Uncertainties, adaptability, and verification

Valentina Castiglioni,
Michele Loreti, and
Simone Tini

OPCT, Bertinoro, 26 June 2023

Uncertainties, adaptabiLiTy, and veRificatiON

aka Project

ULTRON

Valentina Castiglioni,
Michele Loreti, and
Simone Tini

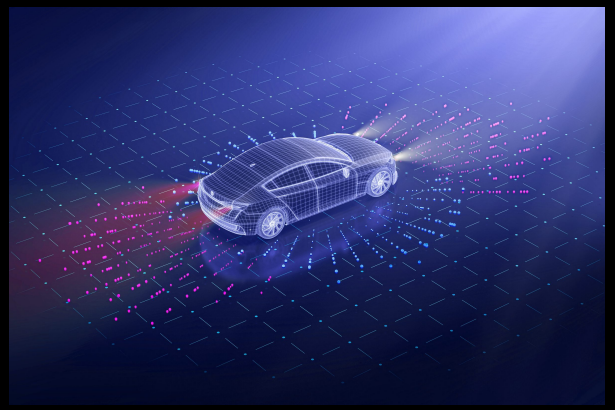
OPCT, Bertinoro, 26 June 2023

The name of the game

System: Agent + environment

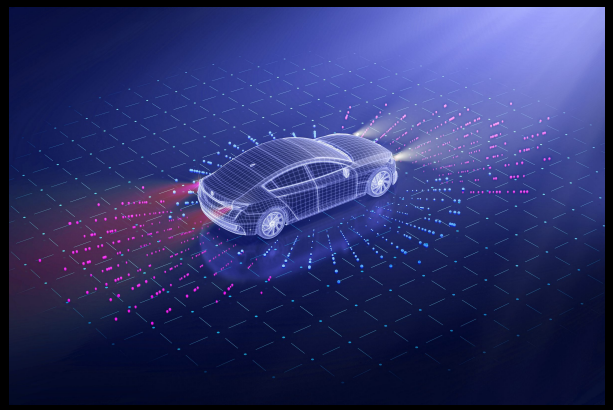
The name of the game

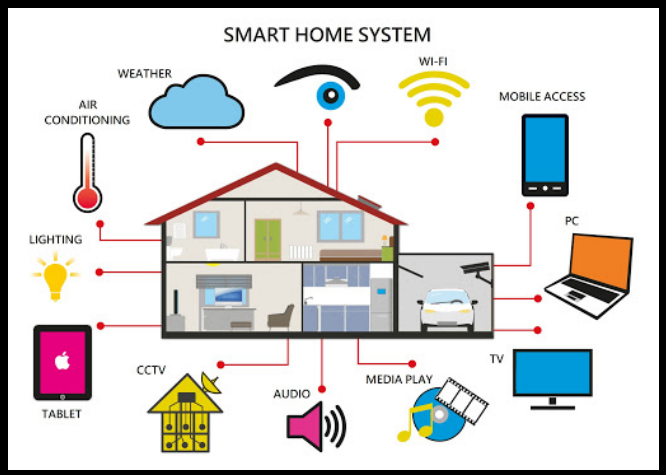
System: Agent + environment

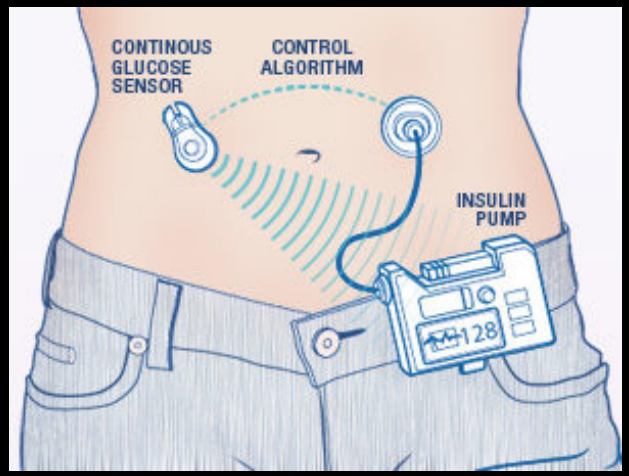


The name of the game

System: Agent + environment

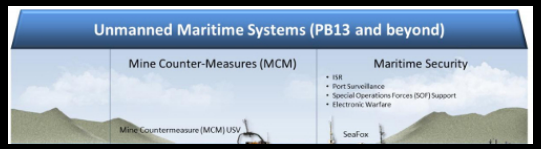


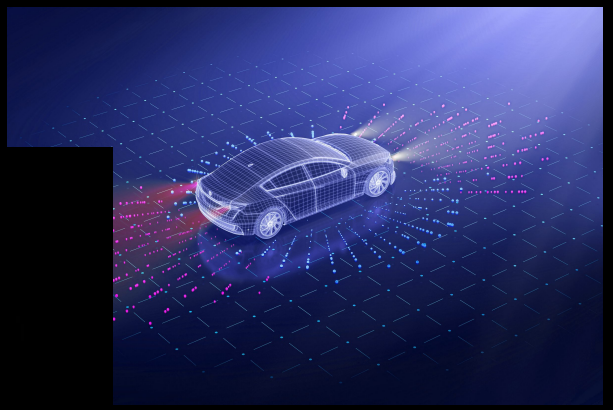


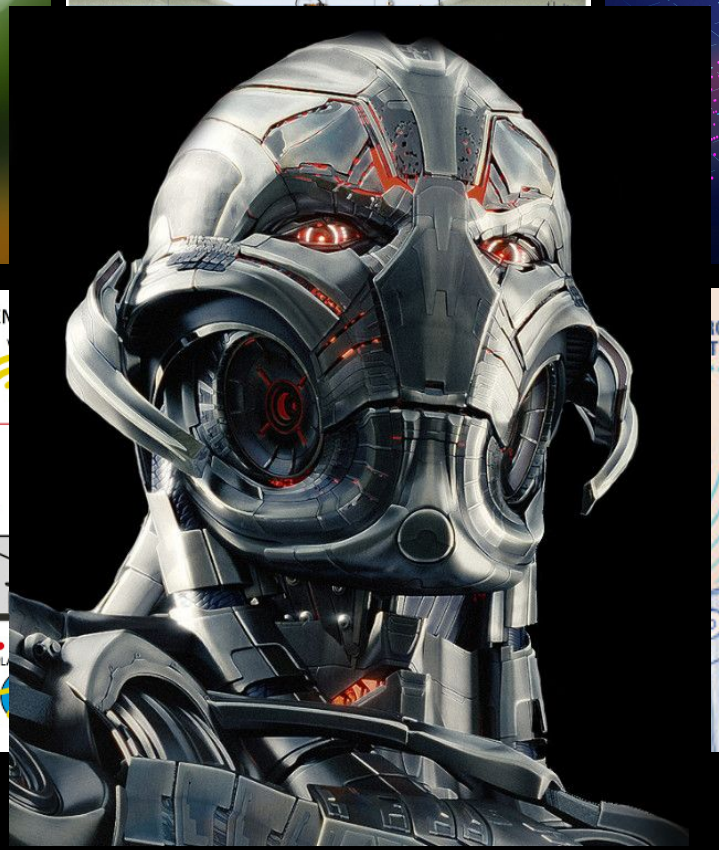


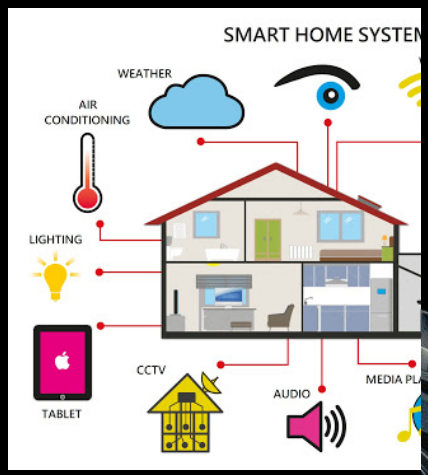
The name of the game

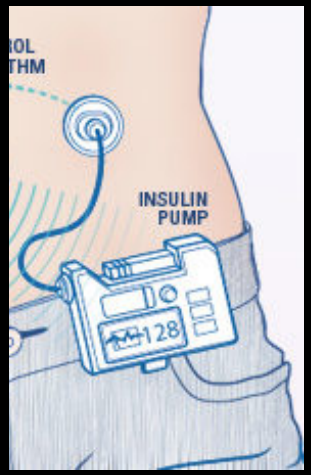
System: Agent + environment





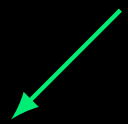






The name of the game

System: Agent + environment



- must **adapt** to changes in the environment
- partially modified by the agent

The name of the game

System: Agent + environment

- must **adapt** to changes in the environment
- subject to
  - randomisation
  - measurement errors
  - attacks

- partially modified by the agent
- **unpredictable** behaviour
  - physical phenomena
  - heterogeneous devices
  - failures, conflicts, …

The name of the game

System: Agent + environment

- must **adapt** to changes in the environment
- subject to
  - randomisation
  - measurement errors
  - attacks

- partially modified by the agent
- **unpredictable** behaviour
  - physical phenomena
  - heterogeneous devices
  - failures, conflicts, …

Problem: uncertainty

The main challenge: robustness

The main challenge: robustness

An agent is **robust** against uncertainties if it can fulfill its tasks in spite of their presence

The main challenge: robustness

An agent is **robust** against uncertainties if it can fulfill its tasks in spite of their presence

Our aim is to provide the tools to model and verify the robustness of cyber-physical systems against uncertainties

The model

- We model agent and environment **separately**.
- We model their **interaction** in terms of the **changes** they induce on a set of application-relevant **data**.
- Setting: **discrete time**, **continuous space**

The model

- We model agent and environment **separately**.
- We model their **interaction** in terms of the **changes** they induce on a set of application-relevant **data**.
- Setting: **discrete time**, **continuous space**

- System behaviour: **how data evolve in time**.

The model (intuition)

Data space: $\mathcal{D}$ closed subset of $\mathbb{R}^n$

The model (intuition)

Data space: $\mathcal{D}$ closed subset of $\mathbb{R}^n$

Data state: $\mathbf{d} \in \mathcal{D}$ identifying the *current configuration of data*

The model (intuition)

Data space: $\mathcal{D}$ closed subset of $\mathbb{R}^n$

Data state: $\mathbf{d} \in \mathcal{D}$ identifying the *current configuration of data*

Agent: specified by processes in the *generative probabilistic model*

$$
\begin{array}{rcl}
P & ::= & (\overline{e} \to \overline{x}).P' \mid \text{if } [e]\ P_1 \text{ else } P_2 \mid \sum_{i \in I} r_i \cdot P_i \mid P_1 \parallel P_2 \mid A \\
e & ::= & \text{value} \mid x \mid op_k(e_1, \ldots, e_k)
\end{array}
$$

The model (intuition)

Data space: $\mathcal{D}$ closed subset of $\mathbb{R}^n$

Data state: $\mathbf{d} \in \mathcal{D}$ identifying the *current configuration of data*

Agent: specified by processes in the *generative probabilistic model*

$$
\begin{array}{rcl}
P & ::= & (\overline{e} \to \overline{x}).P' \mid \text{if } [e]\ P_1 \text{ else } P_2 \mid \sum_{i \in I} r_i \cdot P_i \mid P_1 \parallel P_2 \mid A \\
e & ::= & \text{value} \mid x \mid op_k(e_1, \ldots, e_k)
\end{array}
$$

Environment: function $\mathcal{E} : \mathcal{D} \to \Pi(\mathcal{D})$ that maps a *data state* into a *probability measure over data states*

The model (intuition)

Data space: $\mathcal{D}$ closed subset of $\mathbb{R}^n$

Data state: $\mathbf{d} \in \mathcal{D}$ identifying the *current configuration of data*

Agent: specified by processes in the *generative probabilistic model*

$$
\begin{array}{rcl}
P & ::= & (\overline{e} \to \overline{x}).P' \mid \text{if } [e]\ P_1 \text{ else } P_2 \mid \sum_{i \in I} r_i \cdot P_i \mid P_1 \parallel P_2 \mid A \\
e & ::= & \text{value} \mid x \mid op_k(e_1, \ldots, e_k)
\end{array}
$$

Environment: function $\mathcal{E} : \mathcal{D} \to \Pi(\mathcal{D})$ that maps a *data state* into a *probability measure over data states*

Configuration: a particular state of the system $\mathbf{c} = \langle P, \mathbf{d} \rangle_{\mathcal{E}}$

System behaviour

Agent semantics: $Pstep(P, \mathbf{d}) = \sum_{i \in I} q_i \cdot \delta(P_i, \mathbf{d_i})$

System behaviour

Agent semantics: $Pstep(P, \mathbf{d}) = \sum_{i \in I} q_i \cdot \delta(P_i, \mathbf{d_i})$

Dirac's delta

probability weights

effects on d

process at next step

System behaviour

Agent semantics: $\mathit{Pstep}(P, \mathbf{d}) = \sum_{i \in I} q_i \cdot \delta(P_i, \mathbf{d_i})$

One-step configuration semantics:

$$\mathit{Cstep}(\langle P, \mathbf{d} \rangle_{\mathcal{E}})(C) = \sum_{(P', \mathbf{d}')} \mathit{Pstep}(P, \mathbf{d})(P', \mathbf{d}') \cdot \langle \delta(P'), \mathcal{E}(\mathbf{d}') \rangle_{\mathcal{E}}(C)$$

System behaviour

Agent semantics: $\mathit{Pstep}(P, \mathbf{d}) = \sum_{i \in I} q_i \cdot \delta(P_i, \mathbf{d}_i)$

One-step configuration semantics:

discrete distribution induced by P

$$\mathit{Cstep}(\langle P, \mathbf{d} \rangle_{\mathcal{E}})(C) = \sum_{(P', \mathbf{d}')} \mathit{Pstep}(P, \mathbf{d})(P', \mathbf{d}') \cdot \langle \delta(P'), \mathcal{E}(\mathbf{d}') \rangle_{\mathcal{E}}(C)$$

continuous distribution induced by $\mathcal{E}$

System behaviour

Agent semantics: $\mathit{Pstep}(P, \mathbf{d}) = \sum_{i \in I} q_i \cdot \delta(P_i, \mathbf{d_i})$

One-step configuration semantics:

Markov kernel: $\mathit{Cstep}(c)(C)$

System behaviour

Agent semantics: $Pstep(P, \mathbf{d}) = \sum_{i \in I} q_i \cdot \delta(P_i, \mathbf{d_i})$

One-step configuration semantics:

Markov kernel: $Cstep(c)(C)$

Multi-step configuration semantics:

$$Mstep_{c,i+1}(C) = \int_{\mathcal{C}} Cstep(b)(C)\ d(Mstep_{c,i}(b))$$

System behaviour

Agent semantics: $Pstep(P, \mathbf{d}) = \sum_{i \in I} q_i \cdot \delta(P_i, \mathbf{d_i})$

One-step configuration semantics:

Markov kernel: $Cstep(c)(C)$

Multi-step configuration semantics:

Markov process generated by $Cstep(c)(C)$: $Mstep_{c,\tau}(C)$

System behaviour

Agent semantics: $Pstep(P, \mathbf{d}) = \sum_{i \in I} q_i \cdot \delta(P_i, \mathbf{d_i})$

One-step configuration semantics:

Markov kernel: $Cstep(c)(C)$

Multi-step configuration semantics:

Markov process generated by $Cstep(c)(C)$: $Mstep_{c,\tau}(C)$

System behaviour: the Evolution Sequence

$$\mathcal{S}_{c,\tau}(D) = Mstep_{c,\tau}(\langle \mathcal{P}, D \rangle_{\mathcal{E}})$$

sequence of probability measures over data states

Formalisation of robustness

Formalisation of robustness

Robustness: being able to function correctly even in the presence of uncertainties

Formalisation of robustness

Robustness: being able to function correctly even in the presence of uncertainties

Expressed by measuring the capability of an agent to tolerate perturbations in the environmental conditions and still fulfill its tasks

Formalisation of robustness

Robustness: being able to function correctly even in the presence of uncertainties

Expressed by measuring the capability of an agent to tolerate perturbations in the environmental conditions and still fulfill its tasks

- We need to measure the differences between the behaviour of the system and its behaviour under the effect of perturbations, possibly at different moments in time

Robustness is a temporal property of distances between system behaviours

Robustness Temporal Logic

RobTL: a temporal logic for the specification of requirements on the evolution of distances between systems behaviours

It allows us to:

Robustness Temporal Logic

RobTL: a temporal logic for the specification of requirements on the evolution of distances between systems behaviours

It allows us to:

- Specify different distances:
  - capturing different tasks of the system
  - having different formalisations

Robustness Temporal Logic

RobTL: a temporal logic for the specification of requirements on the evolution of distances between systems behaviours

It allows us to:

- Specify different distances:
  - capturing different tasks of the system
  - having different formalisations

- Compare distances and verify temporal requirements on them

Robustness Temporal Logic

RobTL: a temporal logic for the specification of requirements on the evolution of distances between systems behaviours

It allows us to:

- Specify different distances:
  - capturing different tasks of the system
  - having different formalisations

RobTL expressions

- Compare distances and verify temporal requirements on them

Robustness Temporal Logic

RobTL: a temporal logic for the specification of requirements on the evolution of distances between systems behaviours

It allows us to:

- Specify different distances:
  - capturing different tasks of the system
  - having different formalisations

RobTL expressions

- Compare distances and verify temporal requirements on them

RobTL formulae

RobTL expressions

$$\texttt{exp} ::= \quad <^{\rho} \mid >^{\rho} \mid$$

$$\texttt{F}^{I}\,\texttt{exp} \mid \texttt{G}^{I}\,\texttt{exp} \mid \texttt{exp}\,\texttt{U}^{I}\,\texttt{exp} \mid$$

$$\texttt{min}\,(\texttt{exp}, \texttt{exp}) \mid \texttt{max}\,(\texttt{exp}, \texttt{exp}) \mid \sum_{k \in K} w_k \cdot \texttt{exp}_k \mid$$

$$\sigma(\texttt{exp}, \bowtie \zeta)$$

RobTL expressions

$$\texttt{exp} ::= \quad <^{\rho} \mid >^{\rho} \mid$$

Atomic expressions
Evaluate the distance between two distributions at a given time

$$\texttt{F}^{I}\,\texttt{exp} \mid \texttt{G}^{I}\,\texttt{exp} \mid \texttt{exp}\,\texttt{U}^{I}\,\texttt{exp} \mid$$

$$\texttt{min}(\texttt{exp}, \texttt{exp}) \mid \texttt{max}(\texttt{exp}, \texttt{exp}) \mid \sum_{k \in K} w_k \cdot \texttt{exp}_k \mid$$

$$\sigma(\texttt{exp}, \bowtie \zeta)$$

RobTL expressions

$$\texttt{exp} \;::=\; <^{\rho} \;\mid\; >^{\rho} \;\mid$$

Atomic expressions
Evaluate the distance between two distributions at a given time

Penalty function $\rho\colon \mathcal{D} \to [0,1]$ to quantify flaws in behaviour

Distance over data states $\boldsymbol{m_\rho}\colon \mathcal{D} \times \mathcal{D} \to [0,1]$:

$$m_\rho(\mathbf{d}_1, \mathbf{d}_2) = \max\{\rho(\mathbf{d}_2) - \rho(\mathbf{d}_1), 0\}$$

hemimetric expressing how much $\mathbf{d}_2$ is worse than $\mathbf{d}_1$ (wrt ρ)

Wasserstein lifting to distributions $\mathbf{W}(\boldsymbol{m_\rho})\colon \Pi(\mathcal{D}) \times \Pi(\mathcal{D}) \to [0,1]$

$$\mathbf{W}(m_\rho)(\mu,\nu) = \inf_{\mathfrak{w} \in \mathfrak{W}(\mu,\nu)} \int_{\mathcal{D}\times\mathcal{D}} m_\rho(\mathbf{d}_1, \mathbf{d}_2)\, d\mathfrak{w}(\mathbf{d}_1, \mathbf{d}_2)$$

the inf of the expected values of the distance over the couplings

RobTL expressions

$$\texttt{exp} ::= \quad <^{\rho} \mid >^{\rho} \mid$$

temporal expressions
Evaluate distance over time
(idea $\exists = \min$, $\forall = \max$)

$$\texttt{F}'\,\texttt{exp} \quad \mid \quad \texttt{G}'\,\texttt{exp} \quad \mid \quad \texttt{exp}\,\texttt{U}'\,\texttt{exp} \quad \mid$$

$$\min(\texttt{exp}, \texttt{exp}) \quad \mid \quad \max(\texttt{exp}, \texttt{exp}) \quad \mid \quad \sum_{k \in K} w_k \cdot \texttt{exp}_k \quad \mid$$

$$\sigma(\texttt{exp}, \bowtie \zeta)$$

RobTL expressions

$$\texttt{exp} ::= \quad <^{\rho} \mid >^{\rho} \mid$$

$$\texttt{F}^{I}\,\texttt{exp} \mid \texttt{G}^{I}\,\texttt{exp} \mid \texttt{exp}\,\texttt{U}^{I}\,\texttt{exp} \mid$$

$$\texttt{min}\,(\texttt{exp}, \texttt{exp}) \mid \texttt{max}\,(\texttt{exp}, \texttt{exp}) \mid \sum_{k \in K} w_k \cdot \texttt{exp}_k \mid$$

$$\sigma(\texttt{exp}, \bowtie \zeta)$$

mathematical expressions

No surprises

$(\sum_k w_k = 1)$

RobTL expressions

$$\texttt{exp} ::= \quad <^{\rho} \mid >^{\rho} \mid$$

$$\texttt{F}^{I}\,\texttt{exp} \mid \texttt{G}^{I}\,\texttt{exp} \mid \texttt{exp}\,\texttt{U}^{I}\,\texttt{exp} \mid$$

$$\texttt{min}\,(\texttt{exp}, \texttt{exp}) \mid \texttt{max}\,(\texttt{exp}, \texttt{exp}) \mid \sum_{k \in K} w_k \cdot \texttt{exp}_k \mid$$

$\sigma(\texttt{exp}, \bowtie \zeta)$

conditional expression
if exp $\bowtie \zeta$, then 0
else 1

RobTL expressions

$$\texttt{exp} ::= \quad <^{\rho} \mid >^{\rho} \mid$$

$$\texttt{F}^{I}\,\texttt{exp} \mid \texttt{G}^{I}\,\texttt{exp} \mid \texttt{exp}\,\texttt{U}^{I}\,\texttt{exp} \mid$$

$$\texttt{min}\,(\texttt{exp}, \texttt{exp}) \mid \texttt{max}\,(\texttt{exp}, \texttt{exp}) \mid \sum_{k \in K} w_k \cdot \texttt{exp}_k \mid$$

$$\sigma(\texttt{exp}, \bowtie \zeta)$$

RobTL formulae

$$\varphi \ ::= \top \ \mid \ \boldsymbol{\Delta}(\mathtt{exp}, \mathtt{p}) \bowtie \eta \ \mid \ \neg\varphi \ \mid \ \varphi \wedge \varphi \ \mid \ \varphi \, \mathcal{U}^{I} \, \varphi$$

Evaluated over an evolution sequence $\mathcal{S}$ and a time instant τ_0

RobTL formulae

$$\varphi \ ::= \top \ \mid \ \boldsymbol{\Delta}(\mathrm{exp}, \mathrm{p}) \bowtie \eta \ \mid \ \neg\varphi \ \mid \ \varphi \wedge \varphi \ \mid \ \varphi \, \mathcal{U}^{\mathrm{I}} \, \varphi$$

Evaluated over an evolution sequence $\mathcal{S}$ and a time instant τ_0

Atomic formulae

RobTL formulae

$$\varphi \ ::= \top \ \mid \ \Delta(\mathtt{exp}, \mathtt{p}) \bowtie \eta \ \mid \ \neg\varphi \ \mid \ \varphi \wedge \varphi \ \mid \ \varphi \, \mathcal{U}^{\mathrm{I}} \, \varphi$$

Evaluated over an evolution sequence $\mathcal{S}$ and a time instant τ_0

Atomic formulae

- `exp`: RobTL expression defining the distance we want to analyse

RobTL formulae

$$\varphi \;::=\; \top \;\mid\; \Delta(\mathtt{exp}, \mathtt{p}) \bowtie \eta \;\mid\; \neg\varphi \;\mid\; \varphi \wedge \varphi \;\mid\; \varphi \,\mathcal{U}^{I}\, \varphi$$

Evaluated over an evolution sequence $\mathcal{S}$ and a time instant τ_0

Atomic formulae

- `exp`: RobTL expression defining the distance we want to analyse
- `p`: perturbation function applied to $\mathcal{S}$ at time τ_0

$$\mathtt{p} \;::=\; \mathtt{f@}\tau \;\mid\; \mathtt{p\,;\,p} \;\mid\; \mathtt{p}^{n}$$

RobTL formulae

$$\varphi \;::=\; \top \;\mid\; \boldsymbol{\Delta}(\mathtt{exp}, \mathtt{p}) \bowtie \eta \;\mid\; \neg\varphi \;\mid\; \varphi \wedge \varphi \;\mid\; \varphi\, \mathcal{U}^{I}\, \varphi$$

Evaluated over an evolution sequence $\mathcal{S}$ and a time instant τ_0

Atomic formulae

- `exp`: RobTL expression defining the distance we want to analyse
- `p`: perturbation function applied to $\mathcal{S}$ at time τ_0
- the threshold $\eta \in [0, 1]$ ($\bowtie \in \{<, \leq, \geq, >\}$)

$$\mathtt{p} \;::=\; \mathtt{f@}\tau \;\mid\; \mathtt{p\,;\,p} \;\mid\; \mathtt{p}^{n}$$

Software Tool for the Analysis of Robustness in the unKnown environment

The tool

Software Tool for the Analysis of Robustness in the unKnown environment

Source available at `https://github.com/quasylab/jspear`
Demo available at `http://quasylab.unicam.it/stark/`

The tool: STARK

- Specification language
  - agents
  - environment
  - RobTL expressions
  - perturbations
  - RobTL formulae
- Module for the simulation of evolution sequences and their perturbed versions
- Module for the evaluation of RobTL expressions
- Model checker for RobTL formulae

The tool: STARK

- Specification language
  - agents
  - environment
  - RobTL expressions
  - perturbations
  - RobTL formulae
- Module for the simulation of evolution sequences and their perturbed versions
- Module for the evaluation of RobTL expressions (including the evaluation of confidence intervals)
- Model checker for RobTL formulae (three-valued semantics)

Ongoing/future work

Ongoing/future work

- TempOral aNalYsis (TONY) module of STARK
- Application to biological systems and chemical networks

Ongoing/future work

- TempOral aNalYsis (TONY) module of STARK
- Application to biological systems and chemical networks

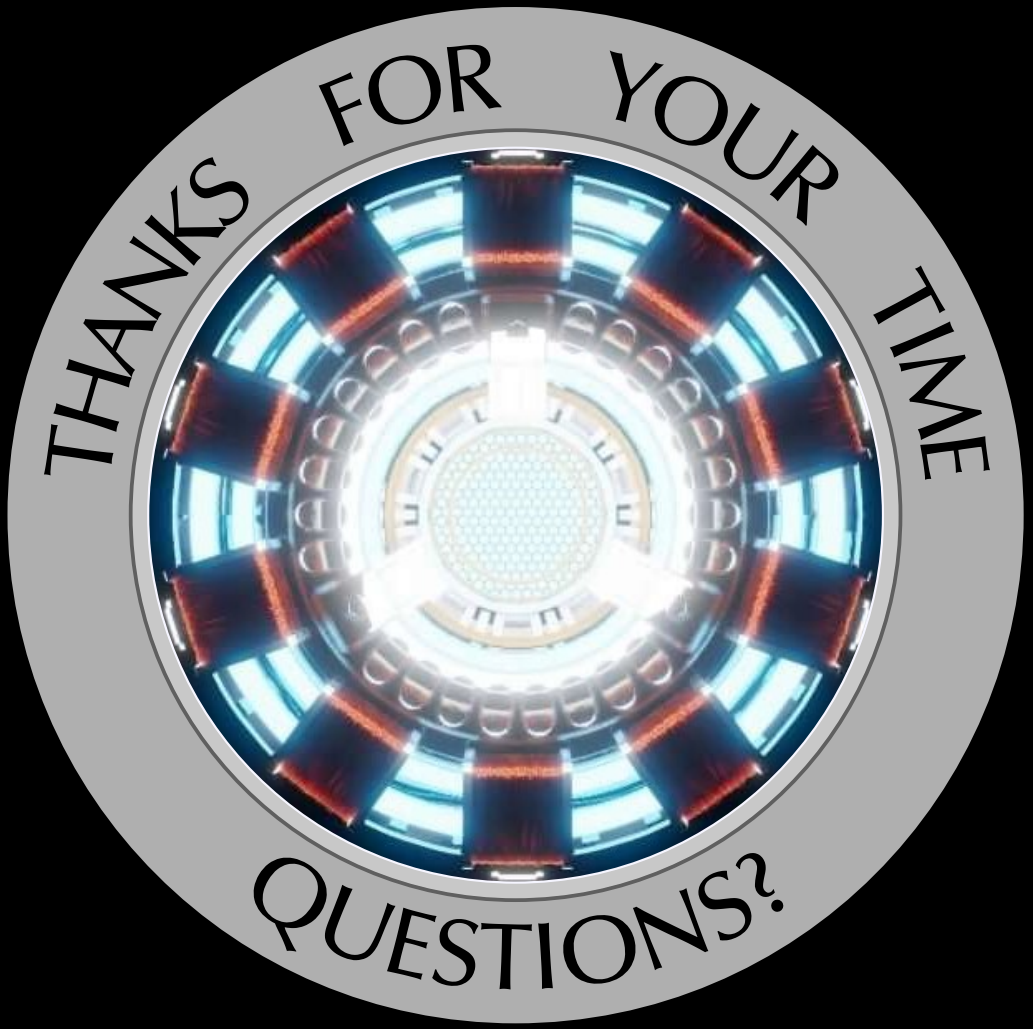
- Application to runtime monitoring
  - Synthesis of monitors from RobTL formulae
- Application to multi-agents systems

Ongoing/future work

- TempOral aNalYsis (TONY) module of STARK
- Application to biological systems and chemical networks

- Application to runtime monitoring
  - Synthesis of monitors from RobTL formulae
- Application to multi-agents systems

- Predictive monitoring?
- Formal framework for AI?
- Analysis of system performance?


THANKS FOR YOUR TIME
QUESTIONS?