2014 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
AUTONOMOUS GROUND SYSTEMS (AGS) TECHNICAL SESSION
AUGUST 12-14, 2014 – NOVI, MICHIGAN

Measuring the Performance of Active Safety Algorithms and Systems

Tony Gioutsos
Jeff Blackburn
Tass International
17199 Laurel Park, Ste 205, Livonia, Mi 48152

Abstract

In any active safety system, it is desired to measure the "performance". For the estimation case, generally a cost function like Mean-Square Error is used. For detection cases, the combination of Probability of Detection and Probability of False Alarm is used. Scenarios that would really expose performance measurement involve complex, dangerous and costly driving situations and are hard to recreate while having a low probability of actually being acquired . Using a virtual tool, we can produce the trials necessary to adequately determine the performance of active safety algorithms and systems. In this paper, we will outline the problem of measuring the performance of active safety algorithms or systems. We will then discuss the approach of using complex scenario design and Monte Carlo techniques to determine performance. We then follow with a brief discussion of Prescan and how it can help in this endeavor. Finally, two Monte Carlo type examples for particular active safety algorithms (LDW and AEB) will be presented.

INTRODUCTION

In recent years, automotive active safety systems have become more prevalent. Furthermore, these systems will be used as the foundation for the roll-out of autonomous driven vehicles. There are a variety of applications that encompass active safety including: Adaptive Cruise Control (ACC), Forward Collision Warning (FCW), Automatic Emergency Braking (AEB), Lane Departure Warning (LDW), Blind Spot Warning (BLSW), Lane Keeping Assistance (LKA), Pedestrian Avoidance (PA), Intelligent Headlight Systems (IHS) and Cooperative Driving Systems (CDS).

Many sensor systems are used throughout the industry in a solitary or fused manner including: Radar, Laser/Lidar, Video camera, Infrared camera, Active Camera, Ultrasonic, V2X antennas and GPS sensors.

The benefit of these systems to vehicle safety is acknowledged by legislative authorities and consumer bodies, which has led to several initiatives to legislate or reward these new safety technologies. [1]

BACKGROUND

In any active safety system (which could be hardware, an algorithm or the combination), it is desired to measure the "performance". When talking about hardware, performance can be measured in several ways such as reliability, sensitivity, MTBF, etc. But how does one measure algorithm (or a system that contains an algorithm) performance?

An algorithm generally tries to detect something or estimate something (e.g. Lane Departure Warning (LDW) - detecting a lane departure or Adaptive Cruise Control – estimating the distance behind a vehicle and the speed of the vehicle). For the estimation case (Bayes Criteria), generally a cost function like Mean-Square Error is used. For detection cases, the combination of Probability of Detection and Probability of False Alarm are used (having a probability of detection of 1.0 and probability of false alarm of 0.0 is the goal). [2]

Scenarios that would really expose active safety system performance measurement involve complex, dangerous and costly driving situations and are hard to recreate or even obtain.

Using a virtual tool, we can produce the trials necessary to adequately determine the performance of active safety algorithms and systems. In this paper, we will outline the problem of measuring the performance of active safety algorithms or systems. We will then discuss the approach of using complex scenario design and Monte Carlo techniques to determine performance. We then follow with a brief discussion of PreScan and how it can help in this endeavor. Finally, two Monte Carlo type examples on particular active safety algorithms (LDW and AEB) will be presented.

RECEIVER OPERATING CHARACTERISTIC (ROC)

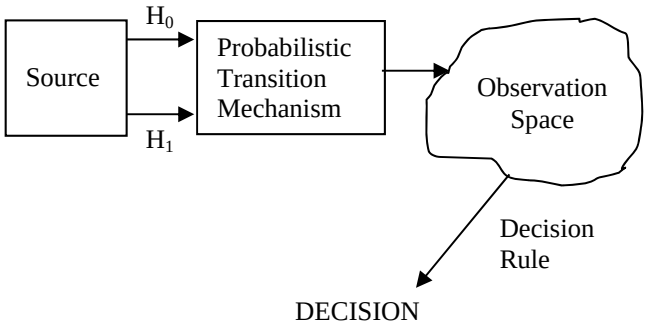

Figure 1 - Components of a decision theory problem

In [2], a thorough explanation of estimation and detection theory is given. In this paper, we will focus on the detection case and in particular the case where the source produces two choices (figure 1). It could be the choice of a vehicle departing from a lane or not. Or it could be the choice of a vehicle about to produce a crash or not. It should be noted that multiple choices from a source or estimating the source, although not discussed in detail in this paper, will produce a similar solution methodology as to that presented in this paper.

The source data is corrupted by a transition mechanism (e.g. noise, scenario complexity, etc.) and given to the observation space where a decision is made. Models for "noise" can be derived from real data (see [3] for an example of this process) or assumptions on the noise can be made (e.g. White Gaussian Noise [2]). The goal, of course, is to produce the same decision as the produced source choice even with the corruption.

In an active safety scenario like Lane Departure Warning (LDW), the vehicle could actually depart from the lane it was in and using sensor data (e.g. camera), we want our decision to correlate to this actual occurrence.

Since in general, the entire problem is rooted in probabilistic theory and the fact that most of detection problems we observe in active safety are Yes or No, we can further simplify the problem. Hence, we want to know the performance of detecting a choice when it exists and a performance of detecting a choice when it does not exist. Placing the problem in a probabilistic environment, the common performance terminology becomes: a probability of detection versus a probability of a false alarm.

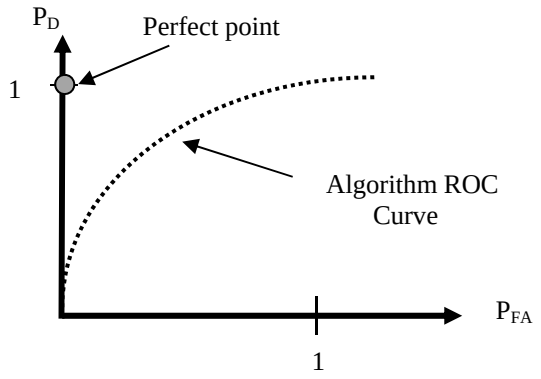

Figure 2 – Receiver Operating Characteristic (ROC)

Figure 2 depicts the classical detection performance measure called: Receiver Operating Characteristic (ROC). It is a two-dimensional performance measurement since the Probability of Detection (P_D) and the Probability of False Alarm (P_{FA}) are both represented for a given algorithm calibration. In figure 2 we depict the perfect algorithm operating point (i.e. $P_D = 1.0$ and $P_{FA} = 0.0$). However, in any probabilistic environment including the real-world this point is simply the goal since it is almost surely not attainable.

An algorithm ROC curve is also depicted. This is an example of the performance of a real-world algorithm. The reason that a line is depicted instead of a point is because the final threshold of an algorithm can be varied to tradeoff P_D versus P_{FA}. This is important when deciding how to operate the final version of the algorithm/calibration. A better algorithm will have a curve moving closer to the perfect operating point. The goal is to get the algorithm ROC curve as close as possible to the perfect operating point and then choose the tradeoff of Detection versus False Alarm.

CURRENT ACTIVE SAFETY ROC

So any active safety system should be measured using a ROC curve. There is an assumption in the industry that by collecting and running real world data we can produce "ROC" curves. The curves are basically never generated but the implication is inherent. The problem with using real world collected data is the fact that even if we use "a million miles" of data collection, most of the data will not test the system in a representative way that corresponds to all real-world events.

Daimler Benz/Mercedes collected all activated Automatic Emergency Braking events for field vehicles equipped since 2013. After five million kilometers of combined travel, only 13 actuations of their AEB system were recorded [4]. Hardly enough to generate truly accurate ROC curves.

Volkswagen reports that current ADAS testing requires over 2 million test km and 1000 test drivers to accomplish. Figure 3 shows the activation frequency for various ADAS technologies per 10,000 km driven.

System	Activation Frequency per 10,000 km	km Until Activation
Distance Warning	40 to 60	170 to 250
Breaking Assistance	1 to 2	10,000 to 20,000
PRE-SAFE Breaking, Level 1	1 to 2	50,000 to 100,000
PaRE-SAFE Braking, Level 2	1	385,000

Figure 3 – Activation Frequency per 10,000 km driven

As ADAS complexity increases, Volkswagen forecasts that a highly automated vehicle will require 100 million km at a cost of several 100 million Euros to be driven to insure sufficient test coverage of relevant scenarios [5].

As pointed out above, even with a million miles driven, the data rarely contains events that would be truly tax the active safety system. Crashes are rare and difficult to capture. Even near crashes are rare. Hence, we end up with much of the collected data being simple false alarm data (i.e. driving with no difficult decision to make)

Furthermore, if we are able to collect a crash or near miss, we are only able to collect that particular event. No variation to that event is available. For example, we could be driving in a vehicle going 40 MPH and another vehicle cuts in front of the vehicle maybe missing an accident by 10 feet. Can we change the real data to make the miss 9 feet? Or 5 feet? Can we change our current speed to 42MPH? Can we add rain to the scene? Change the trajectories of the vehicles? One can see that the variability of real-data is not nearly enough to make an accurate estimate of the ROC.

The cost and time required to collect real data is obviously quite high. This does not even address the safety issue in data collection. Whether it be for the driver of the collection vehicle or in simulating a pedestrian impact, the safety of personnel is paramount.

But there is an even more problematic issue with real data collection. That is the fact that generally speaking it is impossible to know exactly the actual ground truth. In other words, in order to create a ROC, we need to determine if a detection was required or not *exactly*. For example, in a LDW system, we would probably require outside sensing to determine exactly when the departure of the lane occurred or not. Only with exact precision can we measure our active safety system performance.

Without the ground truth available, it is therefore usually the case that the ground truth is estimated. Once that is assumed, then we generally get a ROC point that looks curiously like our perfect point for the real data collected. Because as noted, there are not really many cases that could tax the system and any that do will lead to designing to make sure the correct decision is made.

DOE, THE MONTE CARLO METHOD AND PRESCAN

The Monte Carlo method (or simulation) is used to create a ROC point by simply running the system under test a very large number of times with a statistical representation of real world variations. The method has been used in countless industries including the automotive passive safety sensing area.[3]

The Law of Large Numbers is used as the basis behind the Monte Carlo method. Basically, as we run more trials of a statistically varying system, the results of operating on that system will converge to the actual statistical value expected. In general, a statistical model of a system is created. Speaking in terms of algorithms, a sample path from the statistical model is selected and tested against the algorithm of interest. We then measure the result of the algorithm on that sample path. We then create another sample path and measure again. After a large number of sample paths are tested, we are able to "count" the performance. If we use enough sample paths and statistically model the system correctly, we can produce reliable data.

Although the Monte Carlo method is statistical by its very nature, we can add in elements of a Design of Experiments (DOE) to produce even more sample paths to produce our estimate of performance. For the example given earlier, suppose we were to vary the speed of our collecting vehicle from 35 MPH to 45 MPH in 0.1 MPH increments. Even though this is not statistical in nature, it helps produce more variability in the sample paths allowing for more tests and a better estimate of performance

So combing DOE elements and statistical modeling vis-à-vis the Monte Carlo method seem like the perfect answer to performance measurement (i.e. a ROC curve). It is clear from our discussion on real data that this cannot be done by collection alone. Therefore it is imperative to find another solution.

Virtual simulation has been an extremely popular approach to saving time, money and increasing variability for systems, especially algorithms, under test. PreScan [6] is a virtual simulation tool designed for active safety systems. The tool allows the user to build a complex scenario including road networks, actors (vehicles, pedestrians, etc.), buildings, weather, etc. It also allows the user to define various sensors as well including: radar, lidar, camera, etc. The applications for use include LDW, AEB, BLSW, PA, etc.

Using PreScan will allow the user to create a very large set of tests that can help measure active safety algorithm performance. By using DOE techniques and the Monte Carlo method, PreScan can create thousands of relevant tests that can truly test the capability of the system.

Figure 4 depicts a snapshot of a scene from PreScan. Note that creating pedestrians, buildings, roads, etc. is relatively simple and the scenes can be made as complicated as desired. It is this program that will be used as the basis for our analysis.

Figure 4 – Snapshot of a scene created in PreScan

It should be noted that it is easy to test an algorithm in PreScan. Figure 5 depicts the steps. PreScan creates a Simulink model of the scene and by simply adding in the algorithm under test in Simulink, we can have a closed loop software test. It is also possible to do the testing in hardware or even with a driver-in-the-loop.

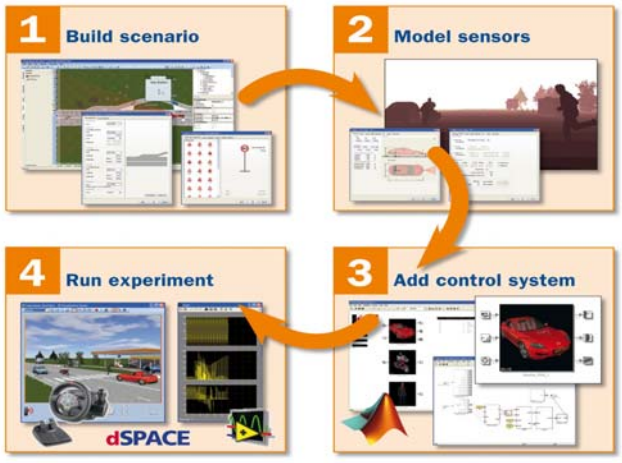

Figure 5 – Building a Test Scenario in PreScan

Once we accept virtual simulation as the approach to use to create a ROC point and measure the performance of our active safety algorithm, it should become apparent that more of the world of active safety system scenarios is covered. Figure 6 depicts a Venn diagram.

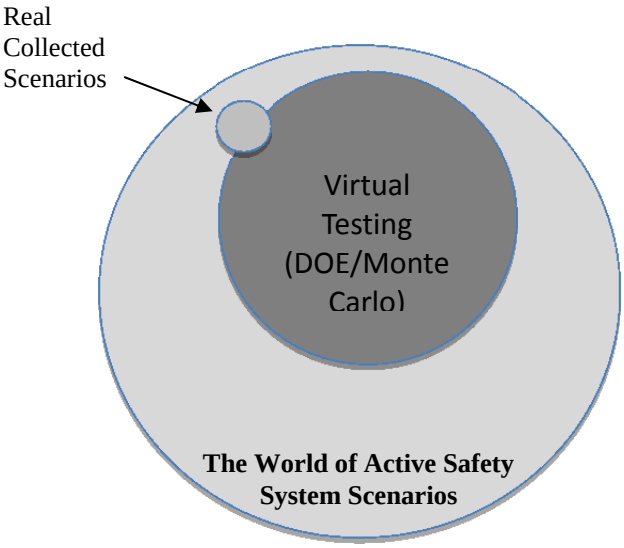

Figure 6 – Real world Active Safety System Scenarios

EXAMPLE 1 – LDW ROC

To show an example of how PRESCAN and DOE/Monte Carlo can be used to measure performance and even calibrate an algorithm, we present a Lane Departure Warning (LDW) example. We use a prepackaged LDW algorithm (comes with PRESCAN) and a camera as the only sensor.

We created a 10 second scene at 50 Hz with the following variables:

* lane width (3.25, 3.5, 3.75) meters

* weather (none, fog, snow)

* light (daylight, evening, night)

* trajectory (straight, lane change, swerve)

* path (straight + 2 bends)

* noise on car's position (none + 2 levels) (i.e. Monte Carlo)

This created 729*500 = 364,500 samples to create one ROC point, we did this for 3 total ROC points (i.e. 3 calibration changes of the LDW algorithm).

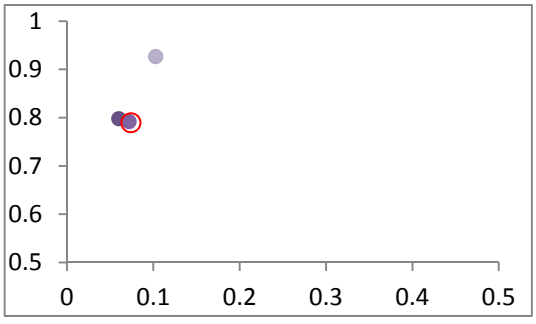

Figure 7 – Three ROC points for an LDW algorithm

It is evident from figure 7 that the performance of the LDW algorithm is not very good based on the scenarios and variations we used in our analysis. But this is the point of this paper: to measure the performance of an active safety algorithm. We can also see that the ROC points do not fall on a monotonic curve. This is because we changed an algorithm parameter instead of just a threshold.

The point that falls inside the other two points, highlighted by the red circle, should never be used in the final calibration. The other two points should be chosen if one prefers a lower false alarm rate or if one prefers a higher detection rate. In the end, this example shows that the LDW algorithm might not be sufficient to meet customer demands in the field and a new approach might be needed (potentially a different calibration might work better).

EXAMPLE 2 – AEB ROC

The second example presented is for an Automatic Emergency Braking (AEB) algorithm. We use the prepackaged AEB algorithm that comes with PRESCAN and requires long-range and short-range Radar sensors.

We varied scenes in the following way:

* 10 test cases

* beginning speed of vehicles (45mph, 36mph, 30mph)

* beginning distance from green to red (60m, 50m, 40m)

* beginning distance between red and blue (5m, 10m, 15m)

* change lane times (near end of experiment, not as near, farther)

* noise on main car's position (none + 1 level)

* noise on target car's position (none + 1 level)

* noise of second target's car (none + 1 levels)

* threshold parameter TTC = [2.9 2.8 2.7 2.6 2.5 2.4 2.3]

This created 7 ROC points. Each point was generated by over 6,000 tests. The 10 test cases were designed in a similar fashion to the one depicted in figure 8. In this test case, all three vehicles begin in the left lane. The blue and green cars continue moving in the left lane at the designated starting speed. The green car, however, switches lanes to the middle lane and stops. This is a false alarm test. The green car should not stop in this scenario. Stopping for false alarm scenarios for an AEB system, besides being a nuisance to the driver, could potentially lead to a crash. Hence, false alarms are very important to avoid.

Figure 9 depicts the ROC points. Note that the false alarms rates are still quite high and this is because we did choose difficult tests in matrix.

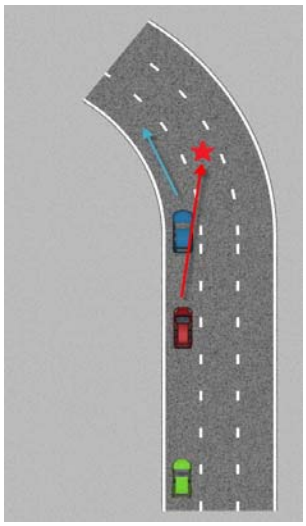

Figure 8 – False Alarm Test Scenario

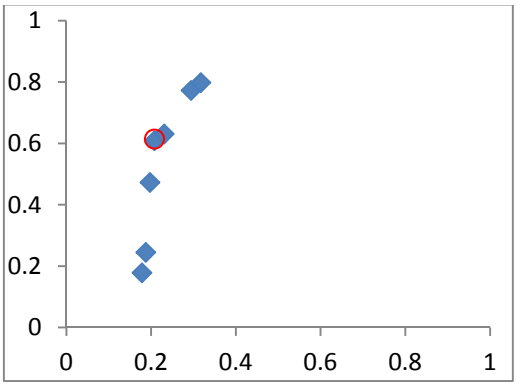

Figure 9 – Seven ROC points for an AEB algorithm

The ROC points do follow a monotonic curve simply because we vary the Time-To-Crash (TTC) threshold. This threshold is the estimated time before a crash. It is a simple threshold and not a complex algorithm calibration change.

The detection probability values are not close to 1.0 because if we do not make a complete stop before the stopped car, then the detection is not counted. However, this could be somewhat misleading. For example, if we were going 45 mph and slowed the car down to 5 mph before contacting the stopped car, a detection is not counted, but the performance is still quite good.

One way to measure this could be to measure the speed of the vehicle when it contacts the stopped vehicle in a histogram fashion. Figure 10 depicts the case using the ROC point circled from figure 9 (i.e. TTC = 2.6). Note that many of the test cases do bring the vehicle to a complete stop. However, many do not, yet few cases are greater than 30 MPH.

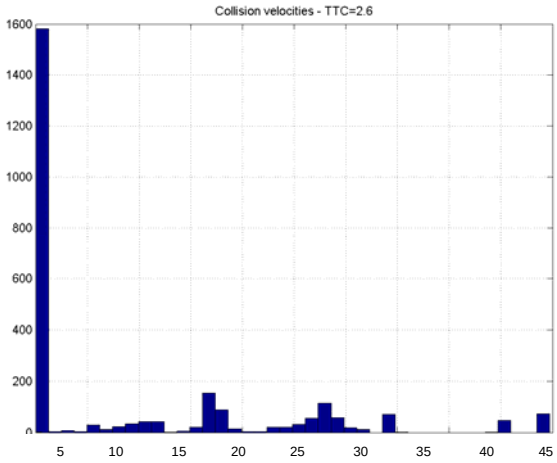

Figure 10 – Histogram of "crashing" velocity using the circled AEB ROC point calibration

Using this histogram of velocity weighted with an "injury saving function" (i.e. a cost function) could be used to replace simple full-stop detection as the only detection. Figure 11 depicts such a function.

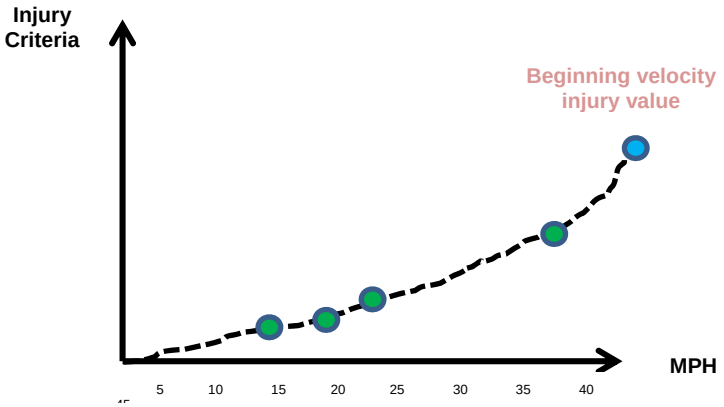

Figure 11 – Injury Saving Function

I. Conclusions

In this paper, we outlined the classical methods for measuring estimation and detection algorithm performance. We argued that real-world data collection is not nearly complete to perform this endeavor. We then detailed a method using Monte Carlo, DOE and virtual simulation to produce better estimates of performance. Finally, we showed two examples of this procedure.

REFERENCES

[1] Roy Bours, Komal Rauf , Kajetan Kietlinski *"A method for developing AEB Systems based on Integration of virtual and experimental tools"*, ESV 2013, Paper Number 13-0347

[2] H. L. Van Trees, *Estimation, Detection and Modulation Theory*, John Wiley and Sons, 1968

[3] Tony Gioutsos and Michael Piskie "Automobile Crash Modeling and the Monte Carlo Method", SAE Technical Paper 920480, 1992

[4] Charles Anthony Kawashima, " Field Benefit Analysis of Advanced Forward Collision Warning Systems - Mercedes-Benz Collision Prevention Assist", SAE Government and Industry Meeting, Jan 22-24 2014, Oral only

[5] Dr. Arne Bartels, "Online Services & Testing for Highly Automated Driving Functions", TRB Autonomous Vehicle Workshop, Stanford University, 17 July 2013

[6] www.tassinternational.com/prescan