Benchmarks for Verification of Autonomous Vehicles

Matthew O'Kelly[1], Houssam Abbas[1], Aditya Pinapala[1], and Rahul Mangharam[1]

University of Pennsylvania, Philadelphia, PA, U.S.A.
mokelly@seas.upenn.edu, habbas@seas.upenn.edu, pinapala@seas.upenn.edu, and rahulm@seas.upenn.edu

Abstract

To do...

1 Introduction

1.1 State of the Art

- Verification Methods and Tools
  - Industry Perspective: Testing and Simulation
  - Control Perspective: Lyapunov Functions
  - Software Perspective: Model Checking
  - Logic Perspective: Theorem Proving

CPS perspective is an integrated approach that captures the manifestation of errors in the physical world.

2 Models

2.1 Vehicle

- Vehicle Dynamics: Bicycle Model
- Planning

2.2 Pure Pursuit

The simplest algorithm for path tracking and trajectory generation... From a type perspective what is the difference between arc and spline? Geometrically, the arc satisfies **convexity**.

- Given a current position at the vehicles rear differential and a goal...
- The algorithm computes a constant curvature arc between the current position and the goal.
- The vehicle may then actuate its steering system such that the curvature of the arc is tracked.
- Downside is discontinuity between curvatures when the algorithm is iterated
- Is also reliant on waypoints, does not generate alternate maneuvers unless explicitly told to switch goals.

Algorithm:

- Update vehicle state

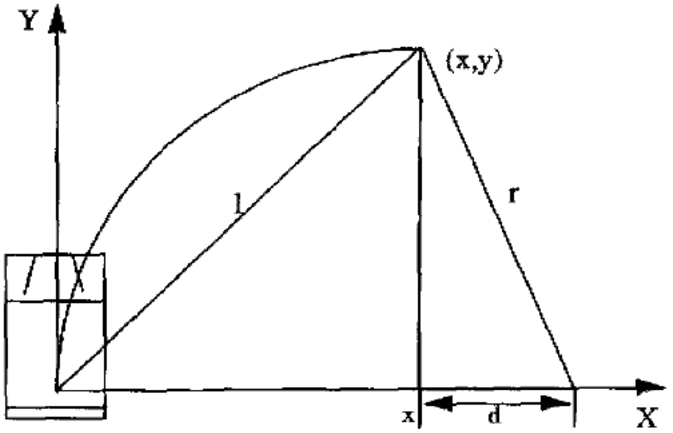

Figure 1. Geometry of the Algorithm.

Figure 1: Geometry of Pure Pursuit Algorithm

- Find nearest path point
- Find the goal point
- Transform goal to vehicle coordinates
- Calculate desired curvature
- Set steering to desired curvature
- Update position

$$x^2 + y^2 = l^2 \tag{1}$$

$$x + d = r \tag{2}$$

Now for the curvature such that:

$$r = \frac{l^2}{2x} \tag{3}$$

$$\gamma = \frac{2x}{l^2} \tag{4}$$

2.3 Behavioral Controller

In [?] the authors develop a behavioral controller which mimics expert driving instructor behavior at blind corners. Specifically such drivers anticipate unseen obstacles and dangers. Such drivers should take different actions depending on their prediction of other potential traffic participants, including both pedestrians and vehicles. We allow that other traffic participants may not be within the sight of the test subject. As such the driver model is defined by three rules each of which produce an update to the vehicles acceleration:

- If a potential (even unseen) traffic participant is likely to collide, the driver will brake proactively,

- else if the current speed is below a desired speed, accelerate,
- otherwise, keep the current speed.

Assuming that the ego vehicle and a potential traffic participant are point masses and will cross perpendicularly, we will model the above rules by acceleration of the ego vehicle:

$$y_{ot} + v_{ot}t_c < \frac{w}{2} \rightarrow a_{et} = \frac{v_{et}^2 - v_{min}^2}{2(x_{et} - x_{min})} \tag{5}$$

$$v_{et} < v_d \rightarrow a_{et} = \frac{v_d^2 - v_{et}^2}{2(x_d - x_{et})} \tag{6}$$

Otherwise...

$$a_{et} = 0 \tag{7}$$

Here, (x_{et}, y_{et}, v_{et}),(x_{ot}, y_{ot}, v_{ot}), (x_d, v_d) are the position and speed of ego vehicle at time t, the current position and speed of a potential traffic participant at time t, and the desired final state of the ego vehicle, respectively. (x_{min}, v_{min}) is the state (position and speed) after finishing deceleration. By introducing a minimum admissible displacement ϵ , x_{min} can be expressed as

$$x_{min} - x_0 = \epsilon \tag{8}$$

Where t_c is a predicted time to collision derived as a function of the speed of the ego-vehicle:

$$t_c = max(\frac{x_{ot} - x_{et}}{v_{et}}, 0) \tag{9}$$

Thus, the condition $y_{ot} + v_{ot}t_c < \frac{w}{2}$ means that the trajectory of the potential traffic participant will reach and stay within a set in which collision is possible w, at the time to collision. We assume that the potential traffic participants begin their paths behind a wall in the occluded region on the left of the intersection.

This changes... because we will add differential inclusions

Each traffic participant may move at a constant speed, from left to right in the scene. This is a reasonable assumption because the side road restricts vehicles to drive one way. Given the wall position (x_w, y_w), the position (x_{ot}, y_{ot}) are expressed as

$$x_{ot} = \gamma d + x_w \tag{10}$$

$$y_{ot} = \gamma d + (x_w - x_{et})tan(\rho_t) + y_{et} \tag{11}$$

Here, d is the width of the side road and ρ_t is the current field of view from the ego vehicle driver computed as:

$$tan(\rho_t) = \frac{y_w - y_{et}}{x_w - x_{et}} \tag{12}$$

Note that is a parameter defining the x position of a potential traffic participant that takes a value from 0 to 1. From equations (1) to (9), one can compute a desired speed profile and trajectory with which the driver can avoid collision to a potential traffic participant coming out from the position (x_{ot}, y_{ot}) at speed v_o.

Traffic Control

- Stop Sign
- Speed Limit

- Yield
- Traffic Light

Pedestrians

- Dynamics
- Grid based abstraction
- Non determinism

3 Benchmarks

3.1 A Simple Lane Change

- We first present this scenario in [**?**]
- Other authors [**?**] have considered with static obstacles
- Counterexamples are somewhat obvious
- Inductive invariance argument allows verification of infinite time properties
- Most examples involving forward safety are monotonic ie selecting a lower operating speed implies an decrease in stopping time and improves the forward safety of the vehicle.
- In such problems there are no "holes" in the interval.
- Resulting proofs seem to indicate that testing would be sufficient to find counterexamples because unsafe results occur at extreme points in the state space only.
  - Verification still provides a guarantee of safety that was not previously available.
  - Verification is still useful for formally verifying specifications and interaction between supplier systems and vehicle dynamics.
  - Helps to search for viable combinations of specifications
- We extend this scenario with differential inclusions.
- Problem: we are forced into the tree representation because we cannot bring the trajectory generator in the dReach framework, requires external libraries and iteration. Could be remedied via creation of a lookup table or Neural Network. Alternative is to change the trajectory generation strategy such as Pure Pursuit.

3.2 Algorithm

How we establish safety of a decision controller. Proofs and lemmas:

- Proof of compositional nature of safety of sequence of trajectories
- Assume gurantee reasoning for parallelization of verification procedure
- Inductive invariance for search termination at a specified depth.

Theorem 1. *Proof of compositional nature of safety of sequence of trajectories*

Theorem 2. *Contract for parallel composition of verification instances*

Theorem 3. *Inductive invariance for search termination at a specified depth.*

Algorithm 1 Check Trajectory Sequence

```
checkSequence (searchDepth)
bool $safetyVar := TRUE$
int $depth := 0$
while $depth \leq searchDepth$ do
    float* $trajectoryParam = getNextTrajectory(depth)$
    $safetyVar := checkTrajectory(trajectoryParam)$

end while
return $safetyVar$
```

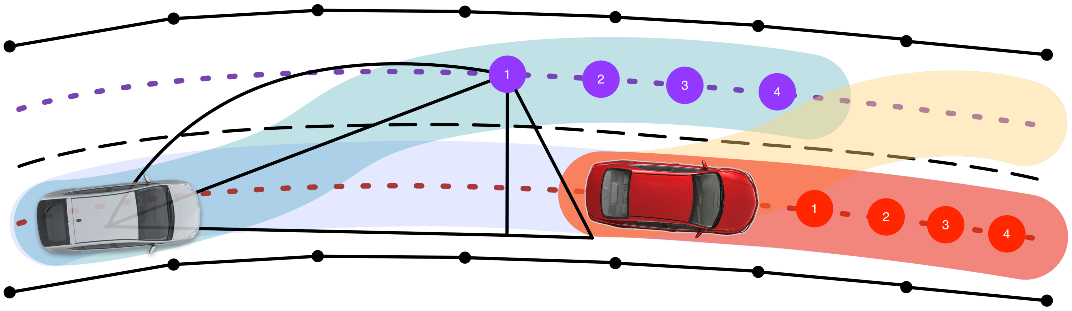

Figure 2: A graphical representation of the lane change scenario on a curved road

3.3 Variations in Road Geometry

- In order to pursue more complex scenarios with more expressive agents we propose a variation in the trajectory generation strategy which admits a closed form
- Implement pure pursuit trajectory generation strategy...
- Open question if we can guarantee infinite time properties in general case, or on road by road basis via inductive invariant.
- A second question of interest is the scalability of results involving multiple agents and curved roads.
- Pure Pursuit can be represented directly in the Hybrid Program and admits a closed form solution for curvature.
- Using a Hybrid program we now show how the number of paths through the state space grows exponentially with the addition of other agents and events as well as increases in search depth.

We present a scenario as shown in Figure **??**

- The previous example is the simplest possible instantiation of a lane change scenario
- Here we present an increasingly nuanced view of the the vehicle environment system
  - Singleton inital sets and deterministic transitions
  - Initial sets defined by intervals

]

- Non-determinism as applied to discrete decisions by the environment
- Automaton complexity in terms of states as number of agents grows
- Automaton complexity in terms of states as number of reference trajectories grows
- Search complexity in terms of path length as number of agents grows
- Search complexity in terms of path length as number of reference trajectories grows
- Non-deterministic scheduling of trajectory update
- Environment agents continuous dynamics are represented as differential inclusions (another form of non-determinism)
- Ego vehicle agents continuous dynamics are disturbed via errors represented as differntial inclusions
- Ego vehicle and environmental agents subjected to discrete events representing sensor or actuator failures.

3.4 Proactive Expert Driver Assistance

Copy Paste from Nagoya stuff I wrote over summer, must be changed later...
In many driving scenarios, the ego-vehicle must anticipate both visible and invisible dangers in the surrounding environment in order to plan safer paths. In residential areas, pedestrians and other vehicles are often completely occluded. In order to reduce the number of severe traffic accidents, proactive driving and pro-active driver models for unseen dangers are important next steps.

Proactive driver models are differentiated from emergency braking systems such as pre-crash safety which react to (1) dangers already in sight, (2) longer-range phenomena still measurable by sensors, (3) or detailed knowledge of road infrastructure. Instead proactive driver models and ADAS must deal with a predicted level of danger in current traffic conditions as an input to the model.

For example, while passing through intersections with a blind corner, the proactive driver model would decelerate in relation to a prediction of the state of other traffic participants, even those which have not yet been sensed. We note, however, that the state space of the other traffic participants contains an infinite number of possible configurations and a large (but finite) number of scenarios, complicating proactive driver model construction. Although attempts have been made to develop controllers, which anticipate hazards in a blind area, the computational principals governing the design and safety of such a system are not yet clear.

In previous works as first step towards creating data-driven proactive ADAS is to understand the behavior of a variety of human drivers. Thus, we collected and analyzed driving data on a public road where drivers need to anticipate unseen dangers. Tests included driving in a residential area, performed by both expert and elderly subjects and accurately analyzed by means of a high-precision data acquisition system.

4 Conclusions

- Types of Uncertainty:
    - Scenario Configuration
    - Initial Estimation Errors
    - Noise in dynamics
    - Sensor estimation errors

  - Sensing thresholds

- Scenario 1: Lane change on straight road with two agents
- Scenario 2: Curved road with hybrid program representation
  - No longer have clear invariant cut
- Scenario 3: Obstructed T-Junction
  - Map provides knowledge of obstruction and automatically adjusts behavior
  - Variant: Traffic light
  - Variant: Pedestrian

References