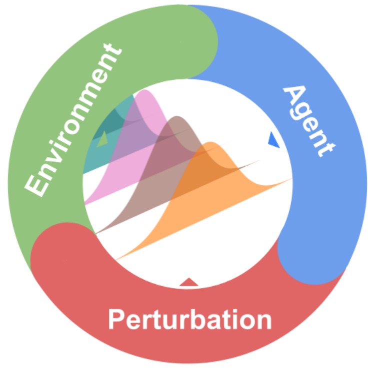
TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

STARK: Software Tool for the Analysis of Robustness in the unKnown environment

Sebastián Betancourt[1], Valentina Castiglioni[1], Michele Loreti[2], Simone Tini[3]

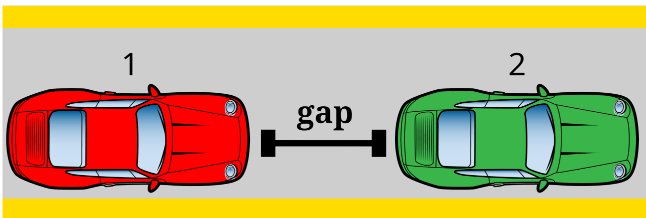
[1] Eindhoven University of Technology [2] University of Camerino [3] University of Insubria

STARK checks safe and correct behaviour of Cyber-Physical Systems under uncertainty.



- **Unreliable sensors/actuators/env** ⇒ State as a *distribution*
- **Computation + physics** ⇒ Agent–environment loop
- **Correctness** ⇒ Verify with **DisTL**
- **Robustness against perturbations** ⇒ Check with **RobTL**

Example: Car with unreliable components



$$\begin{aligned}
\text{Agent} &\stackrel{\text{def}}{=} \text{if } [\mathtt{gap} < 50]\ (\mathtt{FASTER} \to \mathtt{command}).\text{Agent} \qquad \text{(Car 1)}\\
&\quad \text{else if } [\mathtt{gap} > 50]\ (\mathtt{SLOWER} \to \mathtt{command}).\text{Agent}\\
&\quad\quad \text{else } (\mathtt{FASTER} \to \mathtt{command} \parallel_{0.5} \mathtt{SLOWER} \to \mathtt{command}).\text{Agent}
\end{aligned}$$

$$\begin{aligned}
\text{Env} \stackrel{\text{def}}{=} \{\ &\textbf{if } \mathtt{command} = \mathtt{FASTER} \textbf{ then } \mathtt{a1} \sim U[0, a_{max}]\\
&\textbf{else if } \mathtt{command} = \mathtt{SLOWER} \textbf{ then } \mathtt{a1} \sim -U[b_{min}, b_{max}];\\
&\mathtt{v2} \sim U[0, v_{max}];\\
&\ldots \quad \text{(Classical 1D kinematics)}
\end{aligned}$$

Does the red car maintain the safety gap *correctly?*

- Expected behaviour $\mu = \mathtt{gap} \sim N(50, 0.2)$
- Penalty function ρ and tolerance $\varepsilon \in [-1, 1]$

DisTL formula $\varphi = \Box^{[0,100]}\textbf{target}(\mu)^{\rho}_{\varepsilon}$

Is the system *robust against a sensor perturbation?*

- Sensor perturbation: $\mathrm{p} = (f @ 2)^{20}$
  applies f every 2 steps, 5 times, where $f(ds) = ds'$ and

$$ds'(\mathtt{gap}) = \begin{cases} ds(\mathtt{gap}) + \Delta, & \text{with probability } 0.5,\ \Delta \sim U([-10, 10]),\\ ds(\mathtt{gap}), & \text{otherwise.}\end{cases}$$

- $\mathrm{M}_{crash} = \mathtt{G}^{[0,100]} <^{\rho_{crash}}$ measures the maximum probability of a collision to occur in $[0, 100]$, where $\rho_{crash}(ds) = \begin{cases} 1 & \text{if } ds(\mathtt{gap}) \leq 0\\ 0 & \text{otherwise}\end{cases}$
- RobTL formula: Robustness against sensor perturbation

$$\psi = \Delta(\mathrm{M}_{crash}, \mathrm{p}_{dd}) \leq 0.1$$

Other case studies

- Simple physics:
  Three tanks experiment
- Robustness vs cyber-attacks:
  Overstress of an engine
- Effective digital twins:
  - Robots in industrial plant
  - Robots in smart hospital
- Systems Biology:
  - Gene regulatory networks
  - Biochemical networks
- Social networks:
  DeGroot learning
- Games:
  Polistil electric track

Formal System Analysis
Mathematics and Computer Science

V. Castiglioni, M. Loreti, & S. Tini. "STARK: A Software Tool for the Analysis of Robustness in the unKnown environment". COORDINATION 2023. https://doi.org/10.1007/978-3-031-35361-1_6
S. Betancourt, & V. Castiglioni. "On The Road Again (Safely): Modelling and Analysis of Autonomous Driving with Stark". ABZ 2025. https://doi.org/10.1007/978-3-031-94533-5_16