Simulation-based Adversarial Test Generation for Embedded Software in Autonomous Vehicles

Georgios Fainekos – EMSOFT 2018

\[ \Box (g=5 \land \omega < x) \rightarrow \Diamond_{[0,\tau]} g=4 \]

\[ \Box (\text{idle} \rightarrow \omega > 1100 \text{ RPM}) \]

\[ \Box (\text{turnoff} \rightarrow \Diamond_{[0,\tau]} \text{cc=off}) \]

\[ \Box (g \geq 1 \land \text{“other”} \rightarrow \omega_{\text{em}} > 0) \]
Autonomous cars are (almost) here!

Toyota

Waymo

BMW

Ford

Intel

Mercedes

nuTonomy

Volvo

Uber
Software errors in autonomous vehicles: How serious is this problem?
What are these software recalls?
Sampling of recalls from the 2011-12 period ...

- "A software error may prevent the transmission from downshifting, such as shifting from 5th to 4th gear when coasting, which may result in decreased engine RPMs and possible engine stall, increasing the risk of a crash."

- ... the software that “allows the ECU to establish a ‘handshake’ with the engine is in error. The ECU monitors certain driving conditions, and when found to be out of tolerance, the software picks up an anomaly. When this happens, the ECU triggers a fault code. As the ECU tries to find an optimal driving condition outside its prescribed tolerances, a rough idle or stalling situation ensues.”

- ... to update the software that controls the hybrid electric motor. Under certain circumstances, it is possible, according to the company, “...for the electric motor to rotate in the direction opposite to that selected by the transmission.”

- If the fault occurs, cruise control can only be disabled by turning off the ignition while driving - which would mean a loss of some control in many cars also disables power steering. Braking or pressing the cancel button will not work.

- ...
What are these software recalls?
Sampling of recalls from the 2011-12 period ...

- "A software error may prevent the transmission from downshifting, such as shifting from 5th to 4th gear when coasting, resulting in decreased engine RPMs and possible engine stall, increasing the risk of a crash."

- "The software that "allows the ECU to establish a 'handshake' with the engine is in error. The ECU monitors certain driving conditions and when the engine is found to be out of tolerance, the software picks up an anomaly. When this happens, the ECU triggers a fault code. As the ECU tries to find an optimal driving condition outside its prescribed tolerances, a rough idle or stalling situation ensues."

- "To update the software that controls the hybrid electric motor. Under certain circumstances, it is possible, according to the company, "for the electric motor to rotate in the direction opposite to that selected by the transmission."

- "If the fault occurs, cruise control can only be disabled by turning off the ignition while driving - which would mean a loss of some control - and braking or pressing the ‘cancel’ button will not work."

- "The engine should never stall while idle."

- "The electric motor should always rotate in the direction selected by the transmission."

- "The cruise control should always disengage when the "turn off" button is pressed."
Could these requirements be formalized and automatically checked? An imaginary case ...

When in 5\textsuperscript{th} gear and RPM drops below x, then the system should always switch from 5\textsuperscript{th} to 4\textsuperscript{th} gear.

\[ \Box (g=5 \land \omega < x) \rightarrow \Diamond_{[0,\tau]} g=4 \]

The engine should never stall while idle.

\[ \Box (\text{idle} \rightarrow \omega > 1100 \text{ RPM}) \]

The electric motor should always rotate in the direction selected by the transmission.

\[ \Box (g \geq 1 \land \text{"other"} \rightarrow \omega_{\text{em}} > 0) \]

The cruise control should always disengage when the “turn off” button is pressed.

\[ \Box (\text{turnoff} \rightarrow \Diamond_{[0,\tau]} \text{cc=off}) \]
Translating the Boolean Requirements Verification Problems into Optimization Problems

Goal: find a descent direction s.t.:
\[ R_\phi(x_0 + \hat{x}_0, u + \hat{u}) < R_\phi(x_0, u) \]

Abbas et al, Probabilistic Temporal Logic Falsification of Cyber-Physical Systems, TECS 2013

Yaghoubi & Fainekos, Falsification of Temporal Logic Requirements Using Gradient Based Local Search in Space and Time, ADHS 2018
Trial in Actual Control Model (Past defect case)

Detect following defect on SiLS model including all engine control
“monitor value—request value>50” continue over 500msec

There are 75 Control point

<table>
<thead>
<tr>
<th>Generated input</th>
<th>Defect condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gas pedal [%]</td>
<td>① Specific logic on</td>
</tr>
<tr>
<td>Brake [%]</td>
<td>② Engine revolution around 4000rpm</td>
</tr>
<tr>
<td>Shift{P,N,D}</td>
<td>③ Satisfy ①,② and specific accelerator operation</td>
</tr>
<tr>
<td>Water temp[℃]</td>
<td></td>
</tr>
<tr>
<td>Air temp[℃]</td>
<td></td>
</tr>
<tr>
<td>Air pressure[kPa]</td>
<td></td>
</tr>
<tr>
<td>Air conditioner SW</td>
<td></td>
</tr>
</tbody>
</table>

Last case also solved as of 2016 with the method presented in EMSOFT 2015

Tried 6 large-scale models, 5 models were falsified.

(Past defect case, intentional defect by logic developer)

S-Taliro could generate the complicated scenario including the defect

---

Generated input:
- Gas pedal [%]
- Brake [%]
- Shift{P,N,D}
- Water temp [℃]
- Air temp [℃]
- Air pressure [kPa]
- Air conditioner SW

Defect condition:
- ① Specific logic on
- ② Engine revolution around 4000rpm
- ③ Satisfy ①,② and specific accelerator operation

Generated signals automatically:
- Gas pedal [％]
- Temperature [℃]
- Engine revolution [rpm]
- Logic Output

Figure Generated signals automatically

Shunsuke Kobuna
Can we use the same framework to test autonomous vehicles?

Adversarial testing: What is the goal of the adversaries?

Can we use the same framework to test autonomous vehicles?

Challenge: we drive optimistically!

Our claim: We need to detect and robustify “boundary” situations, i.e., we need adversaries to exercise the boundary behaviors between safe and unsafe scenarios.
Testing sensor placement for ADAS

Tuncali, Pavlic, Fainekos, Utilizing S-TaLiRo as an Automatic Test Generation Framework for Autonomous Vehicles, IEEE ITSC, 2016
More adversarial testing …

What can we achieve by carefully encoding the requirements?

Tuncali et al, *Simulation-based Adversarial Test Generation for Autonomous Vehicles with Machine Learning Components*, IEEE IV, 2018
System level requirements

Always ( [ego moving] \implies [no objects collide in front])

Always ( [ego moving] \implies [no objects collide in front subject to some distance and velocity constraints\ast])

System level requirements

Vehicle Control
- Sensor: Camera
- Perception System: Object Detection & Classification (SqueezeDet)
- Controller: Decision Making, Low-level Control

Simulator (Webots)
- Scene Rendering: Camera Projections
- Vehicle & Environment Physical and 3D Models: Physics Engine, Simulated Objects

Requirements Testing on Ego Vehicle
Example: DNN-based collision avoidance

Safety Requirement:

Always ( [ego moving] \Rightarrow [no objects collide in front])

Search space:

- Continuous: walking speed of the pedestrians; initial longitudinal position of Ego; longitudinal position of the blue vehicle
- Discrete: vehicle color and models; pedestrian shirt and pants color

Object tracking algorithm: Median Flow (OpenCv Library)

* Erkan’s toy collision avoidance controller!

Tuncali et al, Simulation-based Adversarial Test Generation for Autonomous Vehicles with Machine Learning Components, IEEE IV, 2018
Search-based Testing for Autonomous Vehicles

Nominal Case: No collision

Search-based Testing Result: Collision
**Perception and System level requirements**

**Vehicle Control**
- Sensors: LIDAR, Camera
- Perception System & Sensor Fusion: Clustering etc, Object Detection & Classification (SqueezeDet)
- Controller: Decision Making, Low-level Control

**Simulator (Webots)**
- Scene Rendering: Camera Projections, Ray Tracing
- Vehicle & Environment Physical and 3D Models: Physics Engine, Simulated Objects

**Main benefit of Requirements driven Simulation-based testing:** We know the ground truth!
Scenario Setup

Parameters:
- Ego initial speed
- Ego initial distance to the intersection;
- Agent 1 initial distance to the intersection,
- Agent 1 initial speed,
- Agent 1 speed when approaching the intersection,
- Agent 1 speed inside the intersection,
- Agent 1 initial lateral position
- Agent 1 target lateral position when approaching the intersection,
- Agent 1 target lateral position inside the intersection;
- Agent 2 initial lateral position,
- Agent 2 speed,
- Agent 2 initial distance to the intersection.
Case Study: Sensor Error Leads to Collision

\[ R_{i,s} = \Box \neg \left( \Box_{[0,t_1]} \left( \neg \pi_{i,\text{coll}} \land W(i,s) \land (\neg D(i,s) \lor E(i,s) > \epsilon_{\text{err}}) \right) \land \Diamond_{(t_1,t_2)} \pi_{i,\text{coll}} \right) \]

(not) Collision  (not) Obj. detected  Localization error

Close to falsification  Falsifying Example
Opportunity: Writing formal requirements for perception systems

Dokhanchi, Ben Amor, Deshmukh, & Fainekos, Evaluating Perception Systems for Autonomous Vehicles using Quality Temporal Logic, RV 2018
KITTI Highway Example

“At every frame, for all the objects \((id)\) in the frame, if the object class is CAR with probability more than 0.7, then in the next 5 frames the object \((id)\) should still be detected and classified as CAR with probability more than 0.6”

\[
\phi_2 = \Box \left( x. \forall id @ x, (C(x, id) = Car \land P(x, id) > 0.7) \right)
\rightarrow \Box \left( y. ((x \leq y \land y \leq x + 5) \rightarrow C(y, id) = Car \land P(y, id) > 0.6) \right)
\]
TQTL requirement is Falsified

\[
\phi_2 = \square \left( x. \forall id @ x, (C(x, id) = Car \land P(x, id) > 0.7) \rightarrow \square \left( y. ((x \leq y \land y \leq x + 5) \rightarrow C(y, id) = Car \land P(y, id) > 0.6) \right) \right)
\]

The front car (Red Box) suddenly be overtaken and becomes undetected in a three frames (Yellow).
Highway Alternative Requirement

“At every frame, for all the objects (id) in the frame, if the object class is CAR with probability more than 0.7, then in the next 5 frames the object (id) should still be detected and classified as CAR with probability more than 0.6 or the object (id) is fully occluded.”

\[
\phi_3 = \Box \left( x. \forall id \in x, (C(x, id) = Car \land P(x, id) > 0.7) \rightarrow \Box \left( (x \leq y \land y \leq x + 5) \rightarrow (C(y, id) = Car \land P(y, id) > 0.6 \lor O(y, id) = FULLY) \right) \right)
\]
Summary & Challenges
Tools & Methods Zoo for CPS Analysis* (with or without DNN in the loop)

*List of references in the accompanying survey paper
Current Challenges?

• When a requirement violation (bug) is found, what is the issue?
  - Is it the DNN/ML? The controller? A combination thereof?
  - Can this be fixed automatically?

• When a requirement violation is not found, what guarantees can we provide?

• Exhaustive verification / reachability analysis with DNN perception in the loop: Is it always a well defined problem?

  • Potentially, a practical solution is monitoring instead of verification
Acknowledgements

Current Students
• Sai K. Bashetty – MS
• J. Campbell* – PhD
• M. Hekmatnejad – PhD
• K. Kim – PhD
• C. E. Tuncali – PhD
• S. Yaghoubi – PhD

Alumni (PhD)
• Houssam Abbas* (Oregon)
• Adel Dokhanachi (ASU)
• Bardh Hoxha (SIU)

Alumni MSc & BS (Special shout-out)
• Y. Annapureddy - MS
• R. Kermani - MS
• S. Srinivas - MS
• R. T. Srinivasa – MS
• S-K Su – MS
• H. Yang – MS
• H. Bach – BS
• J. Mendoza – BS

* co-advised

Recent (Close) Collaborators
Academia:
• CU, Boulder: S. Sankaranarayanan
• OSU: U. Ozguner
• RPI: Agung Julius
• U.Del.: H. G. Tanner
• USC: J. V. Deshmukh
Industry:
• Toyota: H. Ito, J. Kapinski, D. Prokhorov, K. Ueda

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.