, and Miaomiao Zhang. *Learning one-clock timed automata*. Under review.

15 Bai Xue, Martin Fränzle, Hengjun Zhao, Mingshuai Chen, Naijun Zhan, and Arvind Easwaran. *Safety verification of stochastic multi-layer perceptrons*. Under review.

16 Jian Wang, Jie An, Mingshuai Chen, Naijun Zhan, Lulin Wang, and Miaomiao Zhang. *From model to implementation : A network-algorithm programming language*. Under review.

17 Yangjia Li, Hui Lu, Naijun Zhan, Mingshuai Chen, and Guohua Wu. *Termination analysis of polynomial programs with equality conditions*. Under revision.

18 Mingshuai Chen, Ting Gan, Deepak Kapur, Bican Xia, Naijun Zhan, and Hanwen Zhang. *NLFIntp : A tool for synthesizing nonlinear interpolants*. Under revision.

Outline

Moving forward ...

1. Combining and extending all the stuff in either continuous or discrete dynamics to a hybrid setting. Mathematical model for "delayed hybrid systems"?
2. The HJB formulation of reachability : exact description of the reachable set, natural extension to differential games. Symbolic/numerical methods for solving PDEs beyond finite-dimensional Euclidean space?
3. Real-world applications : vehicle-to-vehicle communication, communication protocol, remote control, etc..

Outline

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 algorithm for efficient control synthesis,
- Equivalent controllability with cases of non-order-preserving delays and bounded message loss.

Future Work : We plan to

- carry into a hybrid setting,
- explore HJB-reachability and differential games,
- investigate real-world applications.

Snapshots of My Ph.D.

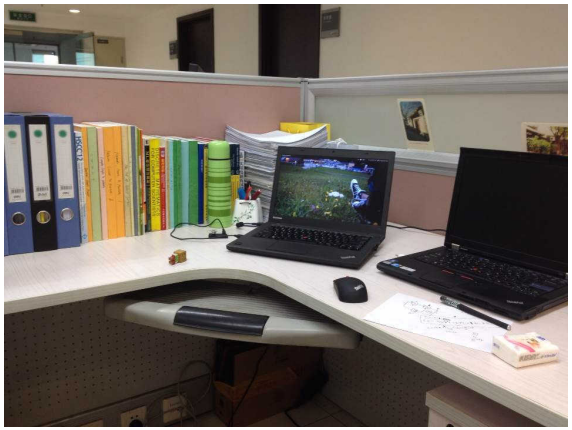


Figure – First day onboard.

Snapshots of My Ph.D.

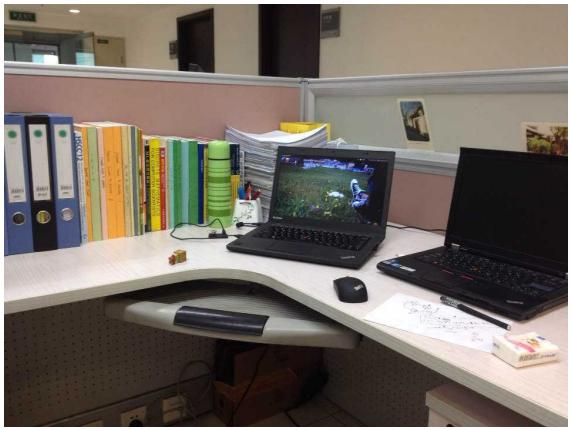


Figure – First day onboard.

⇒

⇒

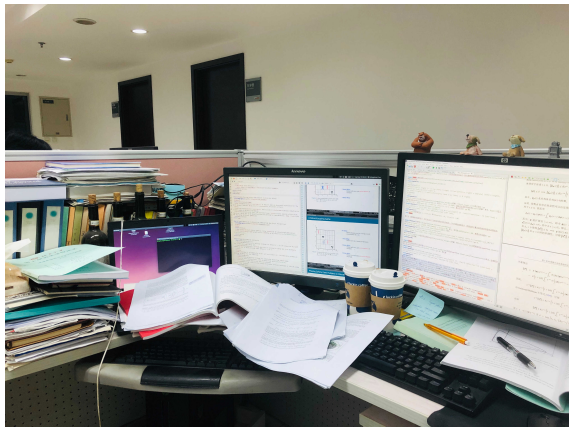


Figure – Last day finishing the thesis.

Snapshots of My Ph.D.

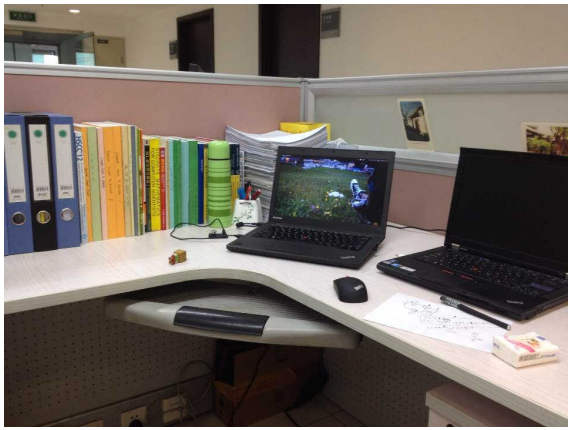


Figure – First day onboard.

⇒

Figure – My Ph.D. life in between.

⇒

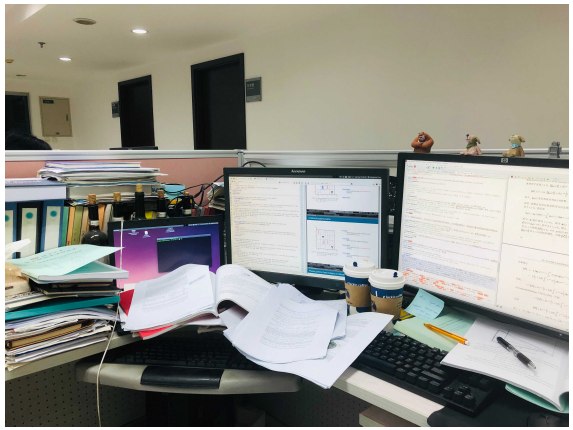


Figure – Last day finishing the thesis.

Concluding Remarks

Concluding Remarks

Whether I Could Now Finish My Ph.D.? Hopefully No Delays Ahead …

Thank You — Q & A?