Synthesis of Supervisory Controllers for Collision Avoidance Using DES Abstractions

Eric Dallal¹, Alessandro Colombo², Domitilla Del Vecchio³, Stéphane Lafortune¹

Motivation

- Vehicle collisions cause, on average, 4,156 injuries and 84 deaths per day in the United States. About a quarter of all reported light vehicle fatalities are due to side impacts, suggesting collisions at traffic intersections and merges.
- We therefore seek an automated mechanism to reduce the frequency of vehicular collisions at intersections. We assume that the vehicles are linked by some form of wireless communication.

Objectives

We wish to synthesize a supervisor for the vehicles which can deal with disturbances (e.g., snow, rain) as well as uncontrolled vehicles (e.g., vehicles that do not have the system onboard or whose wireless communication has failed) and which satisfies the following three conditions:
- **Safety:** collisions must be avoided.
- **Non-Blockingness:** vehicles should not deadlock and must fully cross the intersection.
- **Maximal Permissiveness:** The supervisor should leave as much autonomy to the driver as possible.

Solution Strategy

Synthesizing a solution directly at the continuous level is computationally infeasible. Our solution consists of the following four steps (Fig. 1):
- Abstraction/Modeling
- Translation of specifications from the continuous to the discrete event (DE) level
- Solution computation at the DE Level
- Translation of solution back to continuous domain

Technical Details

- **Abstraction/Modeling:** We discretize the system in space and in time (Fig. 2), resulting in a discrete event system and a finite solution domain. We model the disturbance and uncontrolled vehicles using uncontrollable event transitions.
- **Translation of specifications from the continuous to the discrete event (DE) level:** We translate safety by allowing only DE level transitions that could not correspond to a safety violation of the unabstracted system and we translate non-blockingness by marking states where all vehicles have crossed the intersection and requiring that executions reach these marked states.
- **Solution computation at the DE Level:** We solve for a maximally permissive DE supervisor by computing the supremal controllable sublanguage in the non-blocking case.
- **Translation of solution back to continuous domain:** We use a zero order hold to obtain a continuous domain supervisor.

Results

- The final synthesized continuous level supervisor can be proven to be correct (i.e., safe and non-blocking) and will be maximally permissive with respect to the chosen discretization.
- We have created a visual simulator in Java, which can compute solutions for 5-6 vehicles (Fig. 3).

Future Work

- Algorithmic improvements
- Second order dynamics
- Imperfect state information
- Acceleration constraints


Acknowledgement: Research supported in part by NSF grant CNS-0930081.