Scheduler design for event-triggered control loops

Manuel Mazo Jr.

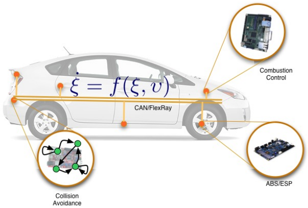
Delft University of Technology

October 26, 2016

Networked Control Systems

Minimize communication load

- To reduce energy consumption in wireless settings
- To efficiently use a communication channel for multiple control loops



Event-triggered control "promises" to solve these problems...

Efficient communications usage

Alternative controller implementations:

- **Periodic**: Controller updated in a periodic fashion
- **Event-triggered**: Controller updated when pre-designed conditions are violated (detected at the sensors)
- **Self-triggered**: Controller updated time dependent on the last measurement obtained (emulate event-triggered)

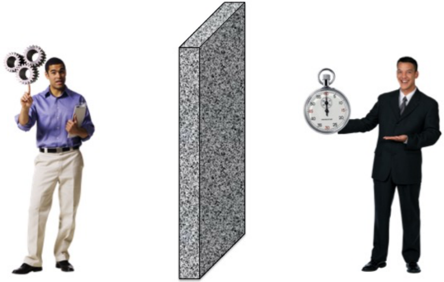
Pros/Cons:

- **Periodic**: Wasteful, easy implementation/scheduling, robust
- **Event-triggered**: Efficient communications use, robust, hard to schedule
- **Self-triggered**: Efficient communications use, easy scheduling, fragile to disturbances

Effective scheduling is critical for network sharing and communication's energy efficiency (in wireless systems).

Our objective: ease the scheduling of **event-triggered** controllers.

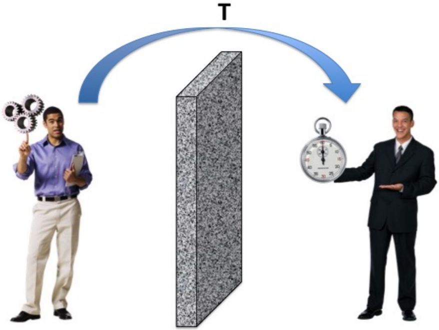
A separation of concerns



- Control engineers design controllers
- Real-time engineers design/implement the scheduling

Which information do they exchange?

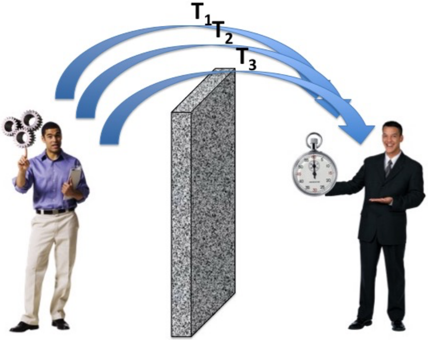
A separation of concerns



- Control engineers design controllers
- Real-time engineers design/implement the scheduling

Periodic: T

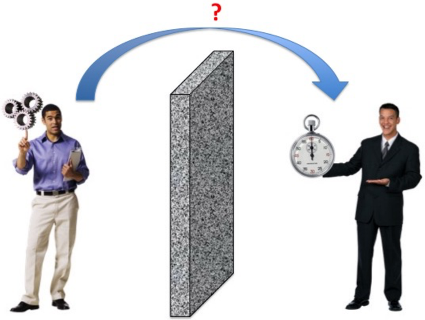
A separation of concerns



- Control engineers design controllers
- Real-time engineers design/implement the scheduling

Self-triggered: T_k

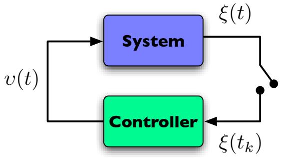
A separation of concerns



- Control engineers design controllers
- Real-time engineers design/implement the scheduling

Event-triggered: ?

Event-triggered LTI control systems



Sample-and-hold implementation:

$$\dot{\xi}(t) = A\xi(t) + B\upsilon(t),\ \xi(t) \in \mathbb{R}^n, \upsilon(t) \in \mathbb{R}^m \tag{1}$$

$$\upsilon(t) = \upsilon(t_k) = K\xi(t_k),\ \forall t \in [t_k, t_{k+1}),\ k \in \mathbb{N}. \tag{2}$$

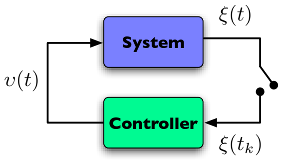
with update times determined by a triggering condition:

$$t_{k+1} = \inf\{t > t_k \mid |e(t)|^2 \geq \alpha|\xi(t)|^2\},\ \alpha \in \mathbb{R}^+,$$

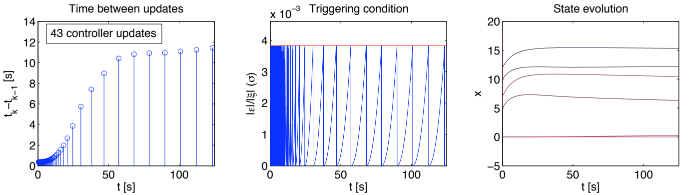
$$e(t) := \xi(t_k) - \xi(t),\ t \in [t_k, t_{k+1}[$$

(all what follows can be extended to any quadratic form on ξ and e)

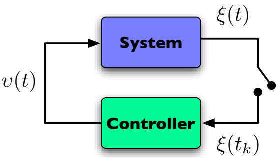
Event-triggered LTI control systems



$$t_{k+1} = \inf\{t > t_k \mid |e(t)|^2 \geq \alpha|\xi(t)|^2\}, \ \alpha \in \mathbb{R}^+,$$



Event-triggered LTI control systems



Sampling interval associated to state x:

$$\tau(x) := \min\{t \mid |e_x(t)|^2 \geq \alpha|\xi_x(t)|^2, \xi_x(0) = x\}. \tag{3}$$

The state-dependent sampling law can be reformulated as:

$$\tau(x) = \min\{\sigma > 0 | \; x^T\Phi(\sigma)x = 0\}, \tag{4}$$

where

$$\Phi(\sigma) = [I - \Lambda^T(\sigma)][I - \Lambda(\sigma)] - \alpha\Lambda^T(\sigma)\Lambda(\sigma),$$

and $\Lambda(\sigma) = [I + \int_0^\sigma e^{Ar}dr(A + BK)]$.

Systems

Definition (System)

A system is a sextuple $(X, X_0, U, \longrightarrow, Y, H)$ consisting of:

- a set of states X;
- a set of initial states $X_0 \subseteq X$;
- a set of inputs U;
- a transition relation $\longrightarrow \subseteq X \times U \times X$;
- a set of outputs Y;
- an output map $H : X \to Y$.

A system is said to be finite if X is a finite countable set, and autonomous if $|U| \leq 1$.

Systems

Definition (Power Quotient System)

Let $S = (X, X_0, \varnothing, \longrightarrow, Y, H)$ be an autonomous system and $R \subseteq X \times X$ be an equivalence relation on X. The power quotient of S by R, denoted by $S_{/R}$, is the autonomous system $(X_{/R}, X_{/R,0}, \varnothing, \underset{/R}{\longrightarrow}, Y_{/R}, H_{/R})$ consisting of:

- $X_{/R} = X/R$;
- $X_{/R,0} = \{x_{/R} \in X_{/R} | x_{/R} \cap X_0 \neq \varnothing\}$;
- $(x_{/R}, u, x'_{/R}) \in \underset{/R}{\longrightarrow}$ if $\exists (x, u, x') \in \longrightarrow$ with $x \in x_{/R}$ and $x' \in x'_{/R}$;
- $Y_{/R} \subset 2^Y$;
- $H_{/R}(x_{/R}) = \bigcup_{x \in x_{/R}} H(x)$.

Systems

Definition (Power Quotient System)

Let $S = (X, X_0, \varnothing, \longrightarrow, Y, H)$ be an autonomous system and $R \subseteq X \times X$ be an equivalence relation on X. The power quotient of S by R, denoted by $S_{/R}$, is the autonomous system $(X_{/R}, X_{/R,0}, \varnothing, \underset{/R}{\longrightarrow}, Y_{/R}, H_{/R})$ consisting of:

- $X_{/R} = X/R$;
- $X_{/R,0} = \{x_{/R} \in X_{/R} | x_{/R} \cap X_0 \neq \varnothing\}$;
- $(x_{/R}, u, x'_{/R}) \in \underset{/R}{\longrightarrow}$ if $\exists (x, u, x') \in \longrightarrow$ with $x \in x_{/R}$ and $x' \in x'_{/R}$;
- $Y_{/R} \subset 2^Y$;
- $H_{/R}(x_{/R}) = \bigcup_{x \in x_{/R}} H(x)$.

Lemma

$S_{/R}$ ε-approximately simulates S, i.e. $S \preceq^{\varepsilon}_{\mathcal{S}} S_{/R}$, for any

$$\varepsilon \geq \max_{\substack{x \in x_{/R} \\ x_{/R} \in X/R}} d(H(x), H_{/R}(x_{/R})),$$

with d the Hausdorff distance over the set 2^Y.

The timing system

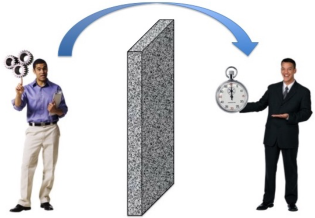
The following system produces as output sequences all possible sequences of inter-sample intervals generated by the event-triggered controller.

$$S = (X, X_0, \varnothing, \longrightarrow, Y, H)$$

- $X = \mathbb{R}^n$;
- $X_0 = \mathbb{R}^n$;
- $(x, x') \in \longrightarrow$ iff $\xi_x(\tau(x)) = x'$ given by (1)-(3);
- $Y \subset \mathbb{R}^+$;
- $H : \mathbb{R}^n \to \mathbb{R}^+$ where $H(x) = \tau(x)$.

Problem: Could we construct a finite system capturing the relevant information of S for scheduler design?

Finite abstractions of the timing system



$S_{/\mathcal{P}} = (X_{/\mathcal{P}}, X_{/\mathcal{P},0}, \varnothing, \xrightarrow[/\mathcal{P}]{}, Y_{/\mathcal{P}}, H_{/\mathcal{P}})$ where

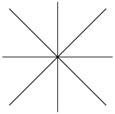
- $X_{/\mathcal{P}} = \mathbb{R}^n_{/\mathcal{P}} := \{\mathcal{R}_1, \ldots, \mathcal{R}_q\}$;
- $X_{/\mathcal{P},0} = \mathbb{R}^n_{/\mathcal{P}}$;
- $(x_{/\mathcal{P}}, x'_{/\mathcal{P}}) \in \xrightarrow[/\mathcal{P}]{}$ if $\exists x \in x_{/\mathcal{P}}$, $\exists x' \in x'_{/\mathcal{P}}$ such that $\xi_x(H(x)) = x'$;
- $Y_{/\mathcal{P}} \subset \mathbb{IR}^+ \subset 2^Y$, where $\mathbb{IR}^+$ is the set of closed intervals $[a, b]$ such that $0 < a \leq b$;
- $H_{/\mathcal{P}}(x_{/\mathcal{P}}) = [\min_{x \in x_{/\mathcal{P}}} H(x), \max_{x \in x_{/\mathcal{P}}} H(x)] := [\underline{\tau}_{x_{/\mathcal{P}}}, \bar{\tau}_{x_{/\mathcal{P}}}]$.

Note: We actually construct "over-approximations" of this symbolic system

Finite abstractions: State set

Remark

All states, excluding the origin, which lie on a line that goes through the origin, have the same inter-sample time, i.e. $\tau(x) = \tau(\lambda x)$, $\forall \lambda \neq 0$.

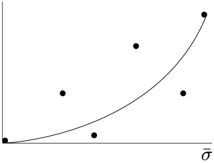


This motivates defining $\mathcal{R}_s$ as **convex polyhedral cones pointed at the origin** (union of rays):

$$\begin{cases} \{x \in \mathbb{R}^2 \,|\, x^T Q_s x \leq 0\} & \text{if } n = 2 \\ \{x \in \mathbb{R}^n \,|\, E_s^T x \preceq 0\} & \text{if } n \geq 3 \end{cases}, \tag{5}$$

Finite abstractions: Output set and map (I)

We compute a time interval $[\underline{\tau}_s, \bar{\tau}_s]$ such that $\forall x \in \mathcal{R}_s, \tau(x) \in [\underline{\tau}_s, \bar{\tau}_s]$.



Lemma

Let $s \in \{1, \ldots, q\}$, and consider a time bound $\underline{\tau}_s \in (0, \bar{\sigma}]$. One can construct a finite set of matrices $\underline{\Phi}_{(i,j),s}$ such that if $x^T \underline{\Phi}_{(i,j),s} x \leq 0$ then:

$$x^T \Phi(\sigma) x \leq 0, \quad \forall \sigma \in [0, \underline{\tau}_s]$$

Lemma

Let $s \in \{1, \ldots, q\}$, and consider a time bound $\bar{\tau}_s \in [\underline{\tau}_s, \bar{\sigma}]$. One can construct a finite set of matrices $\bar{\Phi}_{(i,j),s}$ such that if $x^T \bar{\Phi}_{(i,j),s} x \geq 0$ then:

$$x^T \Phi(\sigma) x \geq 0, \quad \forall \sigma \in [\bar{\tau}_s, \bar{\sigma}]$$

[HET07] L. Hetel, J. Daafouz, and C. Lung, ACC07

Finite abstractions: Output set and map (II)

Applying the S-procedure:

Theorem (Regional Lower Bound Approximation [FIT12])

Consider a scalar $\underline{\tau}_s \in (0, \bar{\sigma}]$ and matrices $\underline{\Phi}_{\kappa,s}$, $\kappa = (i,j) \in \mathcal{K}_s$, as in Lemma 4. If there exist scalars $\underline{\varepsilon}_{\kappa,s} \geq 0$ (for $n = 2$) or symmetric matrices $\underline{U}_{\kappa,s}$ with nonnegative entries (for $n \geq 3$) such that for all $\kappa \in \mathcal{K}_s$ the following LMIs hold:

$$\begin{cases} \underline{\Phi}_{\kappa,s} + \underline{\varepsilon}_{\kappa,s} Q_s \preceq 0 & \text{if } n = 2 \\ \underline{\Phi}_{\kappa,s} + E_s^T \underline{U}_{\kappa,s} E_s \preceq 0 & \text{if } n \geq 3 \end{cases},$$

the inter-sample time (3) of the system (1)-(2) is regionally bounded from below by $\underline{\tau}_s$, $\forall x \in \mathcal{R}_s$.

Theorem (Regional Upper Bound Approximation)

Consider a scalar $\bar{\tau}_s \in [\underline{\tau}_s, \bar{\sigma}]$ and matrices $\bar{\Phi}_{\kappa,s}$, $\kappa = (i,j) \in \mathcal{K}_s$, defined as in Lemma 5. If there exist scalars $\bar{\varepsilon}_{\kappa,s} \geq 0$ (for $n = 2$) or symmetric matrices $\bar{U}_{\kappa,s}$ with nonnegative entries (for $n \geq 3$) such that for all $\kappa \in \mathcal{K}_s$ the following LMIs hold:

$$\begin{cases} \bar{\Phi}_{\kappa,s} - \bar{\varepsilon}_{\kappa,s} Q_s \succeq 0 & \text{if } n = 2 \\ \bar{\Phi}_{\kappa,s} - E_s^T \bar{U}_{\kappa,s} E_s \succeq 0 & \text{if } n \geq 3 \end{cases},$$

the inter-sample time (3) of the system (1)-(2) is regionally bounded from above by $\bar{\tau}_s$, $\forall x \in \mathcal{R}_s$.

[FIT12] C.Fiter, L.Hetel, W.Perruquetti, and J.-P.Richard, Automatica 2012

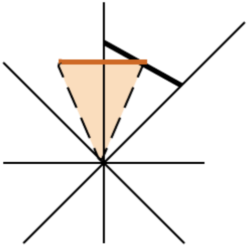
Finite abstractions: Transition relation

Compute an over-approximation $(\bar{\mathcal{X}}_{[\underline{\tau}_s,\bar{\tau}_s]}(x_{/R}))$ of:

$$\mathcal{X}_{[\underline{\tau}_s,\bar{\tau}_s]}(X_s) := \{x' \in \mathbb{R}^n \mid \exists x \in X_s, \exists \tau \in [\underline{\tau}_s,\bar{\tau}_s], x' = \xi_x(\tau)\}$$

$$\mathcal{X}_{[\underline{\tau}_s,\bar{\tau}_s]}(x_{/R}) \subseteq \bar{\mathcal{X}}_{[\underline{\tau}_s,\bar{\tau}_s]}(x_{/R}).$$

Then $(x_{/R}, x'_{/R}) \in \underset{/R}{\longrightarrow}$ if $\bar{\mathcal{X}}_{[\underline{\tau}_s,\bar{\tau}_s]}(x_{/R}) \cap x'_{/R} \neq \emptyset$.

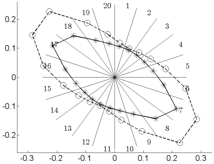


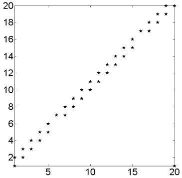
[CHU98] A. Chutinan and B. Krogh, CDC98

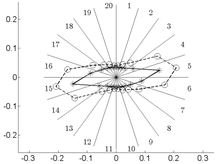
Examples

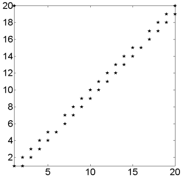
$$\dot{\xi} = \begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix} \xi + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v, \; v = \begin{bmatrix} 1 & -4 \end{bmatrix} \xi. \tag{6}$$

$$\dot{\xi} = \begin{bmatrix} -0.5 & 0 \\ 0 & 3.5 \end{bmatrix} \xi + \begin{bmatrix} 1 \\ 1 \end{bmatrix} v, \; v = \begin{bmatrix} 1.02 & -5.62 \end{bmatrix} \tag{7}$$









[TAB07] P. Tabuada, TAC07

[HET11] L. Hetel, et al, TAC11

Scheduling with Timed-Automata

Observe that the system $S_{/\mathcal{P}}$:

1. remains at $x_{/\mathcal{P}}$ during the time interval $[0, \underline{\tau}_{x_{/\mathcal{P}}}[$,
2. possibly leaves $x_{/\mathcal{P}}$ during the time interval $[\underline{\tau}_{x_{/\mathcal{P}}}, \bar{\tau}_{x_{/\mathcal{P}}}[$ (Guards),
3. is forced to leave $x_{/\mathcal{P}}$ at $\bar{\tau}_{x_{/\mathcal{P}}}$ (Invariants).

making $S_{/\mathcal{P}}$ **semantically equivalent to a Timed Automata (TA)**.

One can then **leverage TA tools to synthesize schedulers** for multiple Event-Triggered Control (ETC) loops by:

1. Construct the TA related to each ETC
2. Enrich these automata with controllable actions (e.g. forcing early triggering)
3. Construct a simple model for network access
4. Compose the enriched TA (a TGA) with the network model and solve safety synthesis problems

Timed-Automata

Definition (Timed Automaton)

A timed automaton TA is a sextuple $(L, \ell_0, \mathrm{Act}, C, E, \mathrm{Inv})$ where

- L is the set of finitely many locations (or nodes);
- $\ell_0 \in L$ is the initial location;
- Act is the set of finitely many actions;
- C is the set of finitely many real-valued clocks;
- $E \subseteq L \times \mathcal{B}(C) \times \mathrm{Act} \times 2^C \times L$ is the set of edges;
- $\mathrm{Inv} : L \to \mathcal{B}(C)$ assigns invariants to locations.

where $\mathcal{B}(C)$ to denote the set of clock constraints.

The states of a TA are pairs of locations ℓ and clock assignments u. The transitions can be:

- **Delayed transition**: $(\ell, u) \xrightarrow[TS]{d} (\ell, u + d)$ if $u \vDash \mathrm{Inv}(\ell)$ and $(u + d) \vDash \mathrm{Inv}(\ell)$ for a non-negative real number $d \in \mathbb{R}_{\geq 0}$;
- **Discrete transition**: $(\ell, u) \xrightarrow[TS]{a} (\ell', u')$ if $\ell \xrightarrow{g,a,\mathbf{r}} \ell'$, $u \vDash g$, $u' = u[\mathbf{r}]$ and $u' \vDash \mathrm{Inv}(\ell')$.

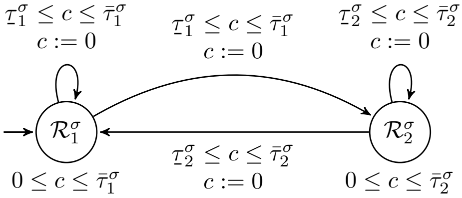
Timed-Automata

Definition (Timed Automaton)

A timed automaton TA is a sextuple $(L, \ell_0, \mathrm{Act}, C, E, \mathrm{Inv})$ where

- L is the set of finitely many locations (or nodes);
- $\ell_0 \in L$ is the initial location;
- Act is the set of finitely many actions;
- C is the set of finitely many real-valued clocks;
- $E \subseteq L \times \mathcal{B}(C) \times \mathrm{Act} \times 2^C \times L$ is the set of edges;
- $\mathrm{Inv} : L \to \mathcal{B}(C)$ assigns invariants to locations.

where $\mathcal{B}(C)$ to denote the set of clock constraints.



Timed Game Automata

Definition (Timed Game Automaton)

A timed game automaton TGA is a septuple $(L, \ell_0, \mathrm{Act}_c, \mathrm{Act}_u, C, E, \mathrm{Inv})$ where $(L, \ell_0, \mathrm{Act}_c \cup \mathrm{Act}_u, C, E, \mathrm{Inv})$ is a timed automaton with:

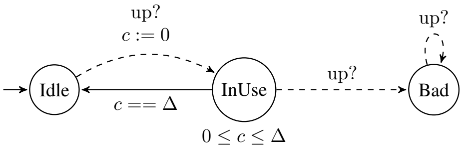
- Act_c is the set of controllable actions;
- Act_u is the set of uncontrollable actions;
- $\mathrm{Act}_c \cap \mathrm{Act}_u = \emptyset$.

We define two different types of TGA:

- TGA^{net} - Capturing the network availability behaviour
- TGA_i^{cl} - An enriched version of the TA abstraction of the event-triggered loop, with controllable actions:
  - Selection of different α triggering coefficient (performance selection)
  - Force an early deterministic update

TUDelft

TGA^{net}



Definition

Let Δ represent the minimum channel occupancy time, $\text{TGA}^{net} = (L^{net}, \ell_0^{net}, \text{Act}_c^{net}, \text{Act}_u^{net}, C^{net}, E^{net}, \text{Inv}^{net})$ is defined by:

- the set of locations $L^{net} = \{Idle, InUse, Bad\}$;
- the initial location $\ell_0^{net} = Idle$;
- the controllable actions $\text{Act}_c^{net} = \{*\}$;
- the uncontrollable actions $\text{Act}_u^{net} = \{up?\}$;
- the set of clock variables $C^{net} = \{c\}$;
- the set of edges $E^{net} = \{(Idle, true, up?, \{c\}, InUse), (InUse, c = \Delta, *, \emptyset, Idle), (InUse, true, up?, \emptyset, Bad), (Bad, true, up?, \emptyset, Bad)\}$;
- $\text{Inv}^{net}(InUse) = \{c \mid 0 \leq c \leq \Delta\}$.

(Networks of) Timed Game Automata

Definition (Parallel Composition (NTGA))

The parallel composition of $\mathrm{TGA}_1, \ldots, \mathrm{TGA}_n$ denoted by $\mathrm{TGA}_1 \mid \cdots \mid \mathrm{TGA}_n$ is a timed game automaton $\mathrm{TGA} = (L, \ell_0, \mathrm{Act}_c, \mathrm{Act}_u, C, E, \mathrm{Inv})$ where

- $L = L^1 \times \cdots \times L^n$;
- $\ell_0 = (\ell_0^1, \ldots, \ell_0^n)$;
- $\mathrm{Act}_c = \{*\} \cup \bigcup_{i=1}^n \{a \in \mathrm{Act}_c^i \mid a \text{ is an internal action}\}$;
- $\mathrm{Act}_u = \{\circledast\} \cup \bigcup_{i=1}^n \{a \in \mathrm{Act}_u^i \mid a \text{ is an internal action}\}$;
- $C = C^1 \cup \cdots \cup C^n$;
- E is defined according to the following two rules:
    - a TA makes a move on its own via its internal action: the edge is controllable iff the internal action is controllable;
    - two TA move simultaneously via a synchronizing action: the edge is controllable iff both input and output actions are controllable (i.e. the environment has priority over the controller);
- $\mathrm{Inv}((\ell_1, \ldots, \ell_n)) = \mathrm{Inv}^1(\ell_1) \wedge \cdots \wedge \mathrm{Inv}^n(\ell_n)$.

Synthesis problem

1. Construct TGA_i^{cl} for each ETC loop.
2. Construct TGA^{net}
3. Construct $\mathrm{NTGA} = \mathrm{TGA}_1^{cl} \mid \mathrm{TGA}_1^{cl} \mid \cdots \mid \mathrm{TGA}^{net}$
4. **solve safety synthesis problems over** NTGA

Synthesis problem: Design strategies to avoid visiting the NTGA set of states:

$$\mathcal{A} = \{(\ell_{net}, \ell_1, \ldots, \ell_N, u_{net}, u_1, \ldots, u_N) \mid \ell_{net} = Bad\}.$$

Issue: Synthesis tries to force always "early triggering".

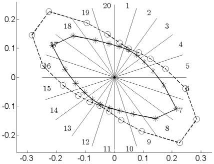
Solutions:

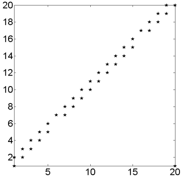
- Define costs of actions and find "optimal" strategies (no synthesis results available).
- Forbid the use of more than a pre-specified number of consecutive "early triggers".

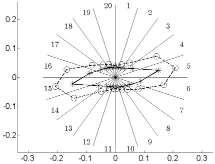
Abstractions

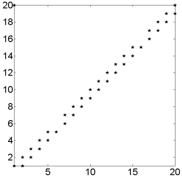
$$\dot{\xi} = \begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix} \xi + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \upsilon, \ \upsilon = \begin{bmatrix} 1 & -4 \end{bmatrix} \xi. \tag{8}$$

$$\dot{\xi} = \begin{bmatrix} -0.5 & 0 \\ 0 & 3.5 \end{bmatrix} \xi + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \upsilon, \ \upsilon = \begin{bmatrix} 1.02 & -5.62 \end{bmatrix} \tag{9}$$









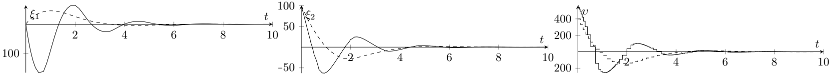
[TAB07] P. Tabuada, TAC07

[HET11] L. Hetel, et al, TAC11

Schedule

- Channel occupancy after event: $\Delta = 0.005$ s;
- Controllable action: force an early triggering $d = 0.005$ s before $\underline{\tau}_s$ (of each region);
- Abstractions with 200 conic regions;
- Maximum consecutive earlier triggering: 4.

Long bars = Event-triggered   Top [TAB07]
Short bars = Early triggering   Bottom [HET11]



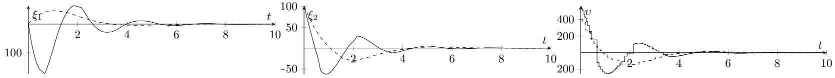
The scheduler is synthesized on **UPAAL-Tiga** to solve the safety game [CAS05].

[CAS05] F. Cassez, A. David, E. Fleury, K. Larsen, and D. Lime, CONCUR'05

Schedule

- Channel occupancy after event: $\Delta = 0.005$ s;
- Controllable action: three different triggering coefficients $\alpha_1 < \alpha_2 < \alpha_3$;
- Abstractions with 200 conic regions;
- Maximum consecutive earlier triggering: 0.

Top [TAB07], Bottom [HET11]
Short/Medium/Long bars = $\alpha_1/\alpha_2/\alpha_3$



The scheduler is synthesized on **UPAAL-Tiga** to solve the safety game [CAS05].

[CAS05] F. Cassez, A. David, E. Fleury, K. Larsen, and D. Lime, CONCUR'05

Wrap-up

We provide:

- A class of abstractions (in the form of timed-automata) for the timing of event-triggered systems
- An approach to scheduler design for ETCs

On-going/Future work:

- Extension to disturbed systems (To be presented at CDC16)
- Extensions to classes of non-linear systems
- More advanced (decentralized?) scheduling approaches
- Performance optimization through TPGA
- Fault/Attack detection
- Toolbox


Thanks for your attention!