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

1.2 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
- 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.
- 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.

1.3 Variations in Road Geometry

- Implement pure pursuit trajectory generation strategy...
- 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
- Open question if we can guarantee infinite time properties in general case, or on road by road basis via inductive invariant.

1.4 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.

1.5 Contributions

- Types of Uncertainty:
  - Scenario Configuration
  - Initial Estimation Errors
  - Noise in dynamics
  - Sensor estimation errors
  - Sensing thresholds
- Scenario 1: Obstacle avoidance on a sharp curve
  - No longer have clear invariant cut
- Scenario 2: T-Junction

  - Variant: Traffic light
  - Variant: Pedestrian
- Scenario 3: Obstructed T-Junction
  - Map provides knowledge of obstruction and automatically adjusts behavior

2 Models

New ideas:

- Perception
- Computation and Scheduling
- Traffic Participants

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

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

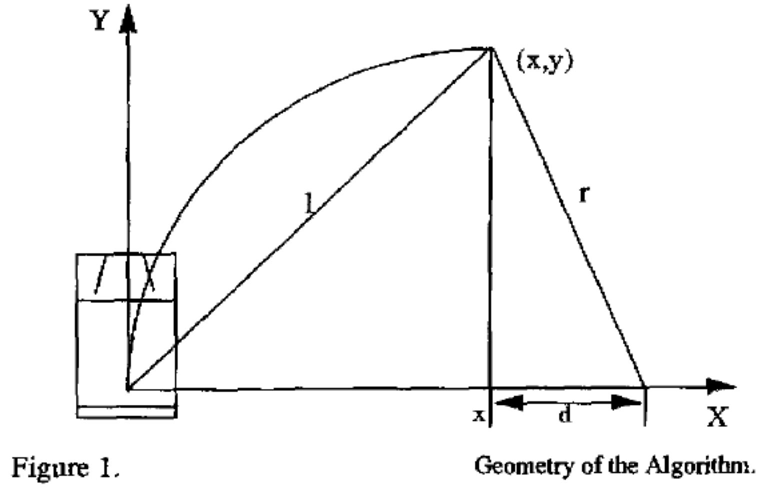

Figure 1: Geometry of Pure Pursuit Algorithm

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 Scenarios

4 Verification Engines

5 Results

6 Conclusions

References

[1] David Carlisle. graphicx: Enhanced support for graphics. http://www.ctan.org/tex-archive/help/Catalogue/entries/graphicx.html, last viewed April 2010, 1995–1999.

A Formatting Information

1. The default paper size is US letter. It can be explicitly set to A4 (`a4paper`) or letter (`letterpaper`) paper in the document class entry, e.g.:
   `\documentclass[a4paper]{easychair}`
2. The print area for both letter and A4 paper sizes is 145x224 mm. This size has been selected to allow for inexpensive printing using our current print-on-demand publisher.
3. The base font is Computer Modern. The base font size is 10pt. If you use any other font size, there is no guarantee that the produced document will look nice or fit into our standard page size.
4. The references list is condensed. The default bibliography styles, such as `plain`, `abbrv`, and `alpha`, are suggested.
5. PNG, JPG, and PDF images are supported, i.e., those that are supported by the standard `graphicx` package [1], and render nicely in online versions of PDF documents. This document shows some examples of JPG and PDF images, for example in Figure **??**. If the papers are designed for publishing in print, the images should be at least 300dpi in resolution.

A.1 Tables

Many page overflows happen because of large tables. In many case these overflows can be easily removed by slightly reducing padding added by LaTeX to every column. It is controlled by the LaTeX command `\tabcolsep` whose value by default is 6pt. Even small changes in the value of this command may give drastic reductions in the width of tables. This is illustrated in Figure 2 on page 7. Note though that there is no free lunch: smaller values for this command may result in lower redability.

A.2 Images

Images included using `\includegraphics` are easy to resize since one can specify the size of the result explicitly. For example, Figure **??** shows three copies of the same image having different sizes obtained using the following commands:

ATP System	LTB	Avg	Prfs	SOTA	μ	CYC	MZR	SMO
Vampire-LTB 11.0	69	24.5	69	0.37	28.1	23	22	24
iProver-SInE 0.7	67	76.5	0	0.36	8.8	28	14	25

ATP System	LTB	Avg	Prfs	SOTA	μ	CYC	MZR	SMO
Vampire-LTB 11.0	69	24.5	69	0.37	28.1	23	22	24
iProver-SInE 0.7	67	76.5	0	0.36	8.8	28	14	25

ATP System	LTB	Avg	Prfs	SOTA	μ	CYC	MZR	SMO
Vampire-LTB 11.0	69	24.5	69	0.37	28.1	23	22	24
iProver-SInE 0.7	67	76.5	0	0.36	8.8	28	14	25

ATP System	LTB	Avg	Prfs	SOTA	μ	CYC	MZR	SMO
Vampire-LTB 11.0	69	24.5	69	0.37	28.1	23	22	24
iProver-SInE 0.7	67	76.5	0	0.36	8.8	28	14	25

Figure 2: Original table and tables with `tabcolsep` set to 5pt, 3pt, and 1pt

```
\includegraphics[width=0.5\textwidth]{throneEC.jpg}
\includegraphics[width=0.3\textwidth]{throneEC.jpg}
\includegraphics[width=0.15\textwidth]{throneEC.jpg}
```

A.3 A Universal Recipe

LaTeX has a very powerful weapon for reducing the size of almost anythings. More precisely, it can reduce anything producing what LaTeX considers a box. This weapon is called `\scalebox`. Consider an example (check the source of this file to see how it was produced).

year	users
2007	47,753
2008	114,494
2009	207,506
2010	371,054

The number of users of EasyChair and one of its logos, scaled to the number of users in 2010

This is what happens when we put (almost) the same LaTeX code in `\scalebox{0.55923}{...}` to scale it down to the number of users in 2009:

year	users
2007	47,753
2008	114,494
2009	207,506
2010	371,054

The number of users of EasyChair and one of its logos, scaled to the number of users in 2009

We can scale it down even further to the 2008 figure using `\scalebox{0.30856}{...}`:

year	users
2007	47,753
2008	114,494
2009	207,506
2010	371,054

The number of users of EasyChair and one of its logos, scaled to the number of users in 2008

or further down to 2007:

year	users
2007	47,753
2008	114,494
2009	207,506
2010	371,054

This size reduction technique is very efficient: using the right scale you may post your whole article on Twitter in a single tweet. However, it may also may parts of your text virtually unreadable with an unfortunate side effect of annoying reviewers.