

Hazard Warning Performance in Light of Vehicle Positioning Accuracy and Map-Less Approach Path Matching

Andreas Barthels, Christian Ress, Martin Wiecker, and Manfred Müller
 Ford Motor Company

ABSTRACT

Vehicle to Vehicle Communication use case performance heavily relies on market penetration rate. The more vehicles support a use case, the better the customer experience. Enabling these use cases with acceptable quality on vehicles without built-in navigation systems, elaborate map matching and highly accurate sensors is challenging. This paper introduces a simulation framework to assess system performance in dependency of vehicle positioning accuracy for matching approach path traces in Decentralized Environmental Notification Messages (DENMs) in absence of navigation systems supporting map matching. DENMs are used for distributing information about hazards on the road network. A vehicle without navigation system and street map can only match its position trajectory with the trajectory carried in the DENM. The framework supports simulating random and systematic positioning error influences, network effects, and different algorithms for approach path trace matching and DENM approach path trace generation. The tool allows assessing the observable system performance in terms of correct and incorrect matching of approach path traces resulting in true and false, positive and negative hazard warnings provided to the user. We provide an exemplary sensitivity analysis with a simple approach path trace matching implementation depending on varying positioning error influences and develop error bounds for acceptable system performance depending on typical street network geometry in the USA and Germany. The presented framework supports both a frontend for visualization, as well as headless operation for conducting large scale Monte-Carlo simulation runs.

CITATION: Barthels, A., Ress, C., Wiecker, M., and Müller, M., "Hazard Warning Performance in Light of Vehicle Positioning Accuracy and Map-Less Approach Path Matching," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.* 10(2):2017, doi:10.4271/2017-01-0073.

INTRODUCTION

Since a few decades, vehicle to vehicle and vehicle to infrastructure communication has been subject of research and technology development. In order to provide as much benefit as possible to each individual driver, the penetration rate of vehicles equipped with this technology is important. That is why there are a number of cross OEM collaboration projects in order to harmonize the technology for exchanging data between vehicles.

In the beginning, dedicated short range communication (DSRC) was the prime method of choice for networking vehicles with each other, due to the limitation of early mobile networking standards. Although the technology has been readily available and tested in pilots such as the Vehicle Infrastructure Integration (VII) initiative of the US DOT, and the sim^{TD} in Germany, making the business case for introducing dedicated hardware for every vehicle was delaying the introduction.

Due to this delay, the US National Highway Transportation Safety Agency (NHTSA) decided to propose a mandate for DSRC based communication technology to be equipped with every new car built in the future [1]. In other regions, such as Europe, there has been a regulatory mandate to equip vehicles with other mobile networking solutions, such as telematics control units with support for Accident Emergency Call Systems (AECS) [2].

While the existing standards for direct short range communication, as well as for mobile network communication have different qualities [3], the upcoming mobile network standardization aims to replace both for the use case of networking vehicles. The goal is to offer both low latency, ultra reliable communication, as well as the possibility to exchange large quantities of data with distant endpoints. Popovski introduces the motivation to support and the terminology to describe ultra-reliable communication in 5G in [4].

The scope of this paper does not include the discussion of the different networking technologies; instead we want to focus on the anticipated application performance in regards to vehicle position accuracy. The cross OEM initiatives to introduce vehicle to vehicle and vehicle to infrastructure communication have identified a number of use cases for introduction scenarios. These use cases center around assisting the driver by providing hazard warning information along the direction of travel.

Day 1 Use Cases

In this paper, we follow the classification of the European CAR 2 CAR communication consortium for day 1 use cases [5]. The analysis in this paper focuses around the following quasi stationary incident warnings.

- Hazardous Location Warning
- Traffic Jam Ahead Warning
- Stationary Vehicle Warning
- Road Work Warning

We assume that an On-Board Unit (OBU) has received stationary DENMs for incidents above, and needs to trigger a hazard warning via the vehicle's human machine interface (HMI). The information is presented in order to prepare drivers for the following incidents along the way of travel.

Related Work

Related work includes a wide variety of simulation environments for both vehicle movement as well as network simulation. In [6], Lochert et al. presented a simulation framework coupling vehicle movement with network simulation.

On top of pure GPS receiver sampling and filtering of data, a technique called dead reckoning has proven to be both capable of extrapolating a vehicle's position during outage, and improving the positioning quality in noisy conditions. For this, the vehicle's sensors are being used to compare the measured distance travelled using wheel ticks and acceleration sensors with the GPS receiver supplied position and heading values. Mattos [7] presented such an integrated system, showed how dead-reckoning diverges increasingly over time, and how it synchronizes back upon availability of GPS position.

In the case in which a vehicle has total knowledge and a dense map in the vehicle, sophisticated map matching can be applied. In [8], Newson and Krumm present a hidden markov model (HMM) based approach. The approach combines the observed distances travelled of the vehicle with the distance between the road network segments. They show that depending on the GPS sampling frequency and the standard deviation, the performance for an exemplary run through Seattle is flawlessly matching the vehicle to the right road in the map.

Contribution

This paper analyzes the region specific sensitivity of road networks to vehicle positioning accuracy. This is done using a simplistic approach path, or trace matching algorithm, which was developed as part of a European Union funded collaboration project called drive c2x [9]. Testing the performance of a simplistic hazard matching algorithm is done in order to test the viability to equip even lowest specification vehicles with the new technology.

Two error modes are being investigated: Scaling white noise to measure robustness of simplistic matching versus position and heading errors, and scaling systematic offset and heading divergence, which typically occur using dead reckoning.

Structure

The following section describes the dynamically linked toolchain, which allows modular assessment of different connected applications, like hazard warning in light of vehicle position accuracy. We continue

describing both the experimentation setup and results. Finally, we provide a conclusion for the analysis of simplistic hazard warning for two typical urban environments in Europe and North America.

SIMULATION TOOL ARCHITECTURE

In order for the simulation tool to import and evaluate the existing methodology for hazard warning and DENM matching, it is built using an OSGi service oriented framework.

OSGi Based Modular Framework

The OSGi platform allows dynamic linkage of different modules for navigating vehicles using different underlying map sources, different applications running on these vehicles, providing a user interface and automated experimentation. The following subsections detail the different OSGi services offered in the framework.

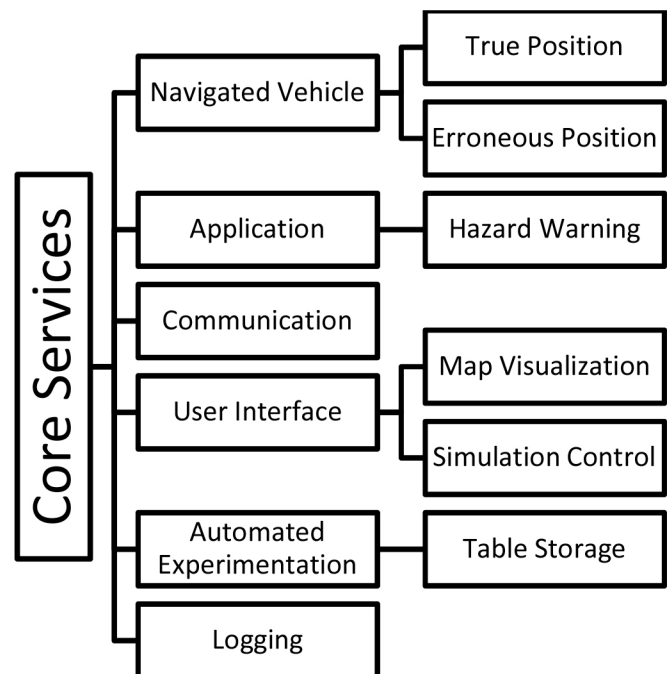


Figure 1. Architecture of Simulation Framework

Simulation Core Services

The simulation core service implementation holds a list of agents which understand a notion of time. It initializes agents, and increments the time until an agent has reached its detached state. Furthermore, it provides central access to a random number generator, which can be seeded appropriately to repeat the experiments which we conducted.

Navigated Vehicle Services

The navigated vehicle service implementation constructs vehicle simulation agents, which follow specific driving models and routes. This service includes the computing the real position, heading and speed of each vehicle, as well as configurable and combinable error models which distort these values.

Additionally, it exposes an interface to compute approach paths towards hazards placed in the road network.

For constructing the vehicle and hazard simulation agents, the navigated vehicle service implementation gets configured with a rectangular bound of two latitude/longitude coordinates.

Each vehicle agent may have a tree of other simulation agents attached to it. Every simulation agent in a vehicle agent's tree is synchronized with the vehicle, meaning the time base is the same and once the vehicle detaches with the simulation core services, so do the synchronized simulation agents.

Hazard Warning Services

The hazard warning service implementation is attached and synchronized with its associated vehicle agent. It can sample the real and erroneous vehicle position and subsequently match the vehicle with all approach path traces within its vicinity.

It runs the trace matching algorithm on all approach paths for both the true as well as the sampled erroneous vehicle position.

Once a positive match occurs, a warning is issued via the HMI and it is only withdrawn as soon as the match quality increases over a hysteresis threshold.

User Interface Service

The user interface service may be used for introspection of the simulation. It may provide the map with an overlay of the vehicle coordinates as shown in [Figure 2](#).

The vehicle agents' movements, their true and erroneous position, as well as hazards and the approach path matches are put in an overlay of the street map of the area.

Telematics Communication Simulation

Each simulated application agent can connect to the communication service and hence exchange data and see only what was being sent to it. The analysis of propagation times and effects of penetration rate are not scope of this paper, and hence the telematics communication simulation service is not described in more detail.

Experimentation and Automation Service

The experimentation and automation service facilitates the parameter study presented in this paper by automatically performing configurable and repeatable simulation runs.

For each simulation run, an output file suffix and parameter set may be defined in a table listing the experiments to be run. An experiment finishes either after a maximum simulation time, or whenever every simulation agent (vehicle) has reached its final destination and does not change its state anymore over time (detached state).

The experimentation service gathers lifecycle events, such as vehicles reaching their destination and detaching, as well as the results for matching the true and erroneous position to the hazards within proximity.

OpenStreetMap Based Geometry

We use OpenStreetMap [10] data for analyzing and comparing the sensitivity of hazard warning performance in regards to road network layout.

Position Error-Model

[Figure 2](#) depicts a trajectory exhibiting a combined position error model with both offset diverge and white noise components. Each position error model described hereafter is assumed to get a vehicle position as input, i.e. Latitude, Longitude, Heading, and Speed, adds error terms and provides a vehicle position as output.

Offset Diverge

The offset diverge Position Error Model is used to simulate divergence of dead reckoning and resynchronization. The offset diverge (OD) error model first samples a normally distributed latitude and longitude offset O with parameters

$$O_{lat}^{OD} \sim N(\mu, \sigma), \text{ and } O_{lon}^{OD} \sim N(\mu, \sigma) \quad (1)$$

The offset divergence is then targeted into normal distributed heading offset with parameters:

$$O_{heading}^{OD} \sim N(0, \sigma) \quad (2)$$

The error term in the vehicle position is assumed to drift linearly towards the sampled offset position and in direction of the disturbed heading. The drift is assumed to happen over a random number of sample counts. After the sample count number is reached, a reset back to the true position shall occur. This sample count is computed through sampling a normal distribution and rounding to the next applicable positive integer.

$$O_{samplecount}^{OD} \sim N(\mu, \sigma) \quad (3)$$

We assume to have a systematic error when dead reckoning and GPS signal diverge, for instance due to driving through a tunnel, or urban canyon. Because of this, we model the sample count as a Gaussian process instead of a Poisson process.

[Figure 2](#) depicts the offset diverge error with the parameters listed in [Table 1](#).

Table 1. Offset Diverge parameter set used for illustration

Random Variable	Distribution Parameter
O_{lat}^{OD}	N(10 m, 5 m)
O_{lon}^{OD}	N(10 m, 5 m)
$O_{heading}$	N(0.0 °, 5.0 °)
$O_{samplecount}$	N(30, 5)

White Noise

The white noise (WN) error model adds normal distributed offsets to the latitude and longitude values, centered on the latitude and longitude values given as input.

The white noise error model additionally distorts the heading to point in the direction of the distorted sample point. Distorting the heading in this way simulates the relative uncertainty of heading in case of slow movement. The slower the vehicle moves, the higher the error in heading due to the white noise displacement of the position.

Table 2. White Noise parameter set used for illustration

Random Variable	Distribution Parameter
O_{lat}^{WN}	N(0, 1 m)
O_{lon}^{WN}	N(0, 1 m)

Combined Position Error Model

Since the aforementioned position error models have the same domain as well as co-domain, they can be composed. The combined position error model allows composing a list of position error models.

For the experimentation described in the following sections and the position error model example in Figure 2, we composed an Offset Diverge with a White Noise Error Model, accounting for random white noise as well as systematic error terms.

$$e_{combined} = e_{WhiteNoise} \circ e_{OffsetDiverge} \quad (4)$$

Approach Path Trace Generation

The set of approach path traces for each hazard is computed using the underlying map. The computation starts with the location of the incident and recursively traverses all incident routes into it. Between two nodes on the road, a new trace point is introduced depending on the deviation of the current position to the position added to the trace for the last time. The deviation is a function of angle and distance and defined as follows:

$$\begin{aligned} &(\text{distance} > \text{STRAIGHT_DENM_SPACING}) \text{ OR} \\ &(\text{angle} > \text{BEARING_THRESHOLD}) \end{aligned} \quad (5)$$



Figure 2. Illustration of Vehicle Agent trajectory with transparent erroneous vehicle position syncing back to solid true trajectory in the city of Cologne, Germany. Underlying map tile from [OpenStreetMap.org](https://www.openstreetmap.org)

Table 3. Parameters used for approach path generation along a road

Parameter	Threshold value
STRAIGHT_DENM_SPACING	200 m
BEARING_THRESHOLD	10°

Every time a new node is reached in the road network, the algorithm recursively branches and forks the existing approach path trace. This is depicted in Figure 3. The hazard is assumed to be sampled on the intersection in the middle. From this node in the road network on, all incident roads are being traversed and sparse approach path trace coordinates are being overlaid. The approach path coordinates are illustrated using yellow circles.

Inspecting the approach path from the south, one can observe that more coordinates are spent, when the road is on a curved segment, due to the threshold in degrees of curvature. On a straight, a new coordinate is included once the distance threshold is exceeded.

Depending on the road class, the recursive algorithm stops traversing, after covering a minimum termination length. The values for the different road classes which are common in OpenStreetMaps, are depicted in Table 2 below.

A parameter study as to how to choose the thresholds for approach path generation is out of scope of this paper. Nevertheless, we provide an analysis of observed approach path trace quality in the next section.