Temporal Logic Planning for Mobile Robots:
What happens when missions cannot be satisfied?

Georgios Fainekos

Maryland Robotics Center @ UMD

fainekos at asu edu
http://www.public.asu.edu/~gfaineko
Search & Rescue in Urban Environments
Search & Rescue in Urban Environments

(Please) Drop water on structure A and patrol all garage structures at least once every 15 min.
These scenarios are not so far in the future ...

AFRL Summer of Innovation 2017
UxAS - AMASE

Please escort the new patient to Examination Room1 and check if the patients in Wing A need any assistance every half an hour.

iRobot RP-VITA assisting in hospitals?!!
Success story:
Use LTL as mission specification language

Diamond (A ∧ drop_water) ∧ □◊[0,15]garage1 ∧ □◊[0,15]garage2 ∧ □◊[0,15]garage3
Example: Synthesis for quadrotors using feedback linearization and PID control

Keep taking photos and upload current photo before taking another photo. Serve dynamic requests first if any detected.
This is now a well understood problem: LTL planning & synthesis for robotics & control

An incomplete and ever growing list of researchers:

- Alur, UPenn
- Baras, UMD
- Belta, U of Boston
- Dimarogonas, KTH
- Egerstedt, Georgia Tech
- Frazzoli, MIT
- Julius, RPI
- Karaman, MIT
- Kavraki, Vardi, Rice
- Kloetzer, TU of Iasi
- Kress-Gazit, Cornell
- Kyriakopoulos, NTUA
- Liu, Waterloo
- Majumdar, MPI-SWS
- Mazo, TU Delft
- Murray, Caltech
- Ozay, UMich
- Pappas, Upenn
- Prabhakar, Kansas State
- Saha, IIT Kanpur
- Tabuada, UCLA
- Topcu, UT Austin
- Tumova, KTH
- Xu, UMD
- Zamani, TU München

Lahijanian, Kloetzer, Itani, Belta, Andersson. ICRA 2009
Bobadilla, Sanchez, Czarnowski, Gossman, LaValle. RSS 2011
LaViers, Chen, Belta, Egerstedt. IEEE RAM 2011
Today ...

What happens if the high level mission goals cannot be satisfied??
Can we get feedback if the mission cannot be satisfied?

\[ \Diamond (A \land \text{drop\_water}) \land \Box \Diamond_{[0,15]} \text{garage1} \land \Box \Diamond_{[0,15]} \text{garage2} \land \Box \Diamond_{[0,15]} \text{garage3} \]

Ok what can you do?

I'm sorry Dave, I'm afraid I can't do that
How planning / synthesis can fail?

Drop water on structure A and patrol all garage structures at least once every 15 min.

Implicit cost constraints:
You may have to load water from a specific location and your battery capacity might not be sufficient.

Remove the no-fly zone or we can only visit G₁.
What if we have to update the mission goals on-line?

(Please) Drop water on structure A and patrol all garage structures every 15 min.

1. Whenever you detect increased heat (infrared sensor), take pictures.
2. Avoid areas with heavy smoke.
Conflicting inputs?

Structure A is on fire, but B is not!

Structure B is on fire, but A is not!
What about uncertain information? (Probabilities are not available in all contexts)

Go to structure A, but avoid region R since most likely there is smoke.
Mission specification revision

First, no costs or weights

- G. Fainekos, Revising Temporal Logic Specifications for Motion Planning, IEEE ICRA, 2011
LTL/MTL Semantics Intuition (Boolean version)

\( a \) - a now

\( G a \) or \( \square a \) - always a

\( F a \) or \( \Diamond a \) - eventually a

\( a \ U b \) - a until b

\( a \ U_{[1,1.5]} b \) - a until b

Now \hspace{2cm} Time
Example (Road network Scenario)

Periodically visit a data gather location \((q_1, q_2, q_3)\) and a data upload location \((q_4, q_5)\)

\[
GF(q_1 \lor q_2 \lor q_3) \land GF(q_4 \lor q_5)
\]

- Srinivas et al, Graphical Language for LTL Motion and Mission Planning IEEE ROBIO, 2013
- Wei et al, Extended LTLvis Motion Planning Interface, IEEE SMC, 2016
Supervisory Controller / TL planner

\[
\varphi = (\neg q_1) U (q_2 \land (\neg q_1) U q_3)
\]

*Simplified figure since, for instance, “q2 \land q3” cannot happen in this example.
When planning succeeds

“Go to q2 not passing q1 and then until reaching q3, do not visit q1.”

\[ \varphi = (\neg q_1)U(q_2 \land (\neg q_1 \land \neg q_4)Uq_3) \]

Supervisor

For any initial state

Find a path to an accepting state

and

Find a path from an accepting state back to itself

Product Automaton
When planning fails …

“Go to q2 not passing q1 and then until reaching q3, do not visit q1.”

\[ \phi = (\neg q_1)U(q_2 \land (\neg q_1 \land \neg q_4)Uq_3) \]

Cannot find a path to an accepting state

or

Cannot find a path from an accepting state back to itself
Problem 1: Why did the specification failed?

• **Straightforward solution:**
  • Find the set of reachable atomic propositions $R_\Pi$

  • If an atomic proposition in the specification $\phi$ is not in $R_\Pi$, then the planning failed due to an unreachable state in the system

  $$\phi = \text{F ROOM2}$$

  • If all the atomic propositions in the specification $\phi$ are in $R_\Pi$, then the planning failed due to “logical inconsistencies”

  $$\phi = (\neg \text{COR}) \cup \text{ROOM1}$$
Revising the specification

• Ok, $\phi = (\neg q_1) \cup (q_2 \land (\neg q_1 \land \neg q_4)) \cup q_3$ cannot be satisfied.

What specification can be satisfied?

• We must restrict our search space
  • “Irrelevant” solutions are not option, e.g., $\phi = G q_5$.
    • E.g., you ask the robot to bring oranges and the robot responds actually I can only bring apples

• Non-minimal solutions are not desirable, e.g., $\phi = \text{true}$
  • E.g., you ask the robot to bring oranges and the robot responds actually I can only stay here and do nothing
Partial Order on the LTL formulas

• Let $\varphi_1$ and $\varphi_2$ be 2 LTL formulas, then

$\varphi_1 < \varphi_2$ if $\varphi_1 \Rightarrow \varphi_2$

• Remark: In order to have a lattice, we need to consider the congruence under the equivalence relation of LTL formulas
  • e.g. $F(p_1 \vee p_2) \equiv Fp_1 \vee Fp_2$

• Hence, the conjunction and disjunction become the meet and join operations over the lattice
Partial Order on the LTL formulas

Let $\varphi_1$ and $\varphi_2$ be 2 LTL formulas, then
$\varphi_1 < \varphi_2$ if $\varphi_1 \Rightarrow \varphi_2$

Notation
$\varphi_1 < \varphi_2$

false < p < Fp < true

$F(p_1 \lor p_2)$

true

Fp

p

false

G(p_1 \land p_2)

p_1 \lor p_2

p_1 \land p_2
Possible space of solutions ...

\[ RF(\varphi, T) = \{ \varphi' \in \text{LTL} \mid T \models \varphi' \text{ and } \varphi < \varphi' \} \cap \text{rel}(\varphi) \]

where \( \text{rel}(\varphi) \) removes some “irrelevant” solutions.
E.g. \( \text{rel}(\varphi) \) maximally relaxes each AP

\[
\text{RF}(\varphi,T) = \{ \varphi' \in \text{LTL} \mid T \models \varphi' \text{ and } \varphi < \varphi' \} \cap \text{rel}(\varphi)
\]

where \( \text{rel}(\varphi) \) removes some “irrelevant” solutions.
Possible space of solutions ...

\[ F(\varphi, T) = \{ \varphi' \in \text{LTL} \mid T \models \varphi' \text{ and } \varphi < \varphi' \} \]

- However, searching for a minimal solution in \( F(\varphi, T) \) is not enough ...

\[ \varphi = \pi_0 \land F(\pi_2 \land F\pi_1) \]

\[ \varphi' = (\pi_0 \land F(\pi_2 \land F\pi_1)) \lor G(\pi_0 \land \neg \pi_1 \land \neg \pi_2) \]
Modified possible space of solutions ...

$$RF(\varphi, T) = \{\varphi' \in LTL \mid T \vDash \varphi' \text{ and } \varphi < \varphi'\} \cap \text{rel}(\varphi)$$

where rel(\varphi) removes some “irrelevant” solutions

• Example: rel(\varphi) is the set of all formulas where each atomic proposition \(\pi\) in \(\varphi\) is replaced by a formula in its upper bound
  • e.g.

  \[
  \begin{align*}
  \text{true} \\
  \mid \\
  F\pi \\
  \mid \\
  \pi
  \end{align*}
  \]
Automatic Revision and Minimality due to unreachable atomic propositions

**Theorem:** Let LTL formula $\varphi$ be unsatisfiable on system $T$ and $U$ be the set of unreachable atomic propositions in $T$. Set $\varphi' = \text{rem}_U(\varphi)$. Then,

1. we have $\varphi < \varphi'$.
2. $\varphi'$ is minimal in $RF(\varphi, T)$

where:

1. $\text{rem}_U(\varphi)$ replaces each atomic proposition in $\varphi$ that is also in $U$ with true
2. $\varphi'$ is minimal in $RF(\varphi, T)$ if for any other $\psi$ in $RF(\varphi, T)$ we have $\psi < \varphi'$, then $\psi \equiv \varphi'$.

$\varphi = \pi_0 \land F(\pi_2 \land F\pi_1)$

$\varphi' = \pi_0 \land F\pi_2$
Automatic Revision and Minimality due to “logical inconsistencies”

• Can we find the minimal revision in RF(φ,T)?
  • Obviously, the problem is decidable
  • Assume that we are going to maximally relax (i.e., set to true) each atomic proposition in φ
  • We need to consider $2^{|\text{AP}(φ)|}$ combinations of maximal relaxations
  • For each combination, we need to run the LTL planning algorithm which is of complexity $|T|2^{O(|φ|)}$
  • Worst case complexity $|T|2^{O(|φ|)2^{|\text{AP}(φ)|}} = |T|2^{O(|φ|)+|\text{AP}(φ)|}$
Translating the minimal specification revision problem into a graph problem ...

Label each edge in the product automaton with the set of atomic propositions from the spec that must be removed for edge to become enabled.
Problem: Find the path on the graph with the smallest number of AP to be removed
Main result: NP-Completeness

**Minimal Accepting Path (MAP) Problem:**
Given an instance of the minimal accepting path problem \( (G, Y, L, v_0, F) \) and a bound \( W \), the decision of whether there exists a truth assignment \( Z \subseteq Y \) such that \( |Z| \leq W \) is NP-Complete.

Proof by reduction from 3-CNF-SAT.
Intuition on why the problem is hard

The optimal subpath property does not hold.
Approximation Algorithm for MRP

- Based on Dijkstra’s shortest path algorithm
- Instead of finding minimum weight, it tracks the number of atomic propositions to be removed from the edges.
Experimental Results:
Heuristic Algorithm based on Dijkstra’s algorithm

<table>
<thead>
<tr>
<th>Nodes</th>
<th>ASP</th>
<th>AAMRP</th>
<th>RATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>min</td>
<td>avg</td>
<td>max</td>
</tr>
<tr>
<td>9</td>
<td>0.003</td>
<td>0.0071</td>
<td>0.012</td>
</tr>
<tr>
<td>100</td>
<td>0.099</td>
<td>0.1954</td>
<td>1.405</td>
</tr>
<tr>
<td>196</td>
<td>0.335</td>
<td>1.25058</td>
<td>6.003</td>
</tr>
<tr>
<td>324</td>
<td>0.869</td>
<td>5.3316</td>
<td>14.731</td>
</tr>
<tr>
<td>400</td>
<td>1.267</td>
<td>12.87</td>
<td>35.58</td>
</tr>
<tr>
<td>529</td>
<td>3.086</td>
<td>34.1642</td>
<td>103.838</td>
</tr>
</tbody>
</table>

**TABLE I**

**Numerical Experiments:** Number of nodes versus the results of ASP solver and AAMRP. Under the ASP and AAMRP columns the numbers indicate computation times in sec. RATIO indicates the experimentally observed approximation ratio to the optimal solution.

<table>
<thead>
<tr>
<th>Nodes</th>
<th>ASP</th>
<th>AAMRP</th>
<th>RATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>min</td>
<td>avg</td>
<td>max</td>
</tr>
<tr>
<td>9</td>
<td>0.005</td>
<td>0.0097</td>
<td>0.039</td>
</tr>
<tr>
<td>100</td>
<td>0.378</td>
<td>18.4679</td>
<td>3502.343</td>
</tr>
<tr>
<td>196</td>
<td>3.336</td>
<td>31.955</td>
<td>685.819</td>
</tr>
<tr>
<td>306</td>
<td>9.801</td>
<td>75.524</td>
<td>2795.337</td>
</tr>
<tr>
<td>400</td>
<td>21.744</td>
<td>124.7486</td>
<td>164.5349</td>
</tr>
<tr>
<td>506</td>
<td>58.67</td>
<td>241.167</td>
<td>1034.98</td>
</tr>
</tbody>
</table>

**TABLE II**

**Numerical Experiments:** Number of nodes versus the results of ASP solver and AAMRP. Under the ASP and AAMRP columns the numbers indicate computation times in sec. RATIO indicates the experimentally observed approximation ratio to the optimal solution.
Experimental Results:
Heuristic Algorithm based on Dijkstra’s algorithm

Preliminary results on scalability using a prototype Python implementation:

<table>
<thead>
<tr>
<th>Nodes</th>
<th>ASP min</th>
<th>ASP avg</th>
<th>ASP max</th>
<th>AAMRP min</th>
<th>AAMRP avg</th>
<th>AAMRP max</th>
<th>RATIO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1024</td>
<td>24.438</td>
<td>168.2133</td>
<td>237.758</td>
<td>0.125</td>
<td>0.23</td>
<td>0.325</td>
<td>9/10</td>
</tr>
<tr>
<td>10000</td>
<td>0/10</td>
<td>15.723</td>
<td>76.164</td>
<td>128.471</td>
<td>9/10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20164</td>
<td>0/10</td>
<td>50.325</td>
<td>570.737</td>
<td>1009.675</td>
<td>8/10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50176</td>
<td>0/10</td>
<td>425.362</td>
<td>1993.449</td>
<td>4013.717</td>
<td>3/10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>60025</td>
<td>0/10</td>
<td>6734.133</td>
<td>6917.094</td>
<td>7100.055</td>
<td>2/10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TABLE III**
Numerical Experiments: Number of nodes versus the results of ASP solver and AAMRP. Under the ASP and AAMRP columns the numbers indicate computation times in sec. RATIO indicates the experimentally observed approximation ratio to the optimal solution.
Download our modified LTL2BA toolbox!

- Git Repo: https://www.assembla.com/spaces/ltl2ba_cpslab/

\[ \phi = (\neg q_1) \cup (q_2 \land (\neg q_1) \cup q_3) \]
Adding weights and preferences
LTL Planning on Weighted Transition Graphs*

Find a path on the graph that
minimizes $U_i P_i$
subject to $\Sigma_i w_i \leq W$

Can be solved using MILP formulations.

LTL Planning on Preference Transition Graphs*

Find a path on the graph that

1) minimizes $\Sigma_i \text{Pref}(P_i)$ where $\text{Pref}: P \rightarrow \mathbb{R}$, or
2) minimizes $\max_i \text{Pref}(P_i)$ where $\text{Pref}: P \rightarrow \mathbb{R}$

Min-sum problem is NP while the min-max problem is in P.

Adding Requirements On-line: Checking consistency
Example

Car-like model from the Robotics Toolbox (Corke 2011):

\[ \begin{align*}
\dot{x} &= v \cos \theta \\
\dot{y} &= v \sin \theta \\
\dot{\theta} &= \frac{v}{L} \tan \gamma
\end{align*} \]

\(v\): forward speed  \\
\(L\): length of vehicle  \\
\(\gamma\): steering angle
PLAN$_{RM}$ Framework

- **Inputs**: Local MTL Specification, Global LTL Specification, Environment Map
- **Offline/Rarely Called Global Planner**
  - Global Planner (1)
  - Global MTL Planner
- **Online/Runtime**
  - MTL Monitoring (3)
  - Path Evaluation Metric (4)
  - Local RRT Planner (2)
  - Robotic System
  - Vehicle Inputs
  - Local Sensing/Update Map
  - State

Arrows indicate the flow of information and processes in the framework.
Checking consistency of the goals

Case 1: Both local and global requirements are in LTL

1. Keep track of the satisfied goals, e.g., Fainekos et al CDC 2005
2. Analyze and detect the conflicts, e.g., Fainekos ICRA 2011

Case 2: Global requirements in LTL and local requirements in MTL

Open problem: A practical way to detect conflicts and inconsistencies.

Case 3: Detect issues among local requirements in bounded MTL

Recent results: Dokhanchi et al. Metric Interval Temporal Logic Specification Elicitation and Debugging, MEMOCODE 2015
Problem

Given an MITL formula $\phi$, find whether $\phi$ has any of the following logical issues:

- **Validity**: the specification is unsatisfiable or a tautology.
- **Redundancy**: the formula has redundant conjuncts.
  
  $$\Phi = \land_{j=1}^{k} \varphi_j; \text{ if } \exists \varphi_i \text{ s.t. } \Phi \setminus \varphi_i \models \varphi_i, \text{ then } \varphi_i \text{ is redundant}$$
- **Vacuity**: some subformulas do not contribute to the satisfiability of the formula.
Implementation and Experiments

- We used MITL satisﬁability solver for this method:
  \[ \varphi \equiv \psi \iff (\varphi \implies \psi) \equiv \top \iff (\neg \varphi \lor \psi) \equiv \top \iff (\varphi \land \neg \psi) \equiv \bot \]
- We detected all the issues with MITL SAT solver

Utilize multi-valued temporal logic and multi-valued transition systems

How do we capture such information in a mathematically succinct way in order to plan or synthesize control policies?

- G. Fainekos and H. G. Tanner, Temporal Logic Control under Incomplete or Conflicting Information, ACC 2017
- M. Hekmatnejad and G. Fainekos, Optimal LTL Planning for Multi-Valued Logics, ACC 2018
LTL Multi-Valued Semantic Intuition

- **a** – a now: $\frac{1}{2}$
- **G a** - always a: 0
- **F a** – eventually a: 1

Diagram:

- A transition from $\frac{1}{2}$ to 0 to 0 to 1 to $\frac{1}{2}$ to 0.
- Nodes label the times:
  - $\frac{1}{2}$: true
  - 0: maybe
  - 1: false
Other truth degree lattices

\[ B_2 = (\{0,1\}, \leq) \]

\[ B_{2,2} = B_2 \times B_2 \]

\[ B_3 = (\{0, \frac{1}{2}, 1\}, \leq) \]

\[ B_{3,3} = B_3 \times B_3 \]
MV-Transition Systems

• An mv Transition Systems is a tuple $M = (S, S_0, \mathcal{L}, \rightarrow, AP, \mathcal{O})$
  - $S$ is a (finite) set of states
  - $S_0$ is a set of initial states ($S_0 \subseteq S$)
  - $\mathcal{L}$ is a lattice or an algebra
  - $\rightarrow \subseteq S \times S$ is a transition relation
  - $AP$ is a (finite) set of atomic propositions
  - $\mathcal{O} : S \times AP \rightarrow L$ is a total labelling function that maps a pair of a state $s$ and an atomic proposition $a$ to an element of the lattice $L$

\[
\begin{align*}
a &= 1 \\
b &= 0
\end{align*}
\[
\begin{align*}
a &= 0 \\
b &= 1/2
\end{align*}
\]

A trace $s : AP \times \mathbb{N} \rightarrow L$
Problem Statement I:
Compute a plan that satisfies the specification with degree at least $\epsilon$
Planning/synthesis through reduction to Boolean methods

Proposition:
Given $\varepsilon \in \mathbb{L}$, an LTL formula $\varphi$ in NNF and $M = (S, S_0, \mathcal{L}, \rightarrow, AP, O)$, then

$$\bar{s} \models_{M, \varepsilon} \varphi \Rightarrow \llbracket \varphi \rrbracket(s, 0) \geq \varepsilon$$
Planning/synthesis through reduction to Boolean methods

What about the other direction? When it is the case

\[ \llbracket \varphi \rrbracket(s, 0) \geq \varepsilon \Rightarrow \tilde{s} \models_{M_{\varepsilon}} \varphi? \]

**Lemma:** If \( x \in L \) is join irreducible, then for \( y, z \in L \), \( x \leq \sup(y, z) \) implies \( x \leq y \) or \( x \leq z \).

\( x \) is join irreducible if \( x \) is not the bottom and \( x = \sup(y, z) \) implies \( x = y \) or \( x = z \) for all \( y, z \) in \( L \)
Summary

Corollary:
The lower bound truth value temporal logic planning problem can be solved at the same cost as the classical Boolean valued temporal logic planning problem when the truth value lattice is totally ordered or the lower bound truth value requested is a join irreducible element.
Problem Statement II:
Compute a plan with maximal degree of satisfaction

Robot(s) model
\[
\dot{x} = f(x, p, u) \\
y = g(x, p, u)
\]
\( X_0 \subseteq X \)

Environment

Abstraction

Linear Temporal Logic (LTL)

Abstraction

TL Planning/Synthesis Algorithm

Planning succeeds

Built the best controller
Computing optimal satisfaction plans

Brute force search:
- Start from the top element and run the classical planning algorithm for each lattice value above $x$.
- Complexity:
  - Running $|L|$ times automata theoretic planning: $O(|L| |M| 2^{O(\varphi)})$
  - Cost of creating abstractions: $O(|S| |AP|)$
Computing optimal satisfaction plans

Direct approach using graph search:

- **Assumption:** Linear order for lattice values

- An mv-LTL formula can be translated* into an mv-automaton**

\[ [L(A)](s) = x \text{ iff } [\varphi](s) = x \]

- Here \[ [L(A)](s) = \sup_{p \in AR(A)} \inf_{i \geq 0} s_i(\Delta(p, p_{i+1})) \]

- Languages accepted by mv-automata are closed under intersection**

* The standard translation algorithm applies by ignoring the equivalences \( a \land \neg a = \bot \) and \( a \lor \neg a = \top \)

** Chechik, et al “Model-checking infinite state-space systems with fine-grained abstractions using SPIN,” SPIN Workshop, 2001
Computing optimal satisfaction plans

\[
[L(A)](s) = \sup_{p \in AR(A)} \inf_{i \geq 0} s_i(\Delta(p_i, p_{i+1}))
\]
Summary

**Corollary:**

The optimal truth value temporal logic planning problem can be solved in polynomial time when the truth value lattice is totally ordered.
Road Network Accessibility

Possible Move Directions
Restricted Accessibility Directions
Not Accessible

M. Hekmatnejad and G. Fainekos, Optimal LTL Planning for Multi-Valued Logics, ACC 2018
The need for formal requirements in testing

\( \square( \text{idle} \rightarrow \omega > 1100 \text{ RPM}) \)

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

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

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

Google

BMW

Toyota

Uber

Mercedes

Ford

Volvo
Software errors in autonomous vehicles: How serious is this problem?

Source: J.D. Power SafetyIQ and NHTSA’s safe.cars.gov
What are these software recalls?
Sampling of recalls from the 2011-12 period …

- "A software error may prevent the engine from moving to lower gear when coasting, such as shifting from 5th to 4th gear when coasting can 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 and in many cars also disables power steering. Braking or pressing the cancel button will not work.

- "..."
"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, if certain driving conditions fall outside of the 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 in many cars also disables power steering. Braking or pressing the cancel button will not work.

...
Could these requirements be formalized and automatically checked? An imaginary case …

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

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

The engine should never stall while idle.

\( \Box \left( \text{idle} \rightarrow \omega > 1100 \text{ RPM} \right) \)

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

\( \Box \left( g \geq 1 \land \text{“other”} \rightarrow \omega_{em} > 0 \right) \)

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

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

**Goal:** find a descent direction s.t.:

\[ R_\phi(x_0 + \hat{x}_0, u + \hat{u}) < R_\phi(x_0, u) \]

\[
\Delta u(t) = - \left( \frac{\partial \hat{X}_{11}(t)}{\partial u} \right)^T \frac{\partial G(\hat{X}_{11}(t))}{\partial \hat{X}}
\]

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

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

Figure Generated signals automatically

StaliRo https://sites.google.com/a/asu.edu/staliro/68
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 to find the boundary behaviors between safe and unsafe scenarios.
Testing sensor placement for ADAS
What about ML based collision avoidance?

Safety Requirement:

Always ( [ego moving] ⇒ [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)
Search-based Testing for Autonomous Vehicles

Nominal Case: No collision

Search-based Testing Result: Collision
To conclude …
Vision: a complete theory for MBD for CPS

Transparent from the user perspective:
1. Automated synthesis
2. Testing and verification support with guarantees

Awards:
1017074, 1116136, 1319560, 1350420, 1446730

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.
S-TaLiRo support in the V-process

1. Testing formal specifications and specification mining [TECS 2013, STTT 2018, …]
2. Conformance testing: models, HIL/PIL or tuned/calibrated model [MEMOCODE 2014]
3. Testing formal specifications on the HIL/PIL calibrated system [TECS 2013, …]
4. Runtime monitoring of formal requirements [RV 2014]
Acknowledgements

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

Alumni (PhD)
• Houssam Abbas (UPenn)
• Adel Dokhanchi (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

Recent (Close) Collaborators

Academia:
• CU, Boulder: S. Sankaranarayanan
• OSU: U. Ozguner
• Princeton: A. Gupta
• RPI: Agung Julius
• U.Del.: H. G. Tanner
• USC: J. V. Deshmukh

ASU: H. Ben Amor, Y. Kobayashi, Y-H Lee, H. Mittelmann, T. Pavlic, Y. Zhang

Industry:
• Toyota: H. Ito, J. Kapinski, K. Ueda
• Google: F. Ivancic

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.

Special Thanks: S. Vrudhula (ASU)