# Safety Shielding under Delayed Observation

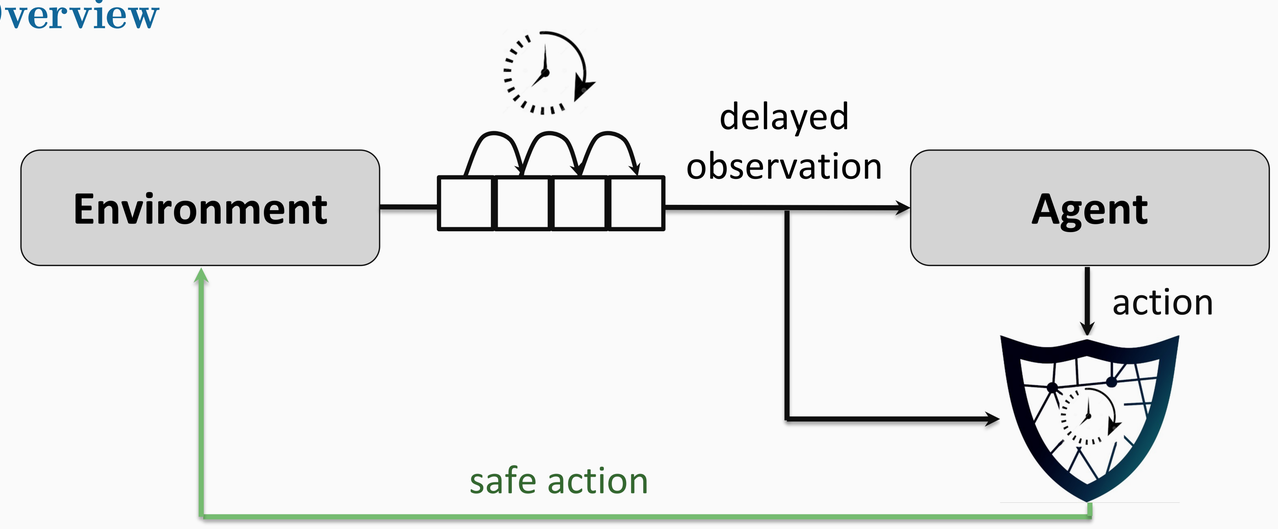
Filip Cano Córdoba[1], Alexander Palmisano[1],
Martin Fränzle[2], Roderick Bloem[1], Bettina Könighofer[1]

[1] Graz University of Technology, [2] Carl von Ossietzky University of Oldenburg

## Overview

Agents in physical environments face *delays in input signals* due to data transmission and sensing. Ignoring these delays in the safety analysis of actions can lead to critical *safety issues*.

- **Shielding.** A shield *enforces safety* by monitoring an agent and correcting its action during runtime. Shields are *computed automatically* from a safety specification and an environmental model.
- **Shielding under delays.** We present safety shields that account for *worst-case input delays*, ensuring safety even under delayed observations.




## Synthesis of Shields

- **Solve a safety game [1]**:
  - Interactions between environment and agent are modelled as a game. At every step, the environment picks an input and the agent picks an action.
  - The agent wins if *no safety-critical state* is visited during the play.
  - Compute *winning strategy* $\rho$ for agent such that no safety-critical state is ever visited.
- **Fix corrective action.** During runtime, shield enforces that all actions are contained in $\rho$.

## Synthesis of Shields under Delayed Inputs

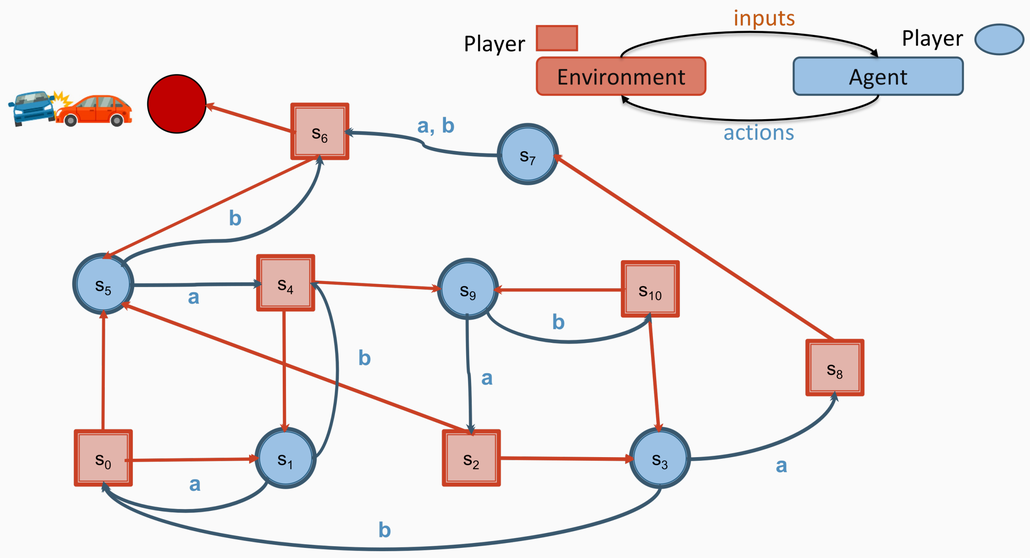
- **Solve a safety game under delays.** We encode a worst-case delay in the safety game, which induces imperfect state information [2].
- **Fix corrective action**. The delay-resilient shield allows all actions contained in the winning strategy of the delayed game. We propose two different heuristics of which corrective action to pick, with the objective to minimize shield interference:
  - Maximize *Controllability:* pick the action that maximizes the maximal delay on the input under which the agent stays safe.
  - Maximize *Robustness*: pick the action that maximizes the length in the game of the minimal path to an unsafe state.

## Safety Games with Delay

Safety-relevant interactions between the environment and the agent modeled as a two player game with discrete states.

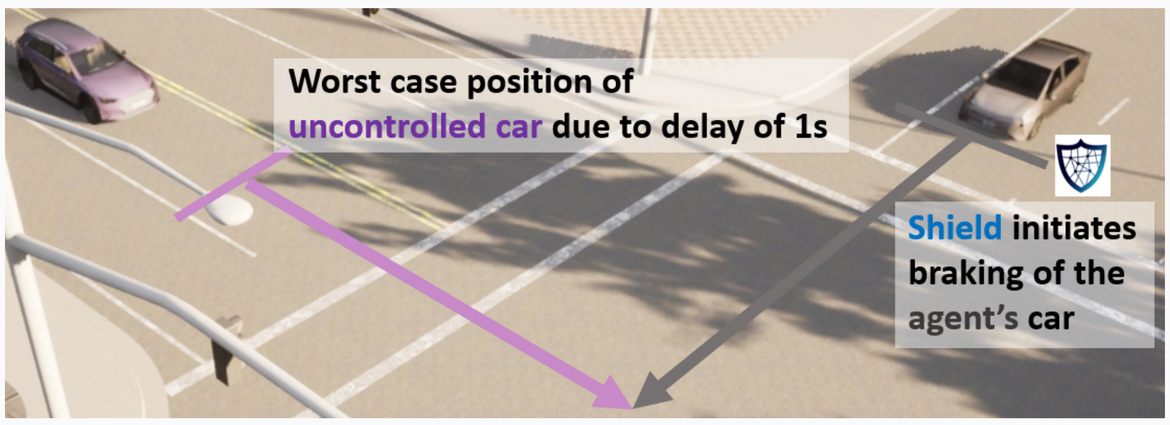
**Delayed setting:**

- The agent picks an action, without knowing the $\delta$ previous inputs.
- The agent has to ensure safety against any possible sequence of (unknown) environmental inputs.




## Case Study: Shielded Driving in Carla

Shielded ego car in driving simulator Carla, against collisions with cars and pedestrians.





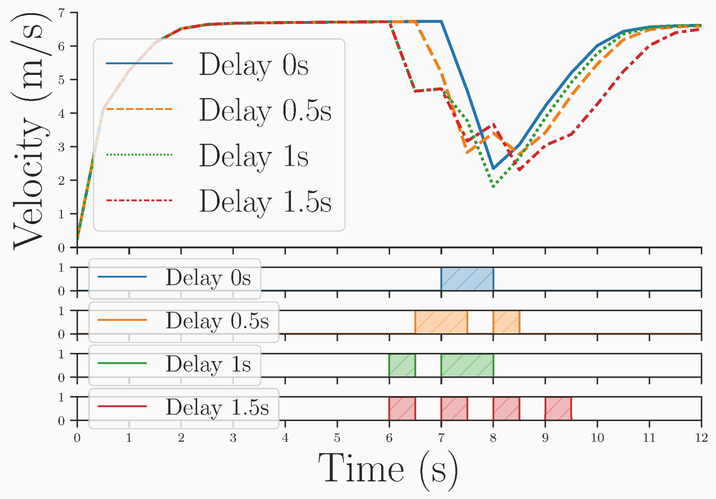
**Shielded behavior**:

- With high delay, shield acts intermitently: When worst-case assumption on other traffic participant's behavior does not happen
- With a higher delay, the uncertainty of the real locations of other traffic participants increases. Thus, the shield enforces higher distances.

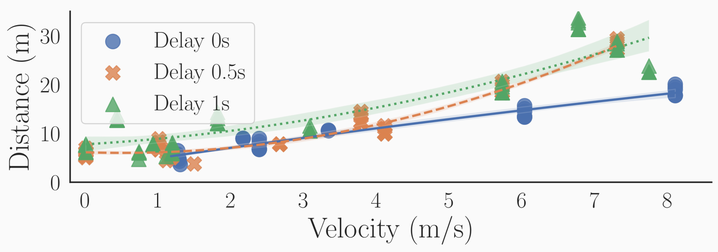
**Computational cost**:

- Shield is synthesized offline and implemented during runtime as a look-up table.
- Runtime overhead introduced by the shield is negligible.
- Shield synthesis cost grows with the maximum delay allowed:

Shield interventions for two-car intersections.




Shield interventions pedestrian crosswalk




Shield synthesis times

| Delay (in s) | | 0 | 0.5 | 1 | 1.5 |
|---|---|---|---|---|---|
| Synhesis times (in s) | Car example | 1.5 | 13 | 48 | 167 |
| | Pedestrian example | 0.8 | 9 | 34 | 119 |

## References

[1] Könighofer *et al.* "*Shield Synthesis*". FMSD (2017).
[2] Chen *et al.* "*Indecision and delays are the parents of failure*". Acta Informatica (2020).

## Future Work

- Delay-resilient shields on probabilistic models.
- Delay-resilient shields on continuous-state models.