Getting Out of the Way – Safety Verification without Compromise

Theodore P. Pavlic, Sai Prathyusha Peddi,
Paolo A.G. Sivilotti, and Bruce W. Weide
Department of Computer Science & Engineering
The Ohio State University
{pavlic.3, peddi.2, sivilotti.1, weide.1}@osu.edu

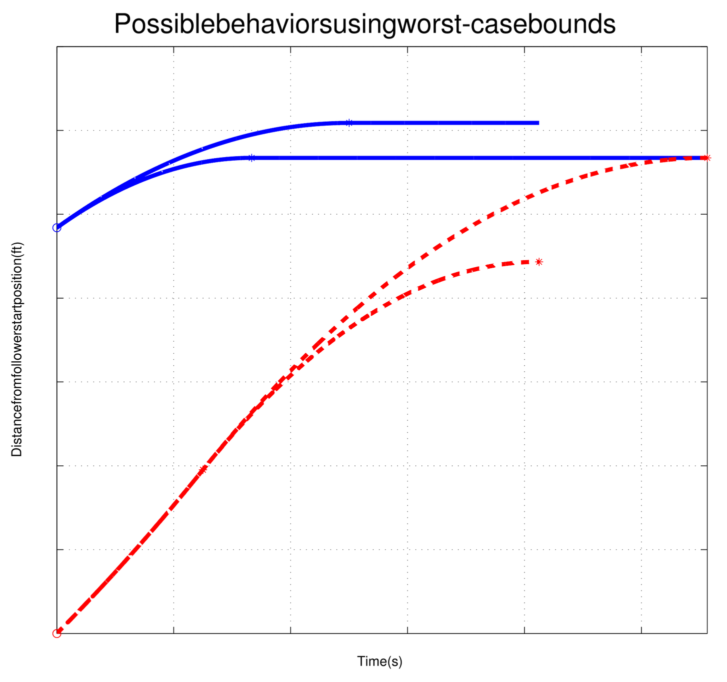
Abstract

Safety is often viewed as a quantity to be traded off for better performance. In our work, we look to provide safety verification tools and techniques that enhance rather than compromise on performance. Here, we examine two examples. Modern adaptive cruise control technologies are designed to improve the comfort or safety of the driver; however, no safety guarantees are asserted by these designs. Furthermore, existing theoretical work in the safety verification of adaptive cruise control algorithms require both discrete braking modes and overly conservative separation distances to make such safety guarantees. Thus, existing work in safety verification both risks reducing driver comfort while also eliminating any of the performance gains typically associated with automated highways. Our work extends verification of automated highway systems to mitigate both of these problems. Motivated by optimal control and verification of software systems, we have developed safety conditions for adaptive cruise control algorithms that do not require discontinuous braking and also allow for substantially lower following distances than existing work in the verification of autonomous highway systems. Moreover, we demonstrate a novel approach for verifying software in hybrid systems by embedding the continuous dynamics into the software specifications. The result is a verified software paradigm consistent with the vision of Hoare's verifying compiler. Finally, we shift gears to consider how variable yellow timing and a new encoding of traffic light signals can be used to guarantee safe intersections that also reduce fuel consumption.
***Keywords:** verification; hybrid systems; safety; adaptive cruise control; signal coordination and timing; yellow light*

Adaptive Cruise Control (ACC)

Conventional Verification of ACC: WCS

- Assume *global* upper and lower braking bounds
- Assume worst-case scenario (WCS)
  - Leader uses *strongest* braking behavior
  - Follower uses *weakest* braking behavior
- Safe distance grows with braking interval

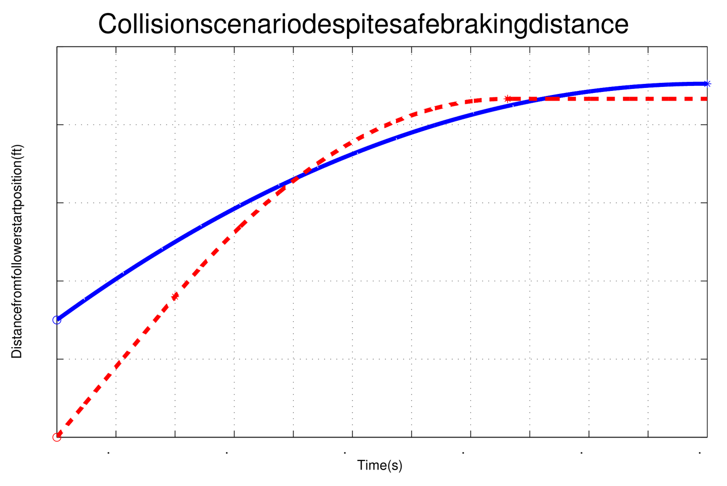


Specifications using worst-case stopping distances

- Worst-case and best-case scenarios are shown. In each:
  - Leader is solid blue
  - Follower is dashed red
- WCS is depicted by intersecting lines
  - Automated proof of safety is relatively simple
- For realizations where follower braking is *controlled* as opposed to *unknown*, the WCS safe-braking distance is **overly conservative**

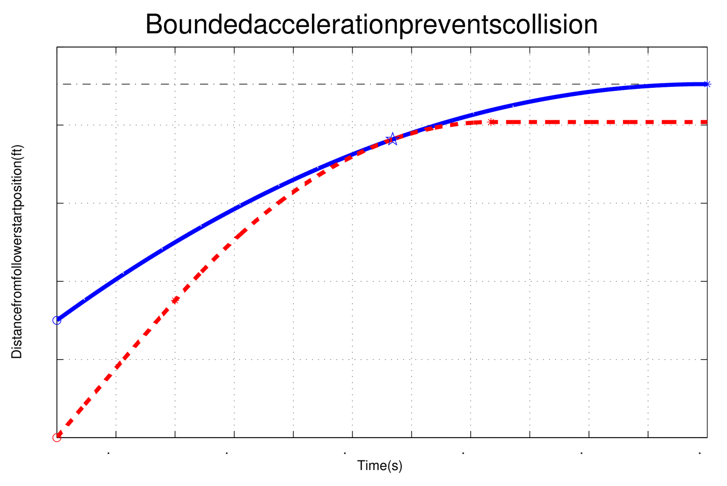
Heterogeneous ACC with Braking Control

- Follower's weakest braking bound is controlled
- Upper bound on each leader is inferred
  - e.g., plate tag indicates braking category
- Safety does not follow from stopping *distances*

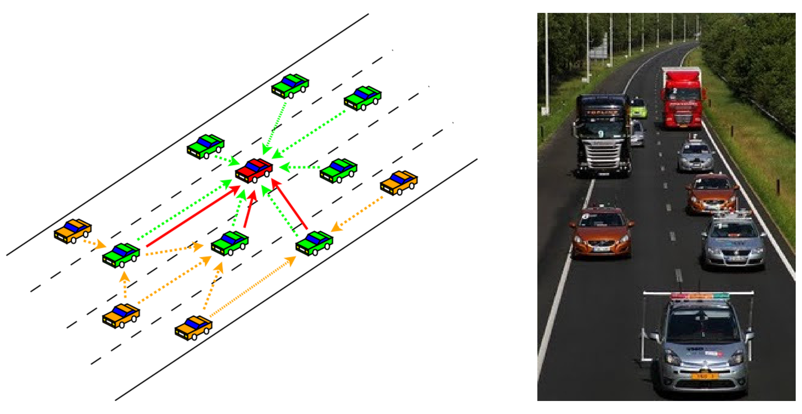


Collision Although Safe Stopping-Distance

- Non-trivial braking constraints for follower
  - **Verification is more challenging**



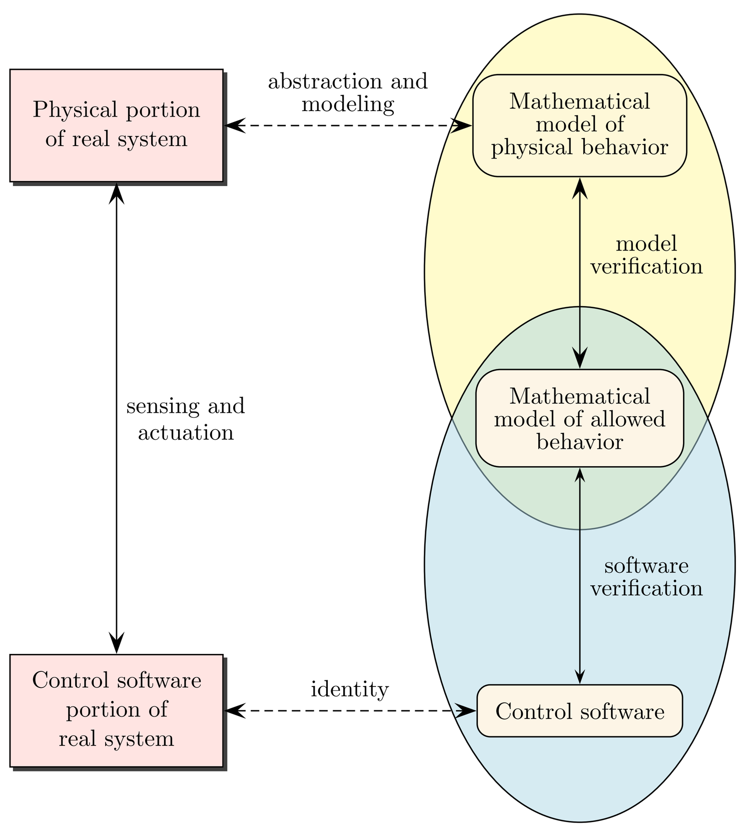
Marginally Safe Stop after Evasive Acceleration



Mixed-Traffic Adaptive Cruise Control

Verifying Cyber-Physical Systems

Conventional verification: either *model* **or** *software*



CPS Concrete–Abstract Correspondence

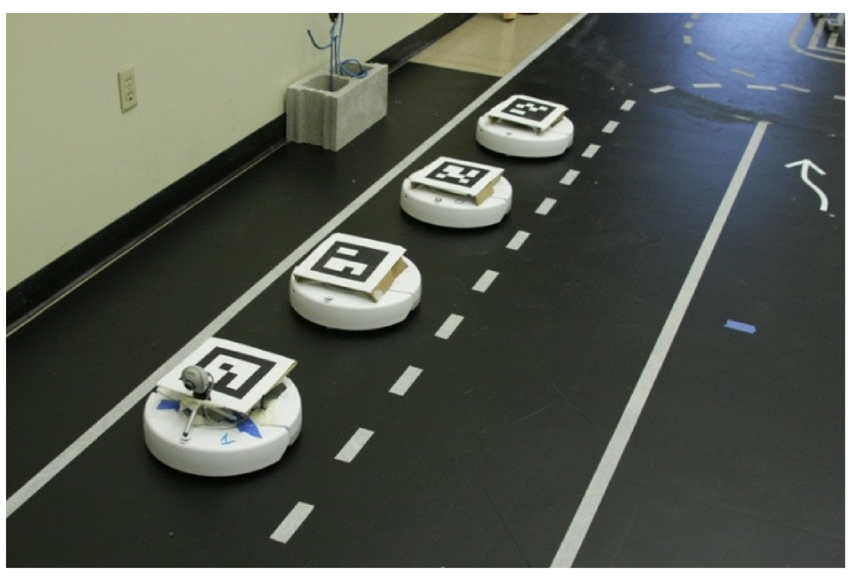
To combine the two:

```
havoc dt
assume  0.0 < dt  and  dt < rho

physical loop
  maintains
    bl = #bl  and  bf = #bf  and
    afMax = #afMax  and rho = #rho  and
    af = #af  and dt = #dt  and
    0.0 <= t  and  t < rho + dt  and
    vl = VEL(#vl, -bl, t)  and
    xl = POS(#xl, #vl, -bl, t)  and
    vf = VEL(#vf, af, t) and
    xf = POS(#xf, #vf, af, t)  and
    xl >= xf
while IsGreater (rho, t) do
  variable zero, dv, dx: Real

  dv := Replica (dt)
  Multiply (dv, bl)
  Subtract (vl, dv)
  if IsGreater (zero, vl) then
      Clear (vl)
  end if
  dx := Replica (dt)
  Multiply (dx, vl)
  Add (xl, dx)

  dv := Replica (dt)
  Multiply (dv, af)
  Add (vf, dv)
  if IsGreater (zero, vf) then
      Clear (vf)
  end if
  dx := Replica (dt)
  Multiply (dx, vf)
  Add (xf, dx)

  Add (t, dt)
end loop
```

Augment Annotated Code with Physical Loop

```
Prove:
VEL (vl_4, −bl_0, t_11) − dt_9 × bl_0
   = VEL (vl_4, −bl_0, t_11 + dt_9)
Given:
0.0 < bl_0
0.0 < bf_0
0.0 < afMax_0
bf_0 ≤ bl_0
0.0 < rho_0
MINGAP (vl_2, bl_0, vf_2, bf_0, afMax_0, rho_0)
   ≤ xl_2 − xf_2
0.0 ≤ vl_2
0.0 ≤ vf_2
0.0 ≤ vl_4
0.0 ≤ vf_4
MINGAP (vl_4, bl_0, vf_4, bf_0, afMax_0, rho_0)
   ≤ xl4 − xf4 −bf_0 ≤ af_8
af_8 ≤ afMax_0
MINGAP (VEL (vl_4, −bl_0, rho_0),
        bl_0, VEL (vf_4, af_8, rho_0),
        bf_0, afMax_0, rho_0)
   ≤ POS (xl_4 − xf_4, vl_4, −bl_0, rho_0)
   − POS (0.0, vf_4, af_8, rho_0)
0.0 < dt_9
dt_9 < rho_0
t_11 < rho_0
0.0 ≤ t_11
t_11 < rho_0 + dt_9
POS (xf_4, vf_4, af_8, t_11)
   ≤ POS (xl_4, vl_4, −bl_0, t_11)
0.0 ≤ VEL (vl_4, −bl_0, t_11) ≤ dt_9 × bl_0
VEL (vf_4, af_8, t_11) + dt_9 × af_8 < 0.0
```

Example Verification Condition (VC)



Urban Traffic Control

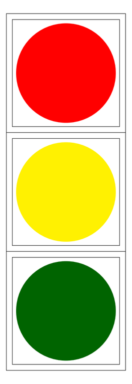
Safe Efficient Intersection Crossing Guards

With little vehicle-to-infrastructure communication, physical inertia can be used to project mutual exclusion distances (i.e., reachability sets).

- Signalling guards can be designed that maintain safety with minimal yellow-time losses.
- The system can operate in mixed environments of human and autonomous drivers.
- With introduction of one additional signal color, fuel efficiency can be improved by preventing unnecessary deceleration.

Safety invariants are guaranteed, which provides the opportunity for **separating designs for safety and optimization.**

Yellow Guards for Variable Cycle Times



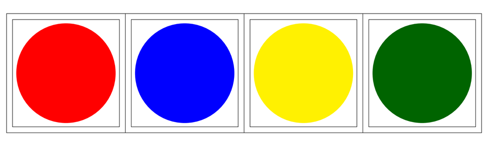
Goal: Minimize simultaneous red

Access direction
<crossing guard> → <light>
$t > 0 \vee x + v^2/(2b) \leq i_x \rightarrow$ green
$t \leq 0 \wedge x + v^2/(2b) > i_x \rightarrow$ yellow

(a simplified example)

Yellow light times are minimized while still allowing cycle times to vary to ensure safety invariants.

Blue Lights for Fixed Cycle Times



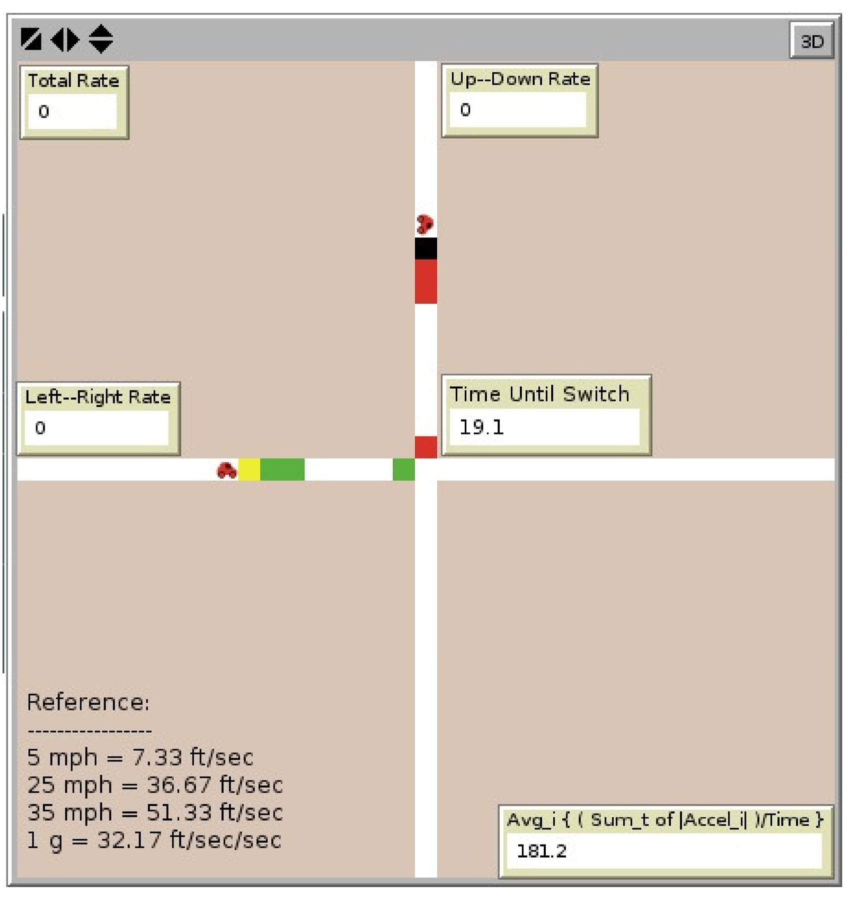
Goal: Maximize simultaneous green

Blocked direction
<crossing guard> → <light>
$t_{\text{MUTEX}} > t \vee (t_{\text{ARR}} \geq t \wedge t_{\text{DEP}} < t + T) \rightarrow$ blue
$t_{\text{MUTEX}} \leq t \wedge (t_{\text{ARR}} < t \vee t_{\text{DEP}} \geq t + T) \rightarrow$ red

(an overly simplified example)

When cycle times are fixed, red lights waste the fuel for vehicles whose reachability sets are far from the intersection. Consequently, a blue color can be introduced to the blocked direction that indicates when braking behavior is not warranted.

Given Safety, Compare Performance



Once guards are provided that guarantee safety invariants, alternate switching protocols can be compared either theoretically or empirically to improve throughput or fuel performance of intersections.

Supported by NSF Grant ECCS-0931669.

mailto:pavlic.3@osu.edu

http://cps.osc.edu/