CPS: Synergy: Collaborative Research: Collaborative Vehicular Systems

Ümit Özgüner[1], Georgios Fainekos[2], Keith A. Redmill[1], Füsun Özgüner[1], Arda Kurt[1], Theodore P. Pavlic[2]

- [1]The Ohio State University, [2]Arizona State University
- citr.osu.edu, www.public.asu.edu/~gfaineko/
- ozguner.1@osu.edu, fainekos@asu.edu
- Awards #1446735, #1446730

THE OHIO STATE UNIVERSITY

ARIZONA STATE UNIVERSITY

Description

Challenge:

- Fully and partially self-driving vehicles, interacting with conventional vehicles on a complex road grid
- The need to develop rules to study and methods to coordinate a network of these

Focus:

- Collaboration
- Scalability
- Testability and Verifiability

Scientific Impact:

- Investigation of scalability questions for a more general class of CPS architecture
- Integration of different concerns such as safety, security and collaboration
- Ways of testing and verifying the above concerns in a meaningful CPS environment
- Methods to estimate, track, predict the behavior of multiple agents in a CPS setting
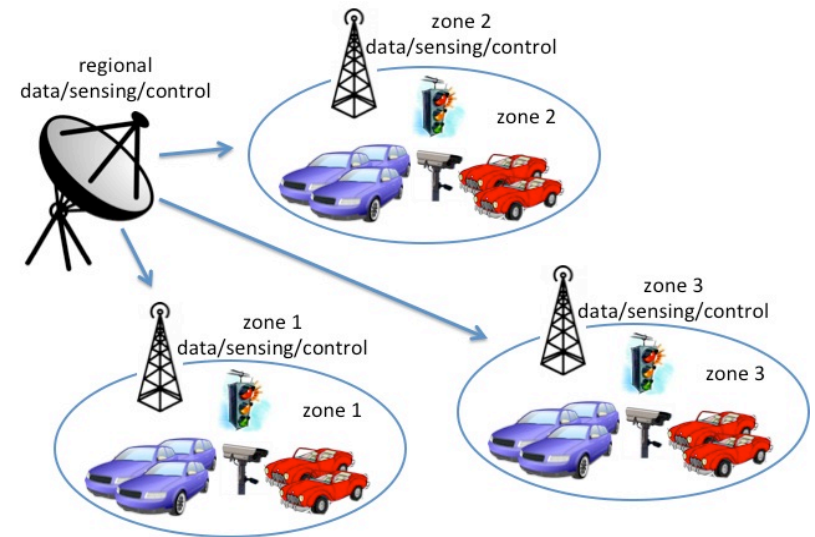
Broader Impact:

- Safer vehicles and more efficient collaborative driving on roadways

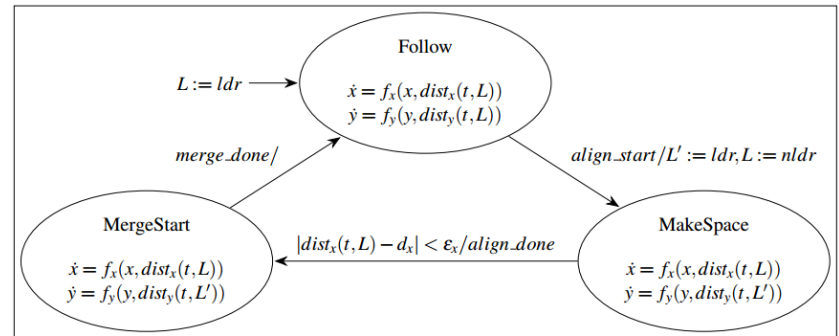THE OHIO STATE UNIVERSITY   ARIZONA STATE UNIVERSITY

Findings (Collaborative work)
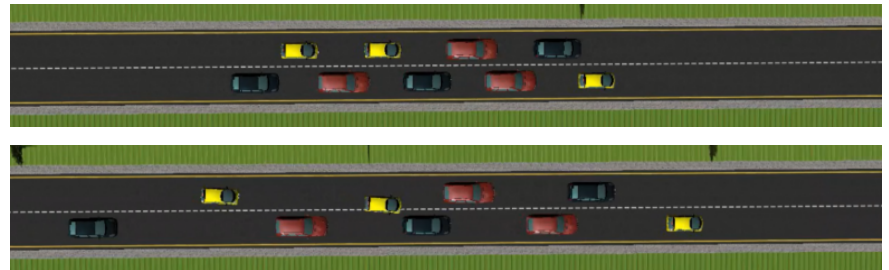
Developed a formal modeling framework for collaborative vehicles

High level defined with π-calculus expressions

1: $\text{Wait}(y) = y.\overline{merge\_done}$
2: $\text{Align}(y) = \overline{align\_start}.align\_done.\overline{y}.\text{Wait}$
3: $\text{Rcv\_Ldr}(y, ldr) = y(nldr).\overline{set\_ldr}<nldr>.\text{Align}(y)$
4: $\text{Send\_Ldr}(y) = get\_ldr(ldr).\overline{y}<ldr>.\text{Rcv\_Ldr}(y, ldr)$
5: $\text{Respond}(y, flag) = flag : [True \Rightarrow \text{Send\_Ldr}(y)]$
6: $\text{Ident}(y) = get\_id(id).\overline{y}<id>.y(flag).\text{Respond}(y, flag)$
7: $\text{Cooperate} = !\mathbf{r}(x).(\nu y)(\overline{x}<y>.\text{Ident}(y))$
8: $\text{Follow} = \overline{keep\_dist}.\text{Follow}$
9: $\text{Follower} = \text{Follow} || \text{Cooperate}$
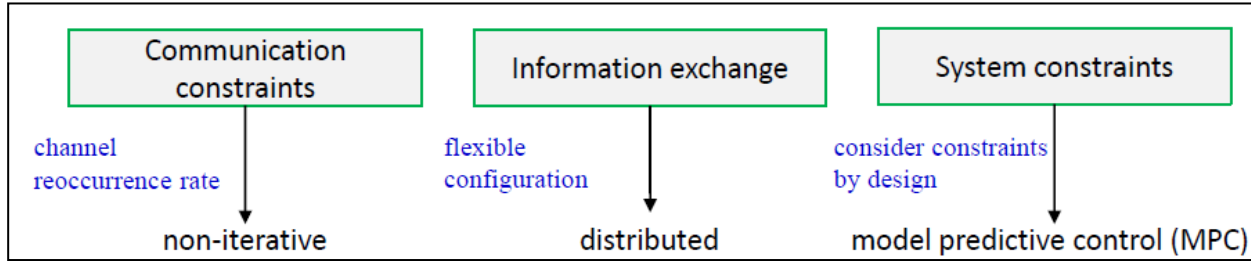
Low level defined with hybrid automata
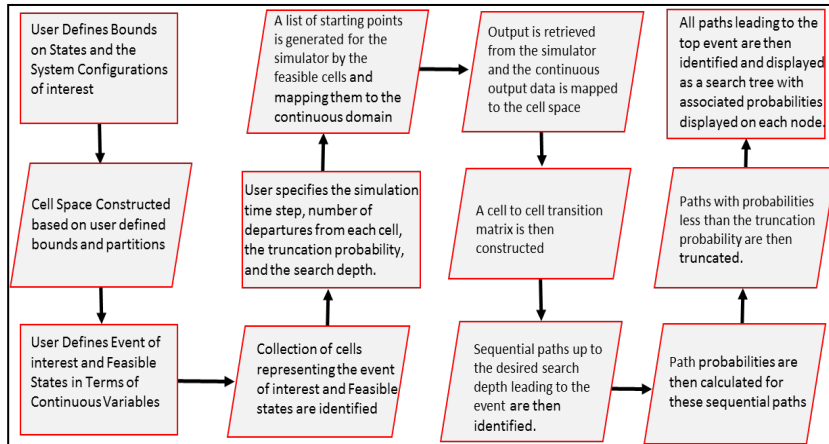

(b) Follower HA
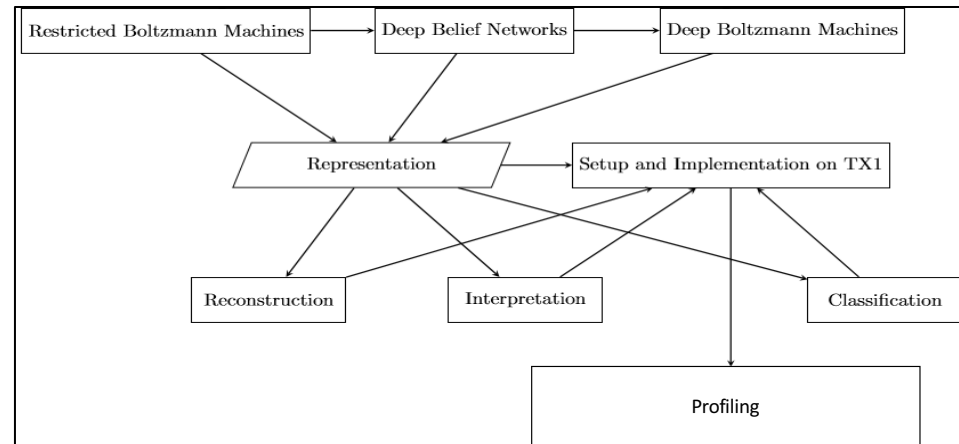
Findings

Collaboration via DMPC

Communication constraints

Information exchange

System constraints

channel reoccurrence rate

flexible configuration

consider constraints by design

non-iterative

distributed

model predictive control (MPC)

Experiment at OSU

Validation via Functional Hierarchies and Backtracking

User Defines Bounds on States and the System Configurations of interest

A list of starting points is generated for the simulator by the feasible cells and mapping them to the continuous domain

Output is retrieved from the simulator and the continuous output data is mapped to the cell space

All paths leading to the top event are then identified and displayed as a search tree with associated probabilities displayed on each node.

Cell Space Constructed based on user defined bounds and partitions

User specifies the simulation time step, number of departures from each cell, the truncation probability, and the search depth.

A cell to cell transition matrix is then constructed

Paths with probabilities less than the truncation probability are then truncated.

User Defines Event of interest and Feasible States in Terms of Continuous Variables

Collection of cells representing the event of interest and Feasible states are identified

Sequential paths up to the desired search depth leading to the event are then identified.

Path probabilities are then calculated for these sequential paths

Real-Time Traffic Scene Perception via Deep Learning

Restricted Boltzmann Machines

Deep Belief Networks

Deep Boltzmann Machines

Representation

Setup and Implementation on TX1

Reconstruction

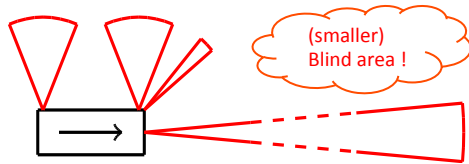Interpretation

Classification

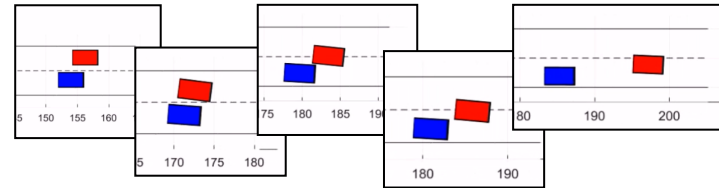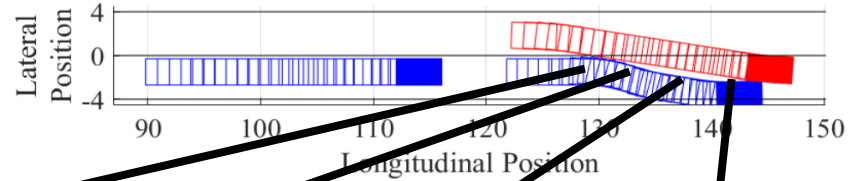Profiling

Findings

Automated testing
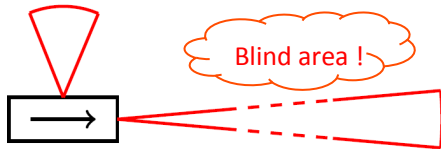
(Poor sensor configuration)

(Improved sensor configuration)

(A front collision right after avoiding a side collision)

(Non-robust scenario automatically created by our framework under the new sensor configuration)

Automated planning for vehicle coordination