Preliminary Results on Correct-by-Construction Control Software Synthesis for Adaptive Cruise Control

Petter Nilsson[1], **Omar Hussien**[2], Yuxiao Chen[1], Ayca Balkan[2], Matthias Rungger[2], Aaron Ames[3], Jessy Grizzle[1], Necmiye Ozay[1], Huei Peng[1], Paulo Tabuada[2]

1. University of Michigan
2. University of California at Los Angeles
3. Texas A&M University

December 15, 2014

Outline

Outline

Introduction

Problem setup

Solution: Continuous reachability computation

Solution: Discrete abstraction [PESSOA]

Simulation results

Comparison and Conclusion

Adaptive Cruise Control

- Maintains set speed when no lead car present.
- Follows lead car at a safe distance.

Formal methods in automotive safety

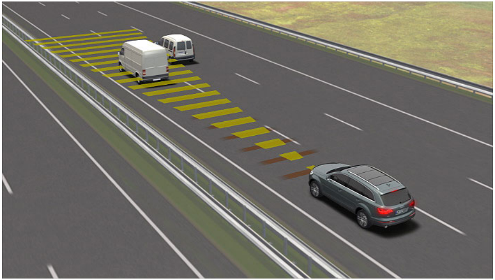


Image credit: Audi

Why formal methods?

- Safety and dynamics intimately connected.
- Explicit characterization of performance limits.
- Enables formal composition of components.

Outline

Introduction

Problem setup

Solution: Continuous reachability computation

Solution: Discrete abstraction [Pessoa]

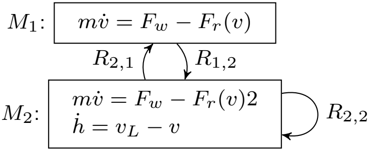
Simulation results

Comparison and Conclusion

Car model:

$$m\dot{v} = F_w - F_r(v), \quad F_r \text{ polynomial.} \tag{1}$$

Two-car system modeled as hybrid system with two modes M_1 (no lead car) and M_2 (lead car):



Reset maps:

- Lead car leaves: use reset map $R_{2,1}$.
- Lead car cuts in: use reset map $R_{1,2}$.
- New lead car: use reset map $R_{2,2}$.

Specifications

English specifications:

Set/LTL specifications:

[1] The time headway, or time to collision, is $\tau = h/v$.

Specifications

English specifications:

1. Satisfy the input constraint at all times.

Set/LTL specifications:

1. $\square(F_w \in \underbrace{[-0.3mg, 0.2mg]}_{S_2})$

[1] The time headway, or time to collision, is $\tau = h/v$.

Specifications

English specifications:

1. Satisfy the input constraint at all times.
2. When no lead car, reach and maintain v_{des}.

Set/LTL specifications:

1. $\square(F_w \in \underbrace{[-0.3mg, 0.2mg]}_{S_2})$
2. $\square(\square M_1 \implies \Diamond\square(v \in \underbrace{[v^-, v^+]}_{G_1}))$

[1] The time headway, or time to collision, is $\tau = h/v$.

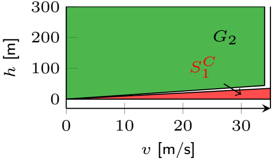
Specifications

English specifications:

1. Satisfy the input constraint at all times.
2. When no lead car, reach and maintain v_{des}.
3. In the presence of a lead car, reach and maintain lower bound on time headway[1] τ_{des} and upper bound on velocity v_{des}. Always satisfy time headway greater than 1 s.

Set/LTL specifications:

1. $\Box(F_w \in \underbrace{[-0.3mg, 0.2mg]}_{S_2})$
2. $\Box(\Box M_1 \implies \Diamond\Box(v \in \underbrace{[v^-, v^+]}_{G_1}))$
3. $\Box\left(\Box M_2 \implies \Diamond\Box(v,h) \in G_2\right) \wedge \Box\left(M_2 \implies (v,h) \notin S_1^C\right)$



[1] The time headway, or time to collision, is $\tau = h/v$.

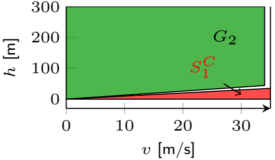
Specifications

English specifications:

1. Satisfy the input constraint at all times.
2. When no lead car, reach and maintain v_{des}.
3. In the presence of a lead car, reach and maintain lower bound on time headway[1] τ_{des} and upper bound on velocity v_{des}. Always satisfy time headway greater than 1 s.

Set/LTL specifications:

1. $\Box(F_w \in \underbrace{[-0.3mg, 0.2mg]}_{S_2})$
2. $\Box(\Box M_1 \implies \Diamond\Box(v \in \underbrace{[v^-, v^+]}_{G_1}))$
3. $\Box\left(\Box M_2 \implies \Diamond\Box(v,h) \in G_2\right)$ $\wedge \Box\left(M_2 \implies (v,h) \notin S_1^C\right)$



Complete specification:

$$\Box\left((M_1 \vee S_1) \wedge S_2\right) \wedge \Box\left(\bigwedge_{i=1}^{2} \Box M_i \implies \Diamond\Box G_i\right) \qquad (2)$$

[1]The time headway, or time to collision, is $\tau = h/v$.

Assumptions and Problem Statement

1. The velocity of the lead car is known and constant, i.e. $\dot{v}_L = 0$ (general problem has been solved, videos later).

Assumptions and Problem Statement

1. The velocity of the lead car is known and constant, i.e. $\dot{v}_L = 0$ (general problem has been solved, videos later).
2. There is at most one lead car within the radar range at all times (technical assumption, details in paper).

Assumptions and Problem Statement

1. The velocity of the lead car is known and constant, i.e. $\dot{v}_L = 0$ (general problem has been solved, videos later).
2. There is at most one lead car within the radar range at all times (technical assumption, details in paper).

Reset maps must be constrained to ensure that a correct controller exists (a lead car can not cut in too close).

Assumptions and Problem Statement

1. The velocity of the lead car is known and constant, i.e. $\dot{v}_L = 0$ (general problem has been solved, videos later).
2. There is at most one lead car within the radar range at all times (technical assumption, details in paper).

Reset maps must be constrained to ensure that a correct controller exists (a lead car can not cut in too close).

Problem statement

Under the preceding assumptions, synthesize a domain $\mathcal{D}$ and a controller for the hybrid dynamics, which enforces the specifications given that the reset maps are restricted to $\mathcal{D}$.

Outline

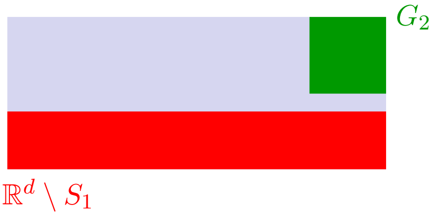
Continuous computation: Idea

Focus on the lead car mode M_2. To satisfy the given "reach-stay-while avoiding" specification:

- Find a **controlled-invariant** set $C_1 \subset G_2$.
- Compute a set of chains that reaches C_1 while staying in S_1.

Definition

Let Σ be the system $x(t+1) = f(x(t), u(t))$ with the constraint $u(t) \in U$. A set C is *controlled-invariant* for Σ if for all $x_0 \in C$ there is a $u_0 \in U$ such that $f(x_0, u_0) \in C$.



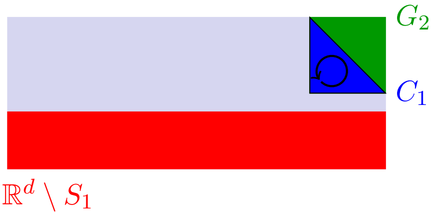
Continuous computation: Idea

Focus on the lead car mode M_2. To satisfy the given "reach-stay-while avoiding" specification:

- Find a **controlled-invariant** set $C_1 \subset G_2$.
- Compute a set of chains that reaches C_1 while staying in S_1.

Definition

Let Σ be the system $x(t+1) = f(x(t), u(t))$ with the constraint $u(t) \in U$. A set C is *controlled-invariant* for Σ if for all $x_0 \in C$ there is a $u_0 \in U$ such that $f(x_0, u_0) \in C$.



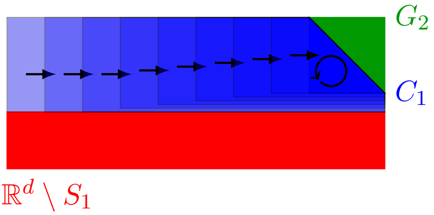
Continuous computation: Idea

Focus on the lead car mode M_2. To satisfy the given "reach-stay-while avoiding" specification:

- Find a **controlled-invariant** set $C_1 \subset G_2$.
- Compute a set of chains that reaches C_1 while staying in S_1.

Definition

Let Σ be the system $x(t+1) = f(x(t), u(t))$ with the constraint $u(t) \in U$. A set C is *controlled-invariant* for Σ if for all $x_0 \in C$ there is a $u_0 \in U$ such that $f(x_0, u_0) \in C$.



Continuous computation: Solution overview

1. Linearize and time-discretize system dynamics in a way that correctness is inherited by the original system.
   - Reachability in linear system must imply reachability in original system.
2. Use linear reachability computations to synthesize a correct-by-construction controller for the linearized system.
3. Implement controller on original system.

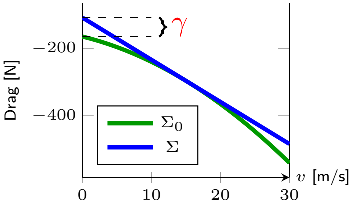
Linearization and time-discretization of system dynamics

We linearize and integrate the dynamics

$$\Sigma_0 : \begin{aligned} m\dot{v} &= F_w - f_0 - f_1 v - f_2 v^2, \\ \dot{h} &= v_L - v \end{aligned}$$

for a time Δ to obtain

$$\Sigma : \begin{bmatrix} v(t+\Delta) \\ h(t+\Delta) \\ v_L(t+\Delta) \end{bmatrix} = A \begin{bmatrix} v(t) \\ h(t) \\ v_L(t) \end{bmatrix} + B\bar{F}_w(t) + K.$$



Proposition

For a fixed $d : [0, \tau] \to D$, if there exists an input $\bar{F}_w : [0, \tau] \to [-0.3mg, 0.2mg - \gamma]$ that steers the state of Σ from (v^0, h^0, v_L^0) to (v^1, h^1, v_L^1), then there exists an input $F_w : [0, \tau] \to [-0.3mg, 0.2mg]$ that steers the state of Σ_0 from (v^0, h^0, v_L^0) to (v^1, h^1, v_L^1).

Reachability computations for affine systems I

Problem

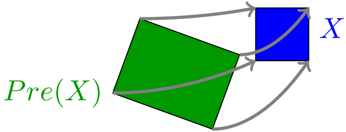
Given a discrete-time affine system

$$x(t+1) = Ax(t) + Bu(t) + K,$$

with (possibly state-dependent) input constraints

$$H_x^u x(t) + H_u^u u(t) \leq h^u$$

and a final set $X = \{x : Hx \leq h\}$*, we want to find the set of initial states* $Pre(X)$ *from where the system can be steered to* X*.*



Reachability computations for affine systems II

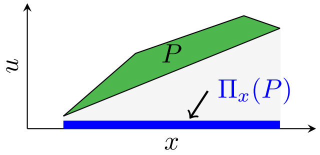
Solution

Write all constraints as a polyhedron in $x-u$-space:

$$P = \left\{ (x,u) : \begin{bmatrix} HA & HB \\ H_x^u & H_u^u \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \leq \begin{bmatrix} h - HK \\ h^u \end{bmatrix} \right\}.$$

Then $Pre(X)$ can be characterized as the x-projection of this polyhedron:

$$Pre(X) = \Pi_x(P).$$



Can be extended to multiple time steps and/or systems with disturbance. Projection can be computed with MPT [Herceg et al. (2013)].

Algorithms

With the reachability operator Pre, the required sets can be found for a 'reach-and-stay while avoiding' type specificatin as follows.

- A controlled-invariant set C_1 inside a goal set G is the fixed point of the operator

$$X \mapsto G \cap Pre(X)$$

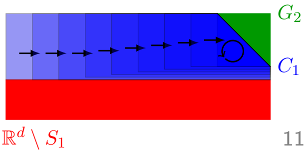
  initialized with G itself.
- To reach C_1 while staying in a safe set S, iterating

$$C_{i+1} = S \cap Pre(C_i)$$

  will give a list of sets $C_1 \leftarrow C_2 \leftarrow C_3 \leftarrow \dots$ such that C_i can be reached from C_{i+1} while staying in S.

Correct control strategy:

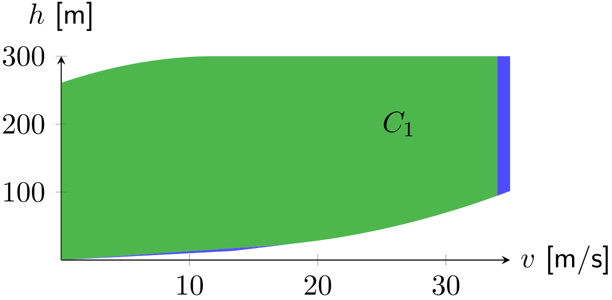
When in C_i, steer to $C_{\max(1,i-1)}$ in finite time.



Result for ACC

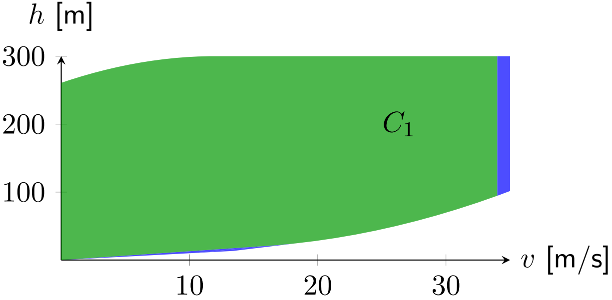
Result for ACC

Controlled-invariant set C_1 in $v - h$ space:



Result for ACC

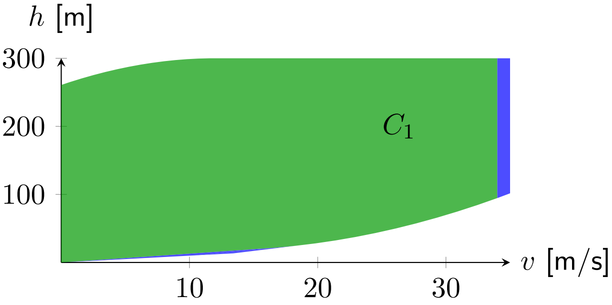
Controlled-invariant set C_1 in $v - h$ space:



C_1 and the sets from where it is reachable define the most relaxed safe assumptions on where cars can cut in.

Result for ACC

Controlled-invariant set C_1 in $v - h$ space:



C_1 and the sets from where it is reachable define the most relaxed safe assumptions on where cars can cut in.

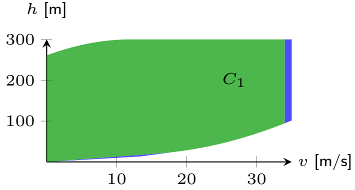
- How to move between the sets returned by the reachability algorithms?

Control strategy implementation: MPC

Use N-step horizon Model-Predictive Control (MPC) to implement low-level controller:

$$u^* = \begin{cases} \min & L(x,u), \\ \text{s.t.} & H_x^u x + H_u^u u \leq h^u, \\ & x \in C_i \end{cases}$$

Feasibility of the QP is guaranteed by construction.

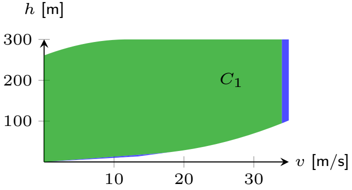


Control strategy implementation: MPC

Use N-step horizon Model-Predictive Control (MPC) to implement low-level controller:

$$u^* = \begin{cases} \min & L(x,u), \\ \text{s.t.} & H_x^u x + H_u^u u \leq h^u, \\ & x \in C_i \end{cases}$$

Feasibility of the QP is guaranteed by construction.

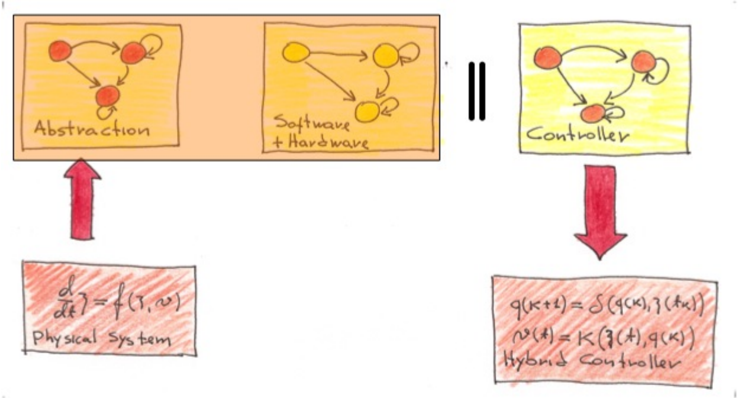


Composition with CCC

- Using any conventional cruise controller (CCC) for mode M_1 together with the presented controller for M_2 is correct if reset maps are restricted.
- For a general hybrid system, control domains need to be propagated back and forth, but in this case there is convergence in the first time step (thanks to the simplicity of CCC).

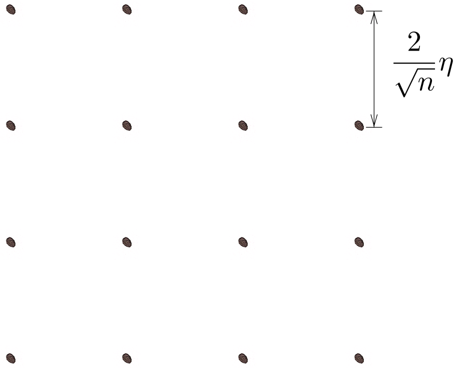
Outline

1. Finite-state abstraction;
2. Finite-state controller synthesis;
3. Controller refinement.



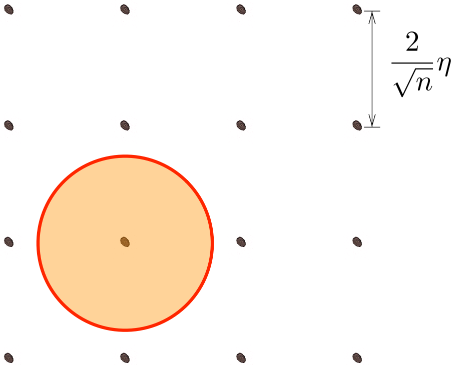
Abstraction Computation

Finite-state abstractions can be extracted from control systems using a very simple construction.



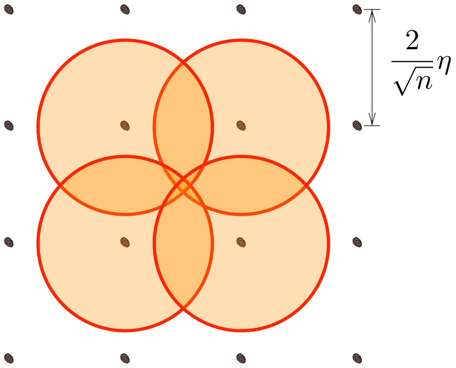
Abstraction Computation

Finite-state abstractions can be extracted from control systems using a very simple construction.



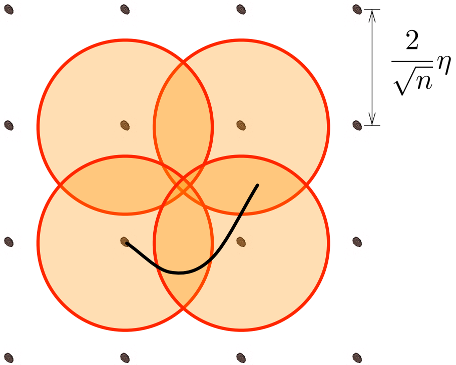
Abstraction Computation

Finite-state abstractions can be extracted from control systems using a very simple construction.



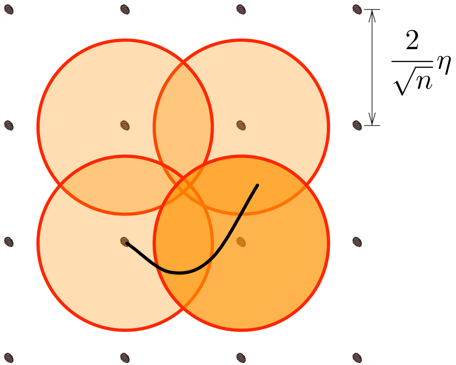
Abstraction Computation

Finite-state abstractions can be extracted from control systems using a very simple construction.



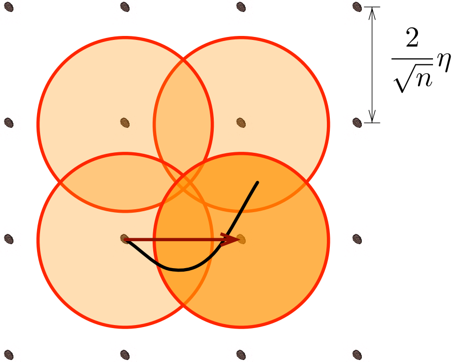
Abstraction Computation

Finite-state abstractions can be extracted from control systems using a very simple construction.



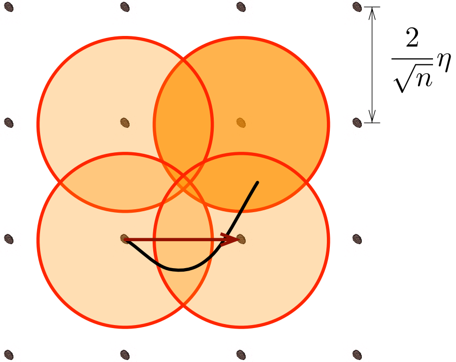
Abstraction Computation

Finite-state abstractions can be extracted from control systems using a very simple construction.



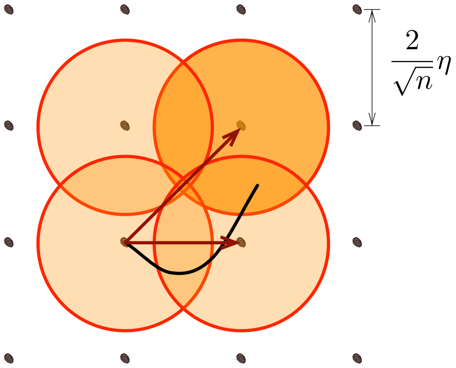
Abstraction Computation

Finite-state abstractions can be extracted from control systems using a very simple construction.



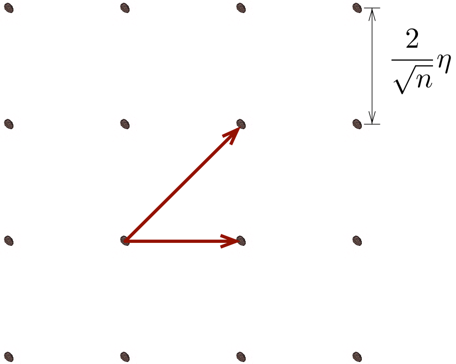
Abstraction Computation

Finite-state abstractions can be extracted from control systems using a very simple construction.

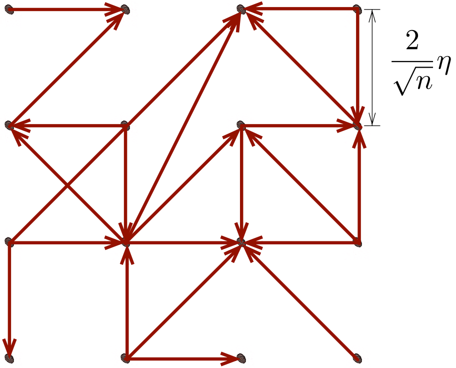


Abstraction Computation

Finite-state abstractions can be extracted from control systems using a very simple construction.



Finite-state abstractions can be extracted from control systems using a very simple construction.



Controller Synthesis

- Controller synthesis problem can be modeled as a game.
  - A game consists of
    1. Transition system abstraction
    2. Wining condition specification
    3. Players controller and environment

Controller Synthesis

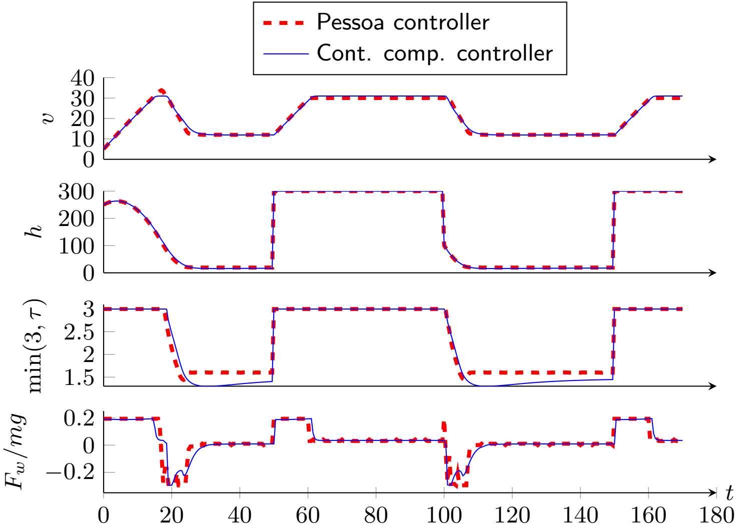
- Controller synthesis problem can be modeled as a game.
  - A game consists of
    1. Transition system abstraction
    2. Wining condition specification
    3. Players controller and environment
- Fixed-point algorithms are used to compute the wining set (controller domain).

Controller Synthesis

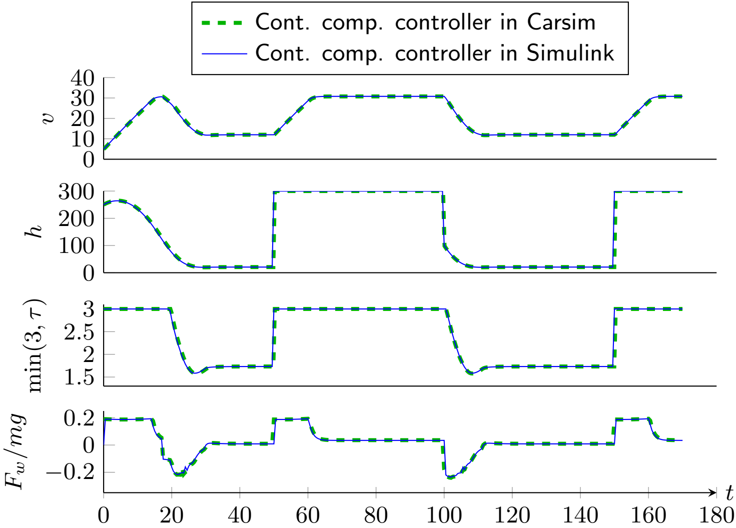
- Controller synthesis problem can be modeled as a game.
  - A game consists of
    1. Transition system abstraction
    2. Wining condition specification
    3. Players controller and environment
- Fixed-point algorithms are used to compute the wining set (controller domain).
- Termination is guaranteed by finiteness of the set of states.

Outline

Simulink simulation

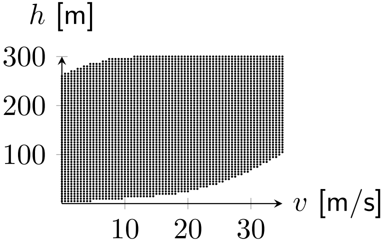


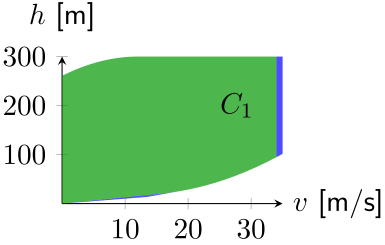
Carsim simulation



Carsim is a standard car simulator for Automotive industry.

Carsim simulation videos





Outline

Comparison

- Both methods employ fixed point algorithms on a continuous and discrete state space, respectively.
- Discrete state space: no linearization required, algorithm guaranteed termination.
- Continuous state space: no abstraction required, tuning flexibility.

Summary

Conclusions:

- Synthesized two correct-by-construction adaptive cruise controllers for a low-order model and a high-order (~ 30 states), industry-standard model.
- Used industry-accepted model and realistic parameters.
- Compared abstraction-based method with continuous reachability computation method.

Summary

Conclusions:

- Synthesized two correct-by-construction adaptive cruise controllers for a low-order model and a high-order (~ 30 states), industry-standard model.
- Used industry-accepted model and realistic parameters.
- Compared abstraction-based method with continuous reachability computation method.

Future work:

- Allow lead car to have variable speed (to appear).
- Implement on model cars.
- HIL simulation.
- Formal composition of interconnected systems equipped with correct-by-construction controllers.

Thank you for your attention.

References

M. Herceg, M. Kvasnica, C. N. Jones, and M. Morari.
Multi-Parametric Toolbox 3.0.
In *Proc. of the ECC*, 2013.