Taming Delays in Cyber-Physical Systems

Towards a Theory of Networked Hybrid Systems

Naijun Zhan[1], Mingshuai Chen[2]

[1]Institute of Software, Chinese Academy of Sciences
[2]Lehrstuhl für Informatik 2, RWTH Aachen University

HTD-Tutorial · Houston · December 2020

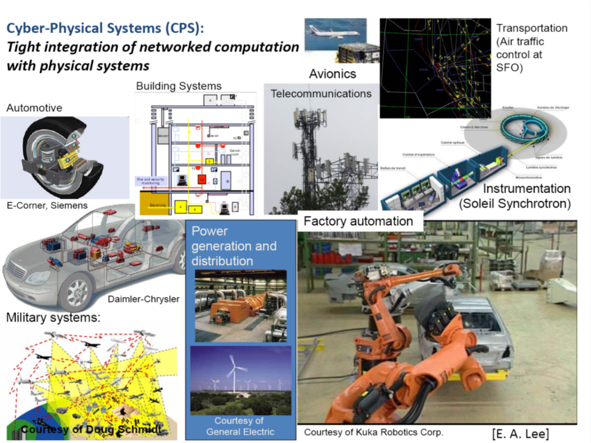
Cyber-Physical Systems

"The term 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, The Impact of Control Technology, 2011]

Cyber-Physical Systems

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



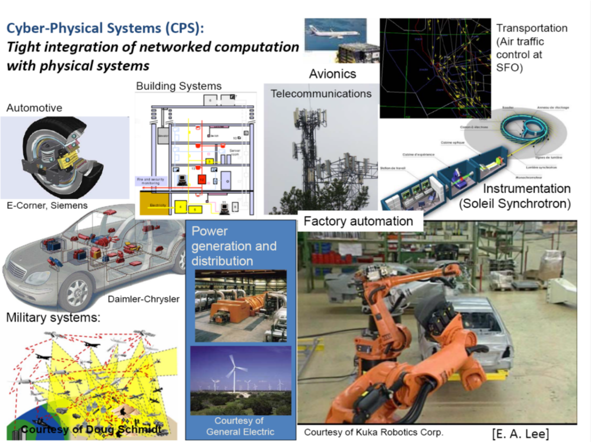
Cyber-Physical Systems (CPS):
Tight integration of networked computation with physical systems

Avionics
Transportation (Air traffic control at SFO)
Building Systems
Telecommunications
Automotive
E-Corner, Siemens
Instrumentation (Soleil Synchrotron)
Daimler-Chrysler
Power generation and distribution
Factory automation
Military systems:
Courtesy of Doug Schmidt
Courtesy of General Electric
Courtesy of Kuka Robotics Corp.
[E. A. Lee]

©S. A. Seshia, 2017

Cyber-Physical Systems

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

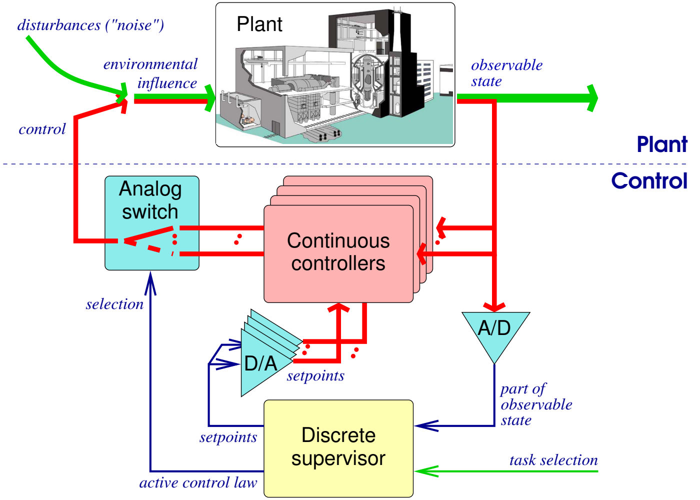

Cyber-Physical Systems (CPS):
Tight integration of networked computation with physical systems
Avionics
Transportation (Air traffic control at SFO)
Building Systems
Telecommunications
Automotive
E-Corner, Siemens
Instrumentation (Soleil Synchrotron)
Factory automation
Power generation and distribution
Daimler-Chrysler
Military systems:
Courtesy of Doug Schmidt
Courtesy of General Electric
Courtesy of Kuka Robotics Corp.
[E. A. Lee]

©S. A. Seshia, 2017

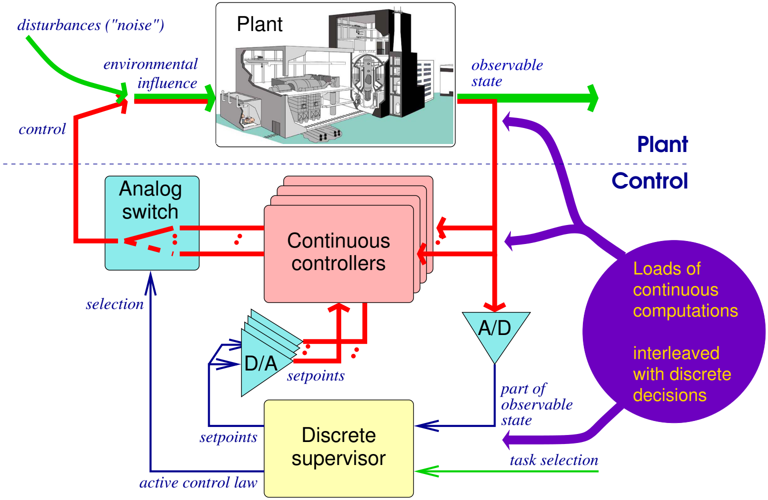
"How can we provide people with CPS they can bet their lives on?"

[Jeannette Wing]

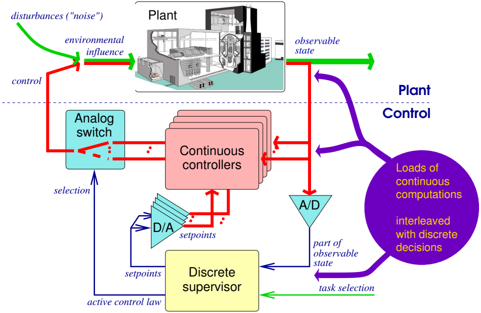
Hybrid Systems



Hybrid Systems



Hybrid Systems



Crucial question :

- How do the controller and the plant interact?

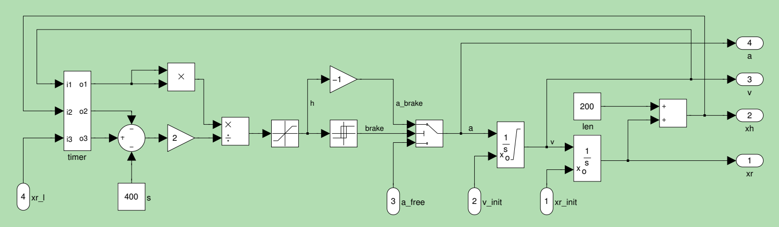
Traditional answer :

- 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

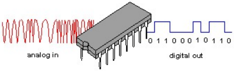
Following the tradition, above (rather typical) Simulink model assumes

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

Central questions :

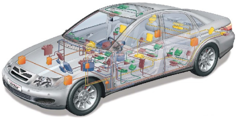
1. Is this realistic?
2. If not, does it have observable effect on control performance?
3. May that effect be detrimental or even harmful?

Q1 : Is Instantaneous Coupling Realistic?



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

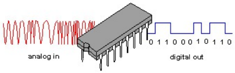
Digital signal processing, especially in complex sensors like CV, needs processing time, adding signal delays.



Networked control introduces communication latency into the feedback control loop.

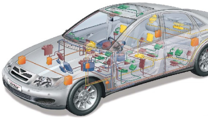
Harvesting, fusing, and forwarding data through sensor networks enlarge the latter by orders of magnitude.

Q1 : Is Instantaneous Coupling Realistic? – No.



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

Digital signal processin[illegible]specially in complex sensors like CV[illegible]g time, adding signal de-l[illegible]



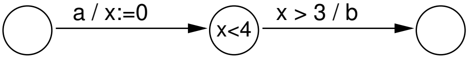
Networked control introduces communication latency in feedback control loop.

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. :



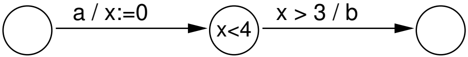
- Thus amenable to the pertinent analysis tools.

⇒ 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.

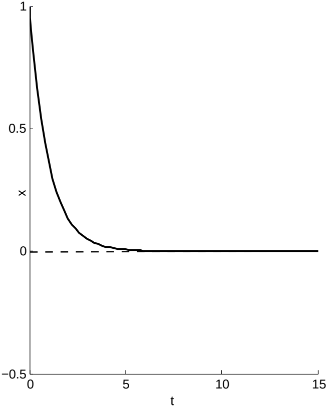
Network delay : Information of different age coexists and is queuing in the network when piped towards target.

- 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.

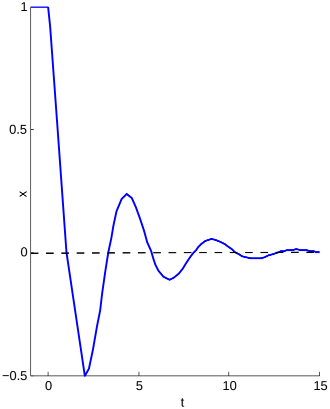
⇒ Our theme today.

Q2 : Do Delays Have Observable Effect?

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



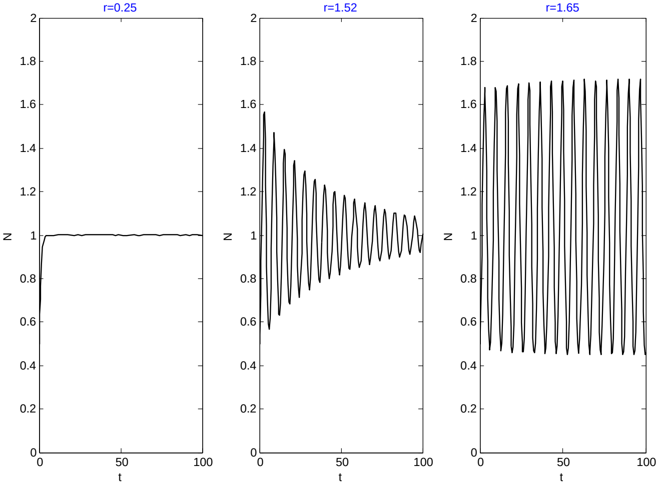
$$\begin{cases} \dot{x}(t) = -x(t{\color{blue}-1}) \\ x([-1,0]) \equiv 1 \end{cases}$$



Q2 : Do Delays Have Observable Effect?

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

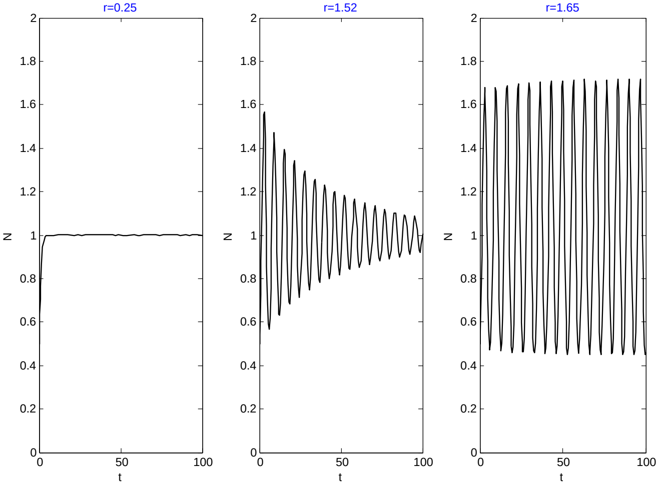
$$\dot{N}(t) = N(t)[1 - N(t - r)]$$



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

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

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



Q3 : May the Effects be Harmful?

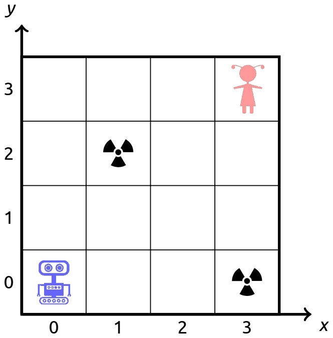


Figure – A robot escape game in a 4×4 room, with
$\Sigma_r = \{\mathtt{RU}, \mathtt{UR}, \mathtt{LU}, \mathtt{UL}, \mathtt{RD}, \mathtt{DR}, \mathtt{LD}, \mathtt{DL}, \epsilon\}$,
$\Sigma_k = \{\mathtt{R}, \mathtt{L}, \mathtt{U}, \mathtt{D}\}$.

Q3 : May the Effects be Harmful?

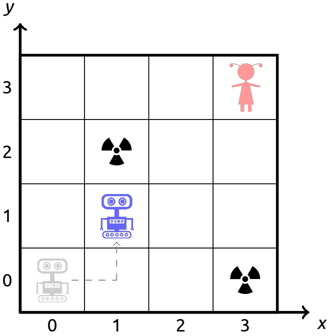


Figure – A robot escape game in a 4 × 4 room, with
$\Sigma_r = \{\mathtt{RU}, \mathtt{UR}, \mathtt{LU}, \mathtt{UL}, \mathtt{RD}, \mathtt{DR}, \mathtt{LD}, \mathtt{DL}, \epsilon\}$,
$\Sigma_k = \{\mathtt{R}, \mathtt{L}, \mathtt{U}, \mathtt{D}\}$.

Q3 : May the Effects be Harmful?

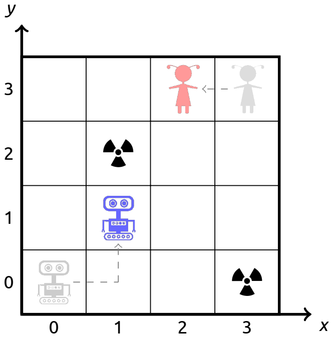


Figure – A robot escape game in a 4×4 room, with
$\Sigma_r = \{\mathtt{RU}, \mathtt{UR}, \mathtt{LU}, \mathtt{UL}, \mathtt{RD}, \mathtt{DR}, \mathtt{LD}, \mathtt{DL}, \epsilon\}$,
$\Sigma_k = \{\mathtt{R}, \mathtt{L}, \mathtt{U}, \mathtt{D}\}$.

Q3 : May the Effects be Harmful?

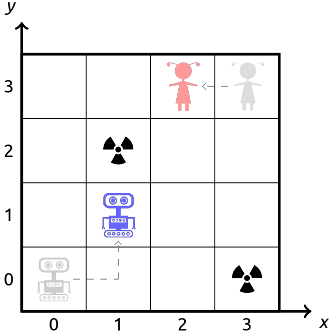


Figure – A robot escape game in a 4 × 4 room, with
$\Sigma_r = \{\mathtt{RU}, \mathtt{UR}, \mathtt{LU}, \mathtt{UL}, \mathtt{RD}, \mathtt{DR}, \mathtt{LD}, \mathtt{DL}, \epsilon\}$,
$\Sigma_k = \{\mathtt{R}, \mathtt{L}, \mathtt{U}, \mathtt{D}\}$.

No delay :

Q3 : May the Effects be Harmful?

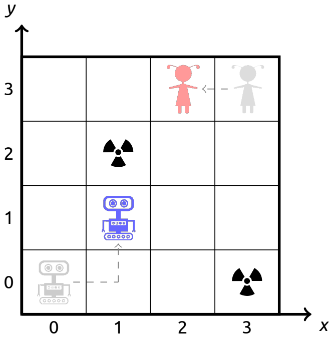


Figure – A robot escape game in a 4×4 room, with
$\Sigma_r = \{\mathtt{RU}, \mathtt{UR}, \mathtt{LU}, \mathtt{UL}, \mathtt{RD}, \mathtt{DR}, \mathtt{LD}, \mathtt{DL}, \epsilon\}$,
$\Sigma_k = \{\mathtt{R}, \mathtt{L}, \mathtt{U}, \mathtt{D}\}$.

No delay :

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

Q3 : May the Effects be Harmful?

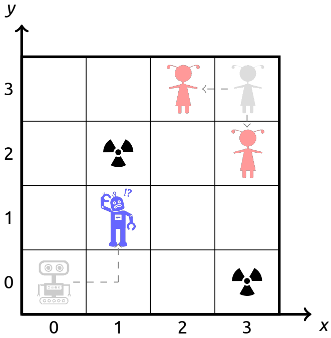


Figure – A robot escape game in a 4×4 room, with $\Sigma_r = \{\mathtt{RU}, \mathtt{UR}, \mathtt{LU}, \mathtt{UL}, \mathtt{RD}, \mathtt{DR}, \mathtt{LD}, \mathtt{DL}, \epsilon\}$, $\Sigma_k = \{\mathtt{R}, \mathtt{L}, \mathtt{U}, \mathtt{D}\}$.

No delay :

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

1 step delay :

Q3 : May the Effects be Harmful ?

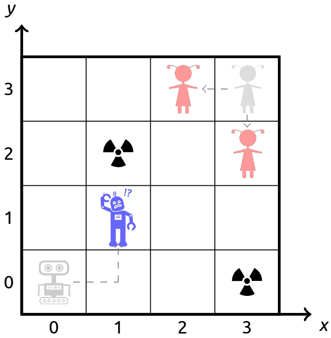


Figure – A robot escape game in a 4 × 4 room, with
$\Sigma_r = \{\mathtt{RU}, \mathtt{UR}, \mathtt{LU}, \mathtt{UL}, \mathtt{RD}, \mathtt{DR}, \mathtt{LD}, \mathtt{DL}, \epsilon\}$,
$\Sigma_k = \{\mathtt{R}, \mathtt{L}, \mathtt{U}, \mathtt{D}\}$.

No delay :

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

1 step delay :

Robot wins by 1-step pre-decision.

Q3 : May the Effects be Harmful?

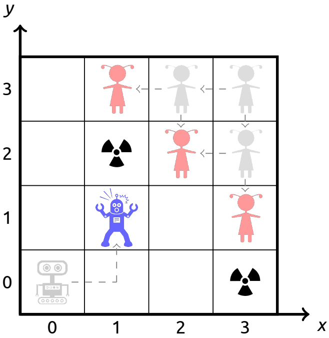


Figure – A robot escape game in a 4 × 4 room, with $\Sigma_r = \{\texttt{RU}, \texttt{UR}, \texttt{LU}, \texttt{UL}, \texttt{RD}, \texttt{DR}, \texttt{LD}, \texttt{DL}, \epsilon\}$, $\Sigma_k = \{\texttt{R}, \texttt{L}, \texttt{U}, \texttt{D}\}$.

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 :

Q3 : May the Effects be Harmful ?

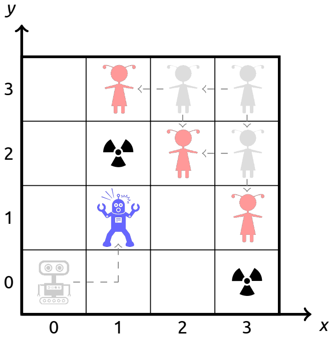


Figure – A robot escape game in a 4×4 room, with $\Sigma_r = \{\mathtt{RU}, \mathtt{UR}, \mathtt{LU}, \mathtt{UL}, \mathtt{RD}, \mathtt{DR}, \mathtt{LD}, \mathtt{DL}, \epsilon\}$, $\Sigma_k = \{\mathtt{R}, \mathtt{L}, \mathtt{U}, \mathtt{D}\}$.

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.

Q3 : May the Effects be Harmful?

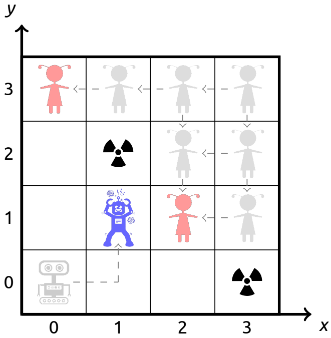


Figure – A robot escape game in a 4 × 4 room, with $\Sigma_r = \{\texttt{RU}, \texttt{UR}, \texttt{LU}, \texttt{UL}, \texttt{RD}, \texttt{DR}, \texttt{LD}, \texttt{DL}, \epsilon\}$, $\Sigma_k = \{\texttt{R}, \texttt{L}, \texttt{U}, \texttt{D}\}$.

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 :

Q3 : May the Effects be Harmful?

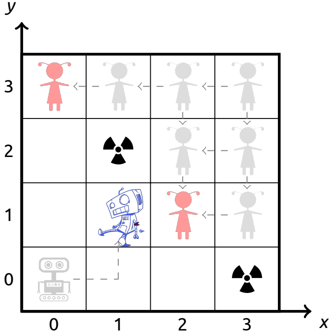


Figure – A robot escape game in a 4 × 4 room, with $\Sigma_r = \{\mathtt{RU}, \mathtt{UR}, \mathtt{LU}, \mathtt{UL}, \mathtt{RD}, \mathtt{DR}, \mathtt{LD}, \mathtt{DL}, \epsilon\}$, $\Sigma_k = \{\mathtt{R}, \mathtt{L}, \mathtt{U}, \mathtt{D}\}$.

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.

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

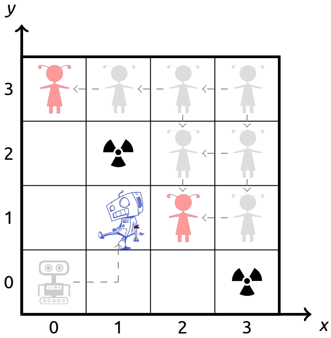


Figure – A robot escape game in a 4 × 4 room, with $\Sigma_r = \{\mathtt{RU}, \mathtt{UR}, \mathtt{LU}, \mathtt{UL}, \mathtt{RD}, \mathtt{DR}, \mathtt{LD}, \mathtt{DL}, \epsilon\}$, $\Sigma_k = \{\mathtt{R}, \mathtt{L}, \mathtt{U}, \mathtt{D}\}$.

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.

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.

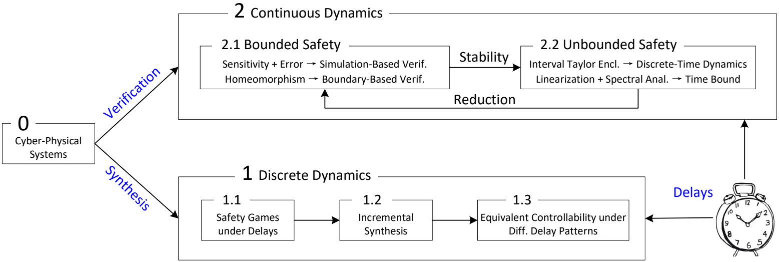
Automatic verification/synthesis methods addressing feedback delays in hybrid systems should therefore abound!

Surprisingly, they don't :

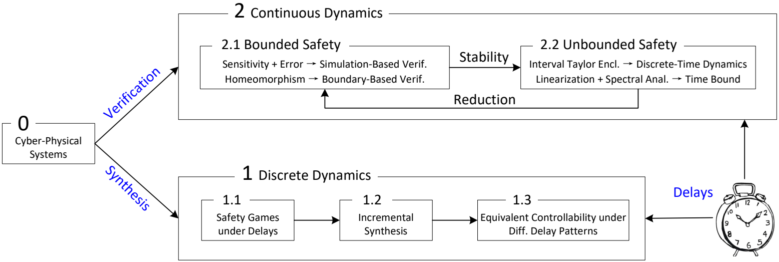
1. M. Peet, S. Lall : *Constructing Lyapunov functions for nonlinear DDEs using SDP* (NOLCOS'04)
2. S. Prajna, A. Jadbabaie : *Meth. f. safety verification of time-delay syst.* (CDC'05)
3. L. Zou, M. Fränzle, N. Zhan, P. N. Mosaad : *Autom. verific. of stabil. and safety* (CAV'15)
4. H. Trinh, P. T. 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. Zimmermann. LICS'18, GandALF'17], [F. Klein & M. Zimmermann. ICALP'15, CSL'15]

(plus a handful of related versions)

Overview of the Tutorial


2 Continuous Dynamics
2.1 Bounded Safety
Sensitivity + Error → Simulation-Based Verif.
Homeomorphism → Boundary-Based Verif.
Stability
2.2 Unbounded Safety
Interval Taylor Encl. → Discrete-Time Dynamics
Linearization + Spectral Anal. → Time Bound
Reduction
0
Cyber-Physical Systems
Verification
Synthesis
1 Discrete Dynamics
1.1
Safety Games under Delays
1.2
Incremental Synthesis
1.3
Equivalent Controllability under Diff. Delay Patterns
Delays

Overview of the Tutorial



0 Cyber-Physical Systems

Verification

Synthesis

2 Continuous Dynamics

2.1 Bounded Safety

Sensitivity + Error → Simulation-Based Verif.

Homeomorphism → Boundary-Based Verif.

Stability

2.2 Unbounded Safety

Interval Taylor Encl. → Discrete-Time Dynamics

Linearization + Spectral Anal. → Time Bound

Reduction

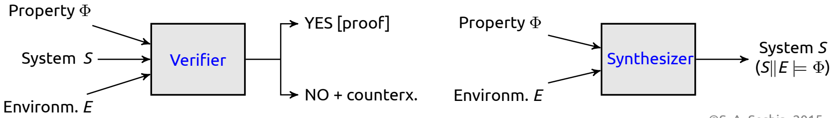
1 Discrete Dynamics

1.1 Safety Games under Delays

1.2 Incremental Synthesis

1.3 Equivalent Controllability under Diff. Delay Patterns

Delays



Property Φ

System S

Environm. E

Verifier

YES [proof]

NO + counterx.

Property Φ

Environm. E

Synthesizer

System S

$(S \| E \models \Phi)$

©S. A. Seshia, 2015

The Agenda

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

Outline

1 Synthesizing Safe Controllers Resilient to Delayed Interaction
- Safety Games under Delays
- Incremental Synthesis
- Equivalent Controllability

2 Verifying Safety of Delayed Differential Dynamics
- Delayed Differential Dynamics
- Bounded Safety Verification
- Unbounded Safety Verification

3 Concluding Remarks
- Summary

Solving Discrete Safety Games

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

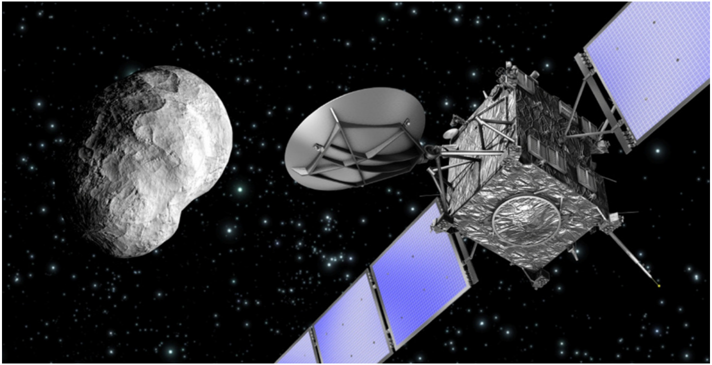
—Joint work with M. Fränzle, Y. Li, and P. N. Mosaad—

ISCAS

CARL VON OSSIETZKY universität OLDENBURG

Staying Safe

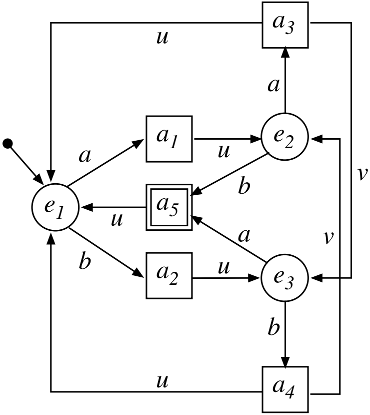
When Observation & Actuation Suffer from Serious Delays



©ESA

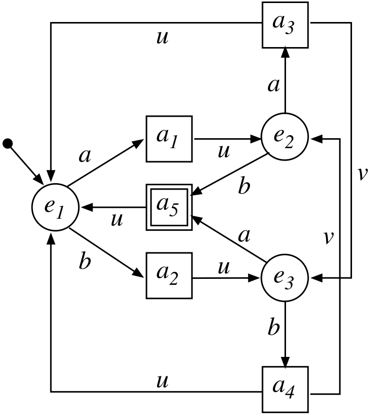
- You could move slowly. (Well, can you?)
- You could trust autonomy.
- Or you have to anticipate and issue actions early.

A Trivial Safety Game



Goal : Avoid $\boxed{\boxed{a_5}}$ by appropriate actions of player *e*.

A Trivial Safety Game

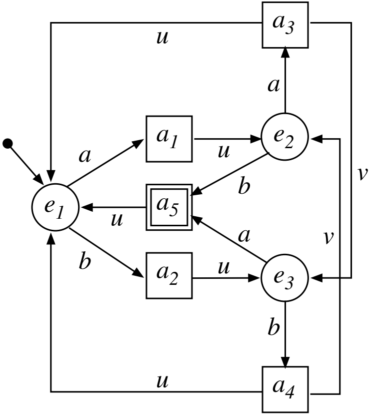


Goal : Avoid $\boxed{\boxed{a_5}}$ by appropriate actions of player *e*.

Strategy : May always play "*a*" except in e_3 :

$$e_1, e_2 \mapsto a$$
$$e_3 \mapsto b$$

A Trivial Safety Game



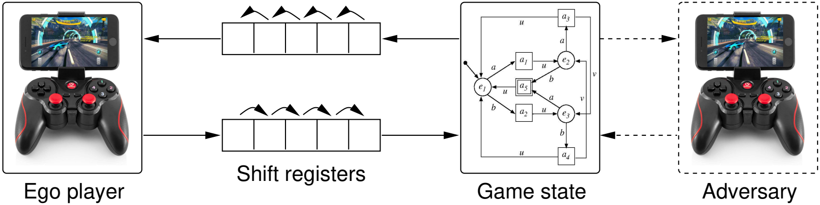
Goal : Avoid $\boxed{\boxed{a_5}}$ by appropriate actions of player *e*.

Strategy : May always play "*a*" except in e_3 :

$$e_1, e_2 \mapsto a$$
$$e_3 \mapsto b$$

Properties : Determinacy and memoryless.

Playing Safety Game Subject to Discrete Delay



Ego player

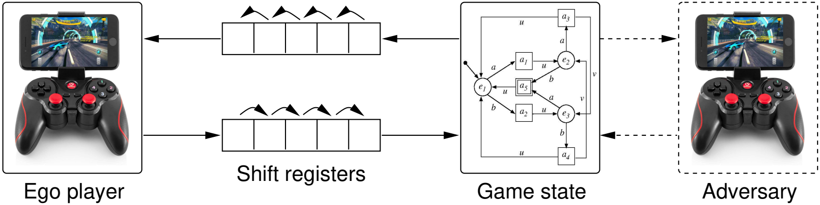
Shift registers

Game state

Adversary

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



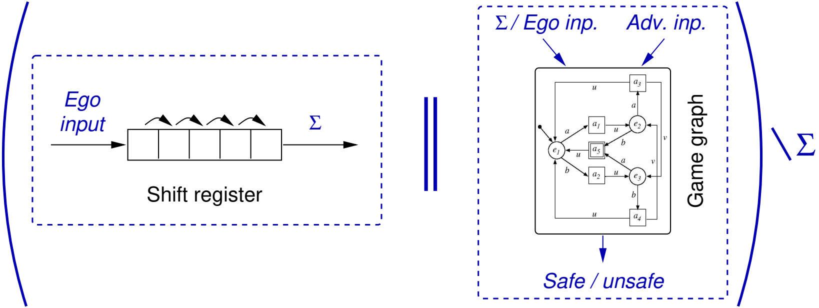
Observation : It doesn’t make an observable difference for the joint dynamics whether delay occurs in perception, actuation, or both.

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!*

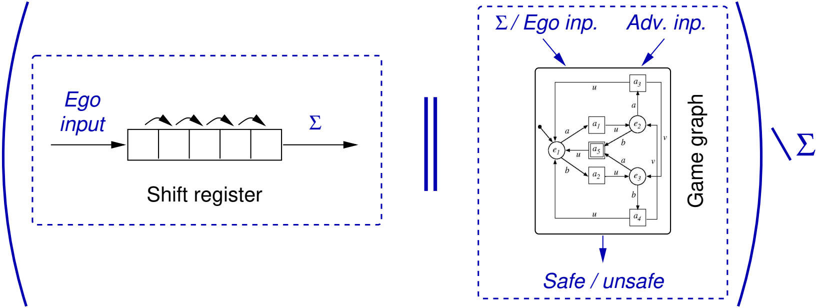
Reduction to Delay-Free Games

from Ego-Player Perspective



Reduction to Delay-Free Games

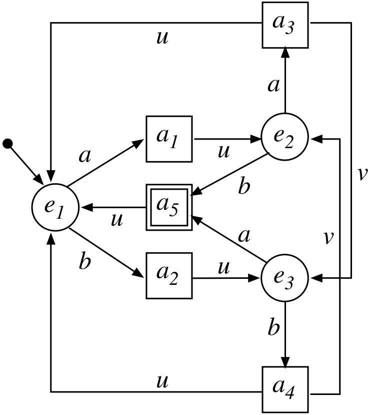
from Ego-Player Perspective



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

The Simple Safety Game

...but with Delay



No delay :

$e_1, e_2 \mapsto a$

$e_3 \mapsto b$

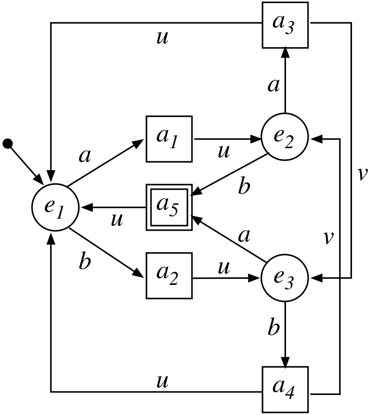
1 step delay : Strategy?

$a_1, a_4 \mapsto a$

$a_2, a_3 \mapsto b$

The Simple Safety Game

…but with Delay



No delay :

$$e_1, e_2 \mapsto a$$
$$e_3 \mapsto b$$

1 step delay : Strategy?

$$a_1, a_4 \mapsto a$$
$$a_2, a_3 \mapsto b$$

2 steps delay : Strategy?

$$e_1 \mapsto \begin{cases} a & \text{if 2 steps back an "}a\text{" was issued,} \\ b & \text{if 2 steps back a "}b\text{" was issued.} \end{cases}$$
$$e_2 \mapsto b$$
$$e_3 \mapsto a$$

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].

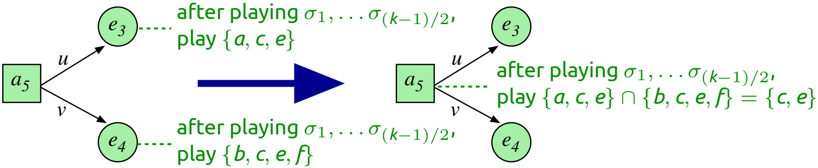
Incremental Synthesis of Delay-Tolerant Strategies

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



after playing $\sigma_1, \ldots \sigma_{(k-1)/2}$, play $\{a, c, e\}$

after playing $\sigma_1, \ldots \sigma_{(k-1)/2}$, play $\{b, c, e, f\}$

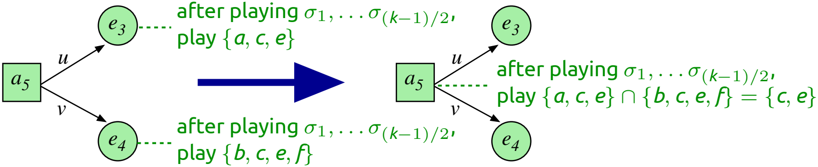
after playing $\sigma_1, \ldots \sigma_{(k-1)/2}$, play $\{a, c, e\} \cap \{b, c, e, f\} = \{c, e\}$

… 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



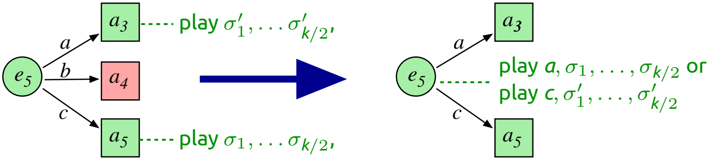
after playing $\sigma_1, \ldots \sigma_{(k-1)/2}$, play $\{a, c, e\}$

after playing $\sigma_1, \ldots \sigma_{(k-1)/2}$, play $\{b, c, e, f\}$

after playing $\sigma_1, \ldots \sigma_{(k-1)/2}$, play $\{a, c, e\} \cap \{b, c, e, f\} = \{c, e\}$

… and eliminate any dead ends by bwd. traversal.

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



play $\sigma'_1, \ldots \sigma'_{k/2}$,

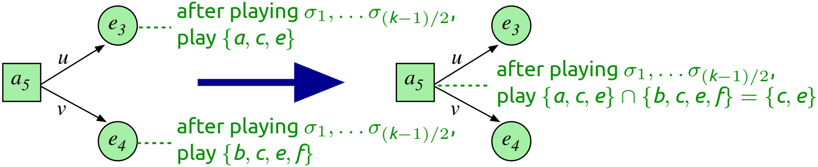
play $\sigma_1, \ldots \sigma_{k/2}$,

play $a, \sigma_1, \ldots, \sigma_{k/2}$ or play $c, \sigma'_1, \ldots, \sigma'_{k/2}$

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



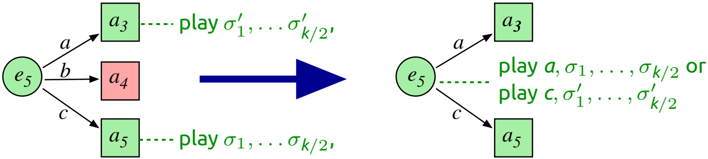
after playing $\sigma_1, \ldots \sigma_{(k-1)/2}$, play $\{a, c, e\}$

after playing $\sigma_1, \ldots \sigma_{(k-1)/2}$, play $\{b, c, e, f\}$

after playing $\sigma_1, \ldots \sigma_{(k-1)/2}$, play $\{a, c, e\} \cap \{b, c, e, f\} = \{c, e\}$

… and eliminate any dead ends by bwd. traversal.

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



play $\sigma'_1, \ldots \sigma'_{k/2}$,

play $\sigma_1, \ldots \sigma_{k/2}$,

play $a, \sigma_1, \ldots, \sigma_{k/2}$ or play $c, \sigma'_1, \ldots, \sigma'_{k/2}$

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

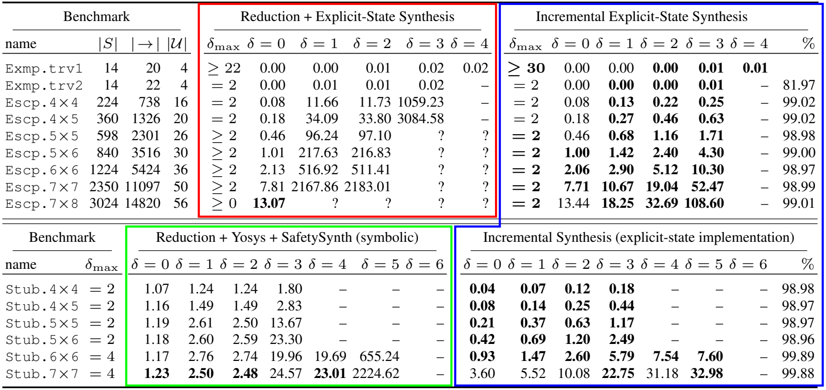
Incremental vs. Reduction-Based

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

Table – Benchmark results in relation to reduction-based approaches (time in seconds)

Incremental vs. Reduction-Based



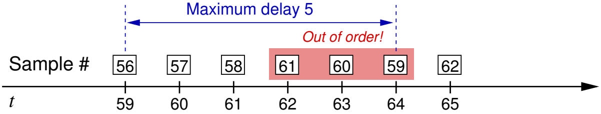
Benchmark				Reduction + Explicit-State Synthesis						Incremental Explicit-State Synthesis						
name	$\lvert S\rvert$	$\lvert\to\rvert$	$\lvert\mathcal{U}\rvert$	$\delta_{\max}$	$\delta = 0$	$\delta = 1$	$\delta = 2$	$\delta = 3$	$\delta = 4$	$\delta_{\max}$	$\delta = 0$	$\delta = 1$	$\delta = 2$	$\delta = 3$	$\delta = 4$	%
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

Table – Benchmark results in relation to reduction-based approaches (time in seconds)

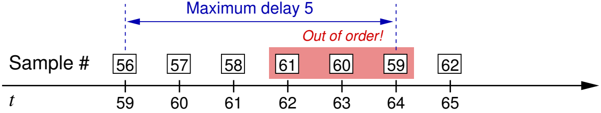
How about Non-Order-Preserving Delays?

- ☹ Observations may arrive out-of-order :

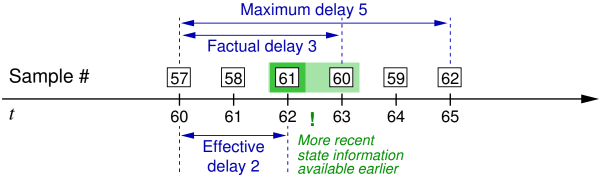


How about Non-Order-Preserving Delays?

☹ Observations may arrive out-of-order :

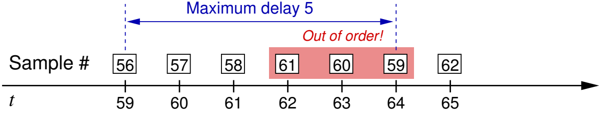


☺ But this may only reduce effective delay, improving controllability :

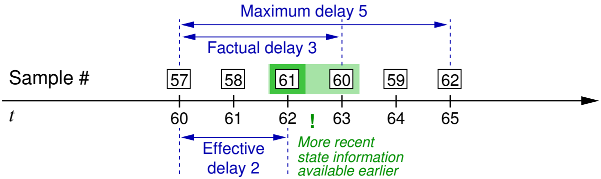


How about Non-Order-Preserving Delays?

- ☹ Observations may arrive out-of-order :



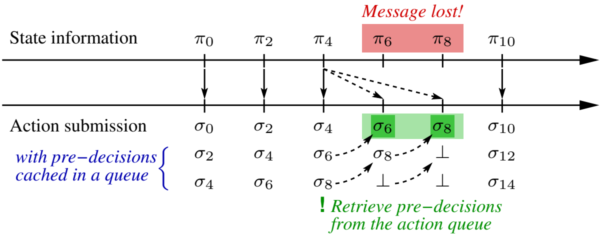
- ☺ But this may only reduce effective delay, improving controllability :



- ☺ 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 :


Message lost!
State information
π_0 π_2 π_4 π_6 π_8 π_{10}
Action submission
σ_0 σ_2 σ_4 σ_6 σ_8 σ_{10}
with pre−decisions cached in a queue
σ_2 σ_4 σ_6 σ_8 $\perp$ σ_{12}
σ_4 σ_6 σ_8 $\perp$ $\perp$ σ_{14}
! Retrieve pre−decisions from the action queue

☺ 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.* Acta Informatica '20.

Outline

1 Synthesizing Safe Controllers Resilient to Delayed Interaction

- Safety Games under Delays
- Incremental Synthesis
- Equivalent Controllability

2 Verifying Safety of Delayed Differential Dynamics

- Delayed Differential Dynamics
- Bounded Safety Verification
- Unbounded Safety Verification

3 Concluding Remarks

- Summary

Solving Delay Differential Equations (DDEs)

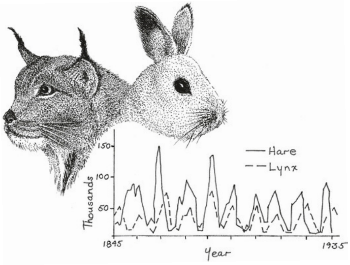
A formal model of delayed feedback control

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

Delayed Coupling in Differential Dynamics

©Wikipedia

Vito Volterra



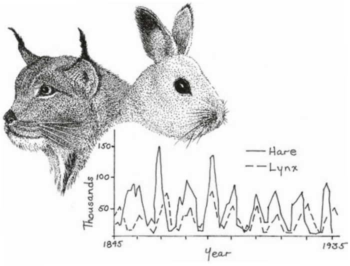
©J. Pastor, 2016

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)

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

Delay Differential Equations (DDEs)

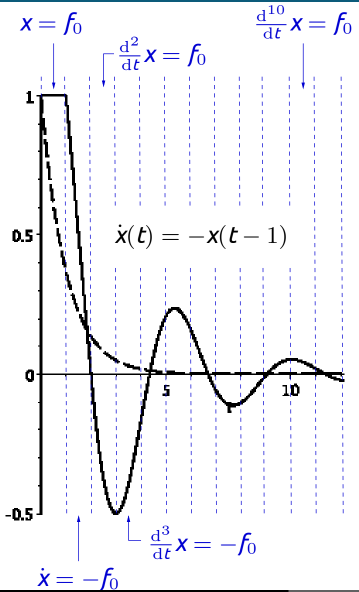
$$\begin{cases} \dot{\mathrm{x}}(t) &= \boldsymbol{f}(\mathrm{x}(t), \mathrm{x}(t-r_1), \ldots, \mathrm{x}(t-r_k)), \quad t \in [0, \infty) \\ \mathrm{x}(t) &= \phi(t), \quad t \in [-r_{\max}, 0] \end{cases}$$

Delay Differential Equations (DDEs)

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

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

Why DDEs are Hard(er)

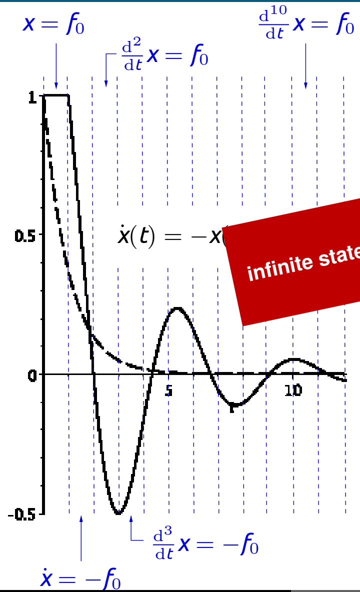


DDEs constitute a model of system dynamics beyond "state snapshots" :

- 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 co… model of system … “state snapshots” :

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

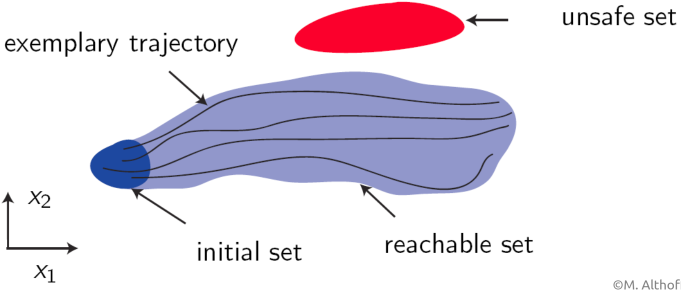
- Thus providing rather infallible, infinite-dimensional memory of the past.

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

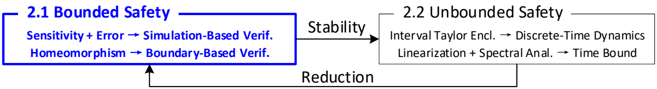
$$\forall \phi \in \{\phi \mid \phi(t) \in \mathcal{X}_0, \forall t \in [-r_{\max}, 0]\} : \quad \left(\bigcup_{t \leq T} \xi_{\mathbf{x}_0}(t)\right) \cap \mathcal{U} = \emptyset \quad ?$$



©M. Althoff, 2010

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

Bounded Safety Verification of DDEs



2.1 Bounded Safety

Sensitivity + Error → Simulation-Based Verif.

Homeomorphism → Boundary-Based Verif.

Stability

2.2 Unbounded Safety

Interval Taylor Encl. → Discrete-Time Dynamics

Linearization + Spectral Anal. → Time Bound

Reduction

Simulation-Based Verification Framework

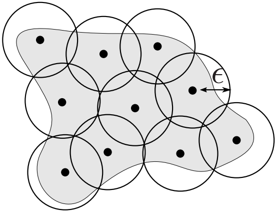


Figure – A finite ϵ-cover of the initial set of states.

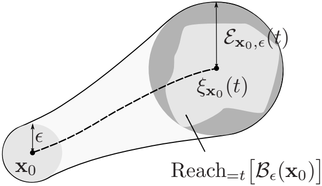
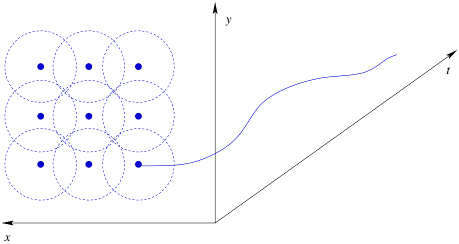

$\mathcal{E}_{\mathbf{x}_0,\epsilon}(t)$
$\xi_{\mathbf{x}_0}(t)$
ϵ
$\mathbf{x}_0$
$\mathrm{Reach}_{=t}\left[\mathcal{B}_\epsilon(\mathbf{x}_0)\right]$

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

©A. Donzé & O. Maler, 2007

Validated Simulation-Based Verification

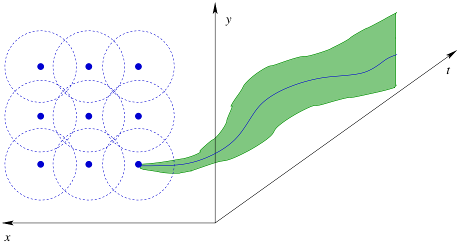
1. Do numerical simulation on a (sufficiently dense) sample of initial states.
2. Add (pessimistic) local-error by solving an optimization problem.
3. “Bloat” the resulting trajectories by sensitivity analysis.



⇒ 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.

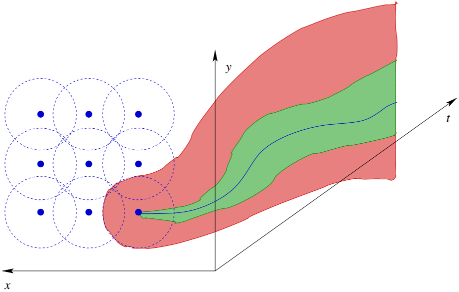


$$E(t) = \begin{cases} d_0\,, & \text{if } t = 0\,, \\ E(t_i) + (t - t_i)e_{i+1}\,, & \text{if } t \in [t_i, t_{i+1}]\,. \end{cases}$$

⇒ 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]

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

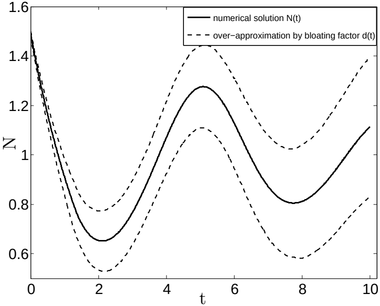


Figure – $\mathcal{X}_0 = \mathcal{B}_{0.01}(1.49)$, $r = 1.3$, $\tau_0 = 0.01$, $T = 10$s.

Example : Delayed Logistic Equation

[G. Hutchinson, 1948]

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

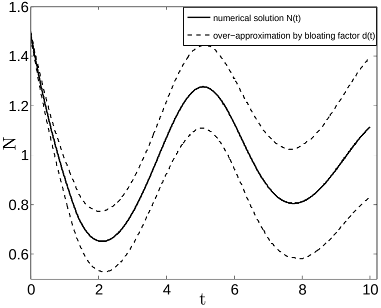


Figure – $\mathcal{X}_0 = \mathcal{B}_{0.01}(1.49)$, $r = 1.3$, $\tau_0 = 0.01$, $T = 10$s.

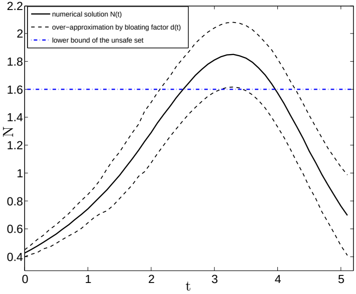
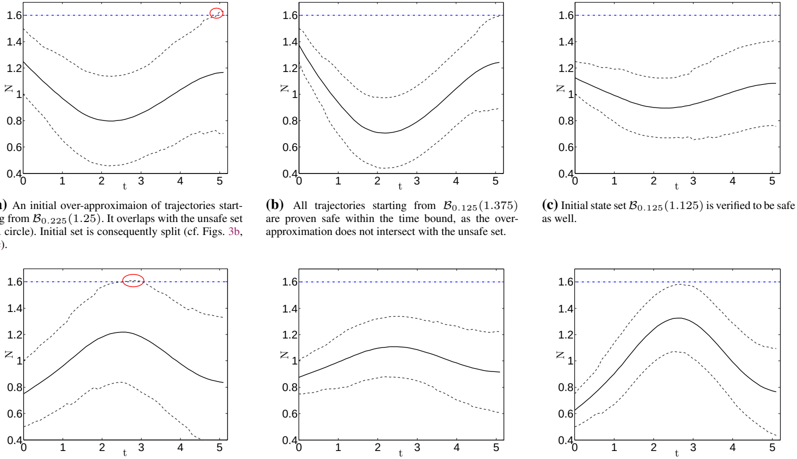


Figure – Over-approximation rigorously proving unsafe, with $r = 1.7$, $\mathcal{X}_0 = \mathcal{B}_{0.025}(0.425)$, $\tau_0 = 0.1$, $T = 5$s, $\mathcal{U} = \{N | N > 1.6\}$.

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]

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

Example : Delayed Microbial Growth

[S. F. Ellermeyer, 1994]

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

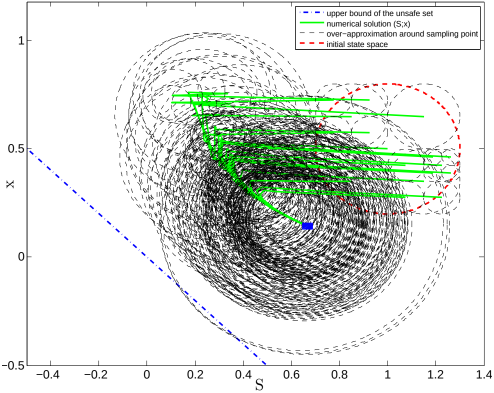
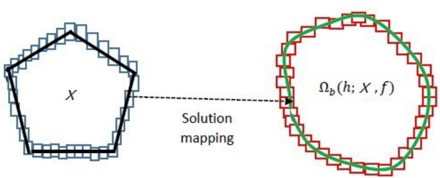


Figure – The microbial system is proven safe by 17 rounds of simulation with $\tau_0 = 0.45$. Here, $f(S) = 2eS/(1+S)$, $r = 0.9$, $\mathcal{X}_0 = \mathcal{B}_{0.3}((1; 0.5))$, $\mathcal{U} = \{(S; x) | S + x < 0\}$, $T = 8$s.

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.

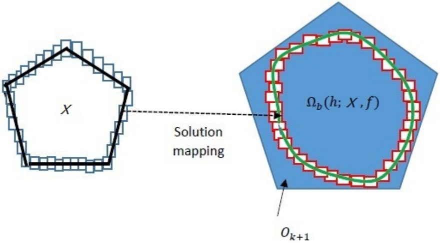


$$r \le \min\left\{ \frac{\epsilon - 1}{\epsilon n^2 M' R}, \frac{\ln R}{2\sqrt{n} n M'}, \frac{\epsilon - 1}{\epsilon(n^2 MR + n^2 NR\epsilon)}, \frac{\ln \frac{R^2+1}{2}}{\sqrt{n}(2nM + n^2 NR\epsilon)} \right\}$$

⇒ 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

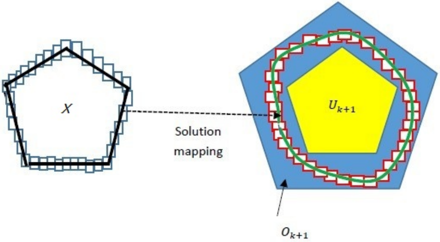
1. Impose a homeomorphism by bounding the time-lag through sensitivity analysis.
2. Compute an enclosure of the reachable set's boundary.
3. Over- (under-)approximate the reachable set by incl. (excl.) the enclosure.



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

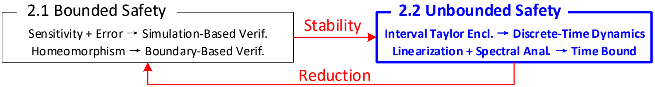
Boundary Propagation-Based Approximation of Reachable Sets

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



2.1 Bounded Safety

Sensitivity + Error → Simulation-Based Verif.

Homeomorphism → Boundary-Based Verif.

Stability

2.2 Unbounded Safety

Interval Taylor Encl. → Discrete-Time Dynamics

Linearization + Spectral Anal. → Time Bound

Reduction

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 so i.g.).

☹ 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 so i.g.).
   - ☹ 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 so i.g.).
   - ☹ 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_{n_i} 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 $f^{(1)}_{n+1}(t)$ and $f^{(2)}_{n+1}(t)$ based on f_n :

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

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 $\mathbf{a}' = \mathcal{M}\mathbf{a}$!

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 $V(\mathbf{A}(n), \eta_n) \leq c \rightarrow \mathcal{S}(f_n(t))$ 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 $\mathcal{S}$.

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 $\boldsymbol{f}^{(1)}_{n+1}, \boldsymbol{f}^{(2)}_{n+1}, \ldots, \boldsymbol{f}^{(k)}_{n+1}$ can now be computed *symbolically* as follows :

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

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

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

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

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

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

☺ Apply polynomial constraint solving to

- pursue BMC exactly as before, unwinding relation (†),
- find a relaxed Lyapunov function by instantiating a polynomial Lyapunov function template w.r.t. (†), 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 :

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

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 :

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

The characteristic equation :

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

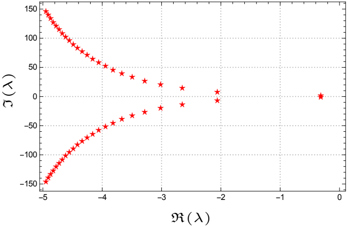
Stability of General Linear Dynamics by Spectral Analysis

For linear DDEs :

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

The characteristic equation :

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



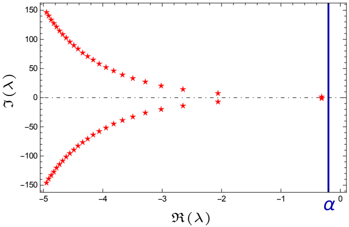
Stability of General Linear Dynamics by Spectral Analysis

For linear DDEs :

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

The characteristic equation :

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



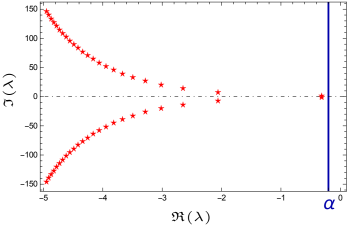
Stability of General Linear Dynamics by Spectral Analysis

For linear DDEs :

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

The characteristic equation :

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



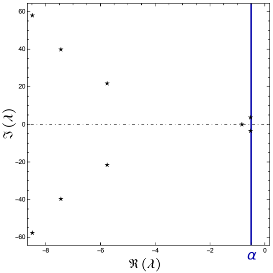
Globally exponentially stable if $\forall \lambda \colon \Re(\lambda) < 0$, i.e.,

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

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 $\iff$ T-safe.



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

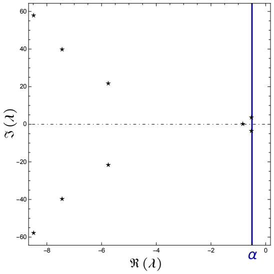
$$\hat{K} = \frac{1}{2\pi}\left(\int_{-M}^{M}\left\|\mathcal{O}\left(\frac{1}{(\alpha+\mathrm{i}\nu)^2}\right)\right\|\,\mathrm{d}\nu + \frac{8n}{M}\left(\|A\| + \|B\|\,\mathrm{e}^{-r\alpha}\right)\right) + 1_0(\alpha)$$

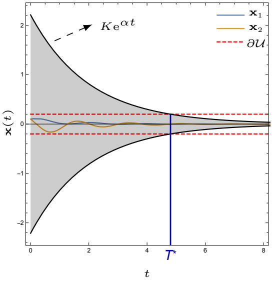
⇒ S. Feng, M. Chen, N. Zhan, M. Fränzle, B. Xue : *Taming delays in dyn. syst. : Unbounded verif. of DDEs*. CAV ’19.

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.



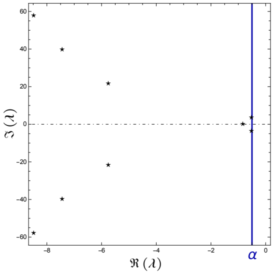


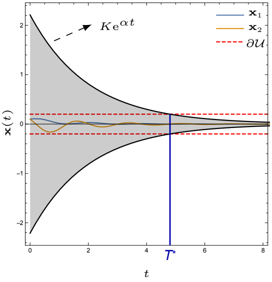
⇒ S. Feng, M. Chen, N. Zhan, M. Fränzle, B. Xue : *Taming delays in dyn. syst. : Unbounded verif. of DDEs.* CAV '19.

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 $\Longleftrightarrow$ 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 :

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

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

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

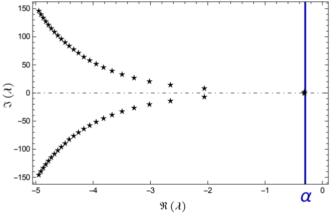
Locally exponentially stable if $\forall\lambda\colon \Re(\lambda) < 0$, i.e.,

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

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.



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

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

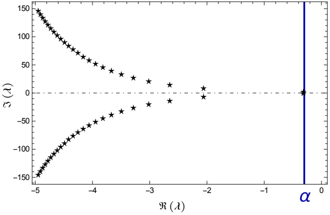
$$\epsilon \le -\alpha/(2\hat{K}\mathrm{e}^{-r\alpha})$$

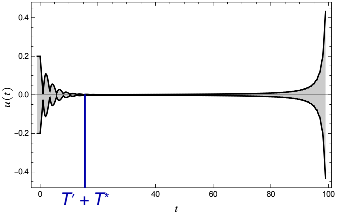
⇒ S. Feng, M. Chen, N. Zhan, M. Fränzle, B. Xue : *Taming delays in dyn. syst. : Unbounded verif. of DDEs*. CAV'19.

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 $\Longleftrightarrow$ T-safe.



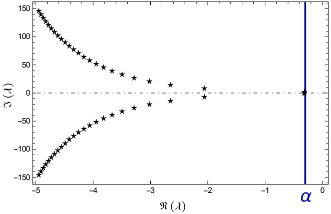


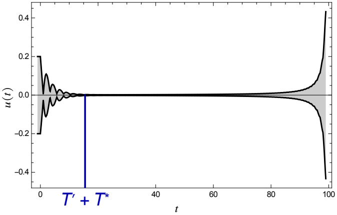
⇒ S. Feng, M. Chen, N. Zhan, M. Fränzle, B. Xue : *Taming delays in dyn. syst. : Unbounded verif. of DDEs*. CAV’19.

Reduction to Bounded Verification

[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]

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

#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 :

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

$$\Downarrow$$

∞-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.

Outline

1 Synthesizing Safe Controllers Resilient to Delayed Interaction
- Safety Games under Delays
- Incremental Synthesis
- Equivalent Controllability

2 Verifying Safety of Delayed Differential Dynamics
- Delayed Differential Dynamics
- Bounded Safety Verification
- Unbounded Safety Verification

3 Concluding Remarks
- Summary

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

- safety games under delays and incremental algorithm for efficient control synthesis,
- bounded safety verification methods for delayed differential dynamics,
- extension to unbounded verification by leveraging stability criteria.

Future Work : We'd explore

- controller synthesis for delayed hybrid systems in the setting of continuous time,
- DDE exhibiting state-dependent or/and stochastic delay,
- hybrid automata comprising DDEs instead of ODEs,
- hybrid automata combining delayed continuous & discrete reactive behaviors,
- invariant generation for time-delayed systems.

HURRAY
FOR DELAY!

Brussels Dichterscollectief
Le Collectif de Poètes Bruxellois
Brussels Poetry Collective

2011	**27**.03