Connected Motorcycle Consortium

CMC Whitepaper

Path Prediction

for PTWs

This white paper document is intended to give an overview of Path Prediction possibilities and limitations for PTWs, based on the current technical state of the art, related to different sensors and algorithms.

Document Information

Document Title:	Path Prediction for PTWs (whitepaper)
Version:	1.0
Release Date:	07/04/2023

Disclaimer

Index

1 Preamble

Path prediction is a key pillar for systems like Advanced Driver Assistance Systems (ADAS) or Cooperative Intelligent Transport Systems (C-ITS) to recognize dangerous situations ahead. For motorcycles (PTWs), path prediction will help to reduce accidents. At the same time, to predict the future path of a PTW, challenges remain. The vehicle dynamics of PTWs differ fundamentally from cars and are more complex.

The principle of path prediction relies on using vehicle data to detect where a vehicle is heading to and that waypoints intersect with the predicted waypoints of another vehicle. In this case, a warning will be issued to prevent a collision.

This white paper document is intended to give an overview of possibilities with the current technical state of the art, related to different sensors and algorithms (e.g. Inertial Measurement Unit (IMU), Global Navigation Satellite System (GNSS)). It further highlights limitations regarding curvature calculation and path prediction on PTWs.

For this white paper, the Connected Motorcycle Consortium (CMC) has decided to focus on a typical ‘left turn’ scenario and use it as an example for path prediction, since this case often involves severe or even fatal consequences.

The level of path prediction is defined as follows:

-Level C: Based on information provided by the standard Cooperative Awareness Messages (CAMs) (Chapter 5.1 “Available parameters”, or European Telecommunications Standards Institute (ETSI), EN 302 637-2 - V1.4.1). Level C is achieved by constant velocity, heading and position using well-known information from CAMs. Level C path prediction cover straight riding/driving.

-Level B: The trajectory calculated as a radius of curvature based on instantaneous state observed with vehicle sensors. In addition of information that Level C used, Level B uses constant turn rate, additionally the radius of curvature (also included in CAMs, but less tested in Day 1 implementations) calculated by ego vehicle. Level B path prediction cover steady state (constant curvature) riding/driving.

-Level A: More advanced and complex methods such as contextual behaviour predictions using machine learning, advanced sensing technology such as cameras. Level A is not specific at this moment. However, Level A path prediction will cover to follow road/lane, turns and lane changes.

This white paper focuses on two variants of path prediction, Level C and Level B. Next possible steps in path prediction are listed in Chapter 4. These are heading in the direction of level A. Using this comparison of Level C and Level B, a consideration is made as to what technological advances would be needed to make acceptable path prediction.
Acceptable path prediction refers, for example, to giving the driver enough time to react when a warning is issued. Based on available research, under certain circumstances brake reaction times between PTW riders and passenger car drivers may even differ up to some seconds [1].

2 Explanation of Path Prediction in left turn scenario

2.1 Why left turn scenario is used as an example

Among the various accident scenarios, the left turn scenario is selected for the study of Path Prediction because left turn accidents are more likely to result in serious or fatal injury compared to other scenarios.
The characteristics of left turn accidents are explained below.
To pursue the goal "improving motorcycle rider safety and comfort", CMC has studied the most frequent PTW accident scenarios in the German In-Depth Accident Study (GIDAS) database in which PTWs are involved (Figure 2.1). Out of those accident scenarios, crossing traffic and left turn scenarios add up to 25.6% of the total of PTW accidents.

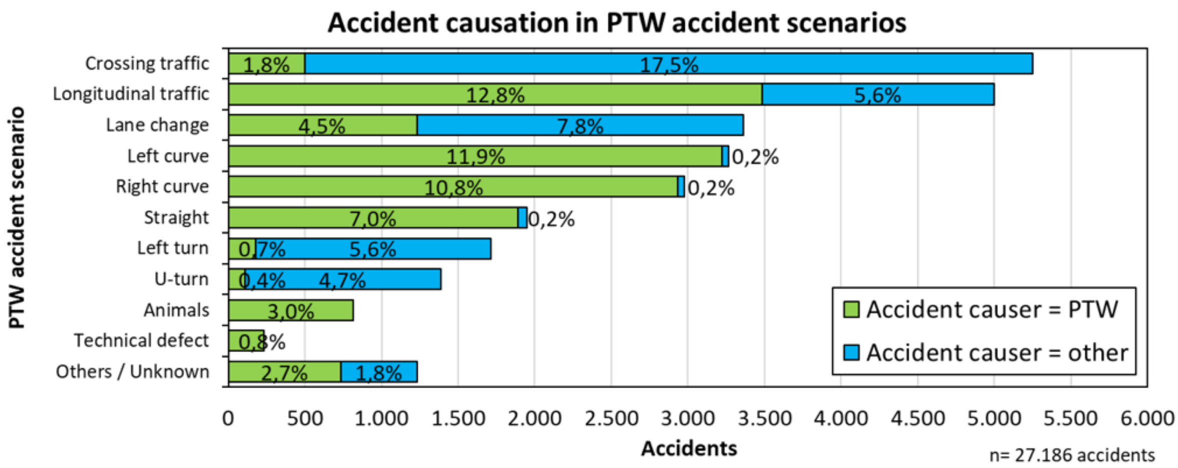


Figure 2.1: Accident causation in the PTW scenarios

According to GIDAS, the "accident type 211" is the most frequent accident scenario within the category of left turn accidents (Figure 2.2). In the typical "accident type 211", the PTW is the straight running vehicle, and has the right of way. Making both vehicles, car and PTW aware of each other could prevent this type of accident.

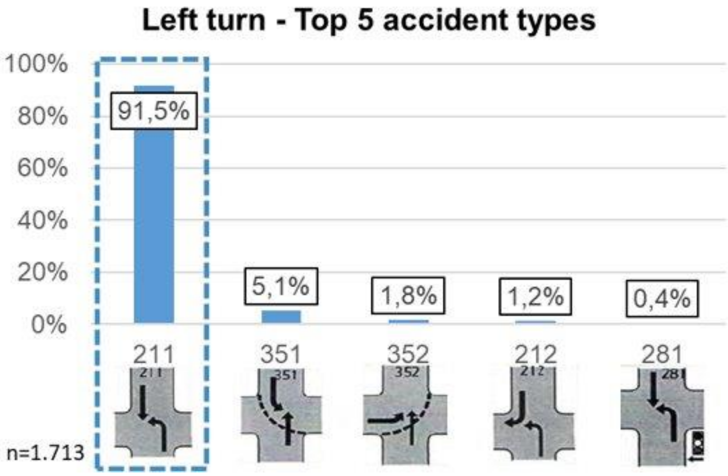


Figure 2.2: Left turn - Top 5 accident types

35 percent of participants involved in the accident type 211 are seriously injured, and 3 percent suffer fatal injuries. The expected benefits of ITS for the use case left turn would be fewer accidents, mitigated injuries and fewer dangerous situations for following traffic.
A typical left turn scenario is shown at the left side of Figure 2.3. Participant A (red car) is turning left at a crossing while participant B (PTW) is coming from the opposite direction. There are also cases in which the view is obstructed while participant A is turning left and therefore cannot see the participant B coming from opposite direction, as shown at the right side of the Figure 2.3.

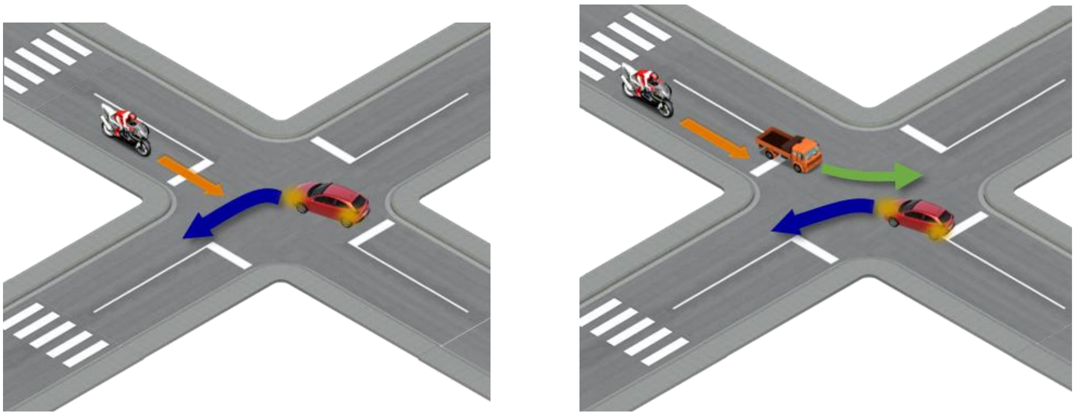


Figure 2.3: Typical situation of a left turn scenario

2.2 Path Prediction of Level C and B in left turn scenario

As mentioned in the preamble this document is focusing on two variants of path prediction, Level C and Level B. Within this document the confidence level has not been assessed. Based on a comparison of path prediction of Level C and Level B, a suggestion can be made as to what technological advances would be needed to come to an acceptable path prediction in the future, such as Level A.

2.2.1 Level C

2.2.1.1 How to calculate Path Prediction of Level C

Level C uses referencePosition, heading and speed of CAM.
If the current position is expressed as "Pt", heading as "Heading" and predict position as "Pt+Δt", the equation is as follows (Figure 2.4) except acceleration. [2]

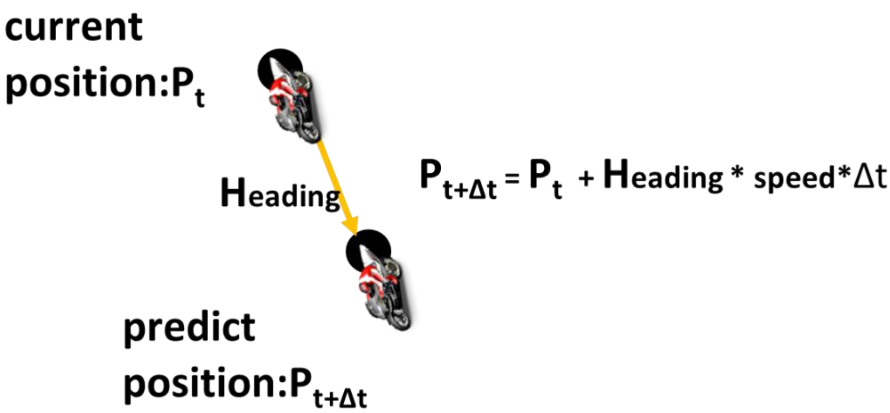


Figure 2.4: Predicted position of Level C

There are various methods to calculate collision risk. The following shows the "Ghost Vehicles" method. In this method, ghosts for notification and warning are calculated from the current vehicle position, heading and speed. The locations of the surrounding vehicles ghosts are predicted as well from the information in the received CAM. Then, the collision risk is calculated from the distance between the ghost of the ego vehicle and the ghosts of surrounding vehicles. If distance between each ghost for notification is smaller than the threshold, a notification is submitted. If the distance between each ghost for warning is smaller than the threshold, a warning is submitted.

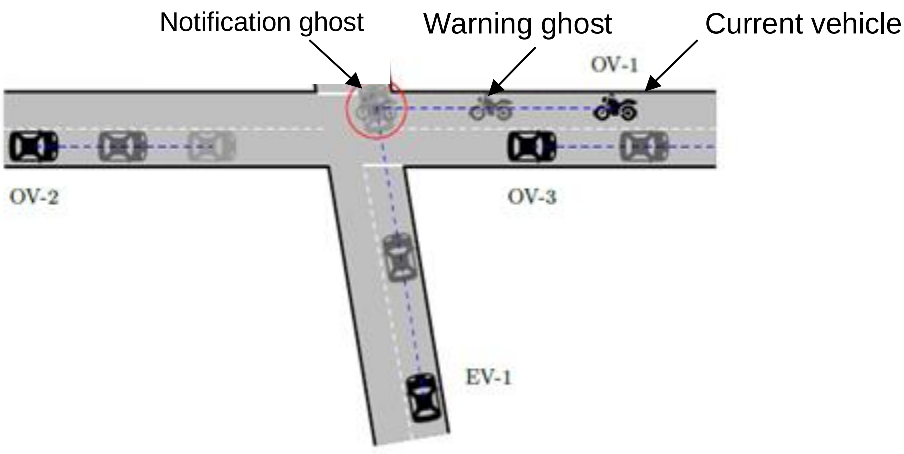


Figure 2.5: Methods of calculating collision risk - "Ghost Vehicles"

This method can be enhanced from a discrete ghost to a continuous rectangle ghost zone (Figure 2.6). If a rectangle ghost zone of notification is crossing, a notification is submitted. If a rectangle ghost zone of warning is crossing, a warning is submitted.

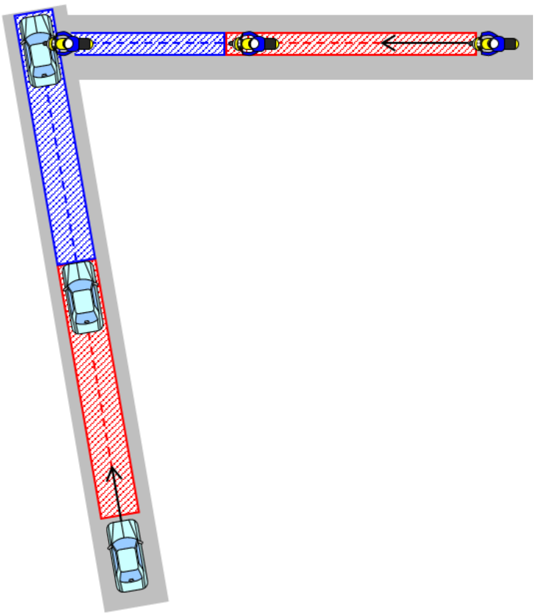


Figure 2.6: Methods of calculating collision risk – Enhanced "Ghost Vehicles"

2.2.1.2 Sending CAM

A CAM is sent periodically. The function flowchart from Service-In to Service-Out of a PTW sending CAM is indicated in the following Figure 2.7.

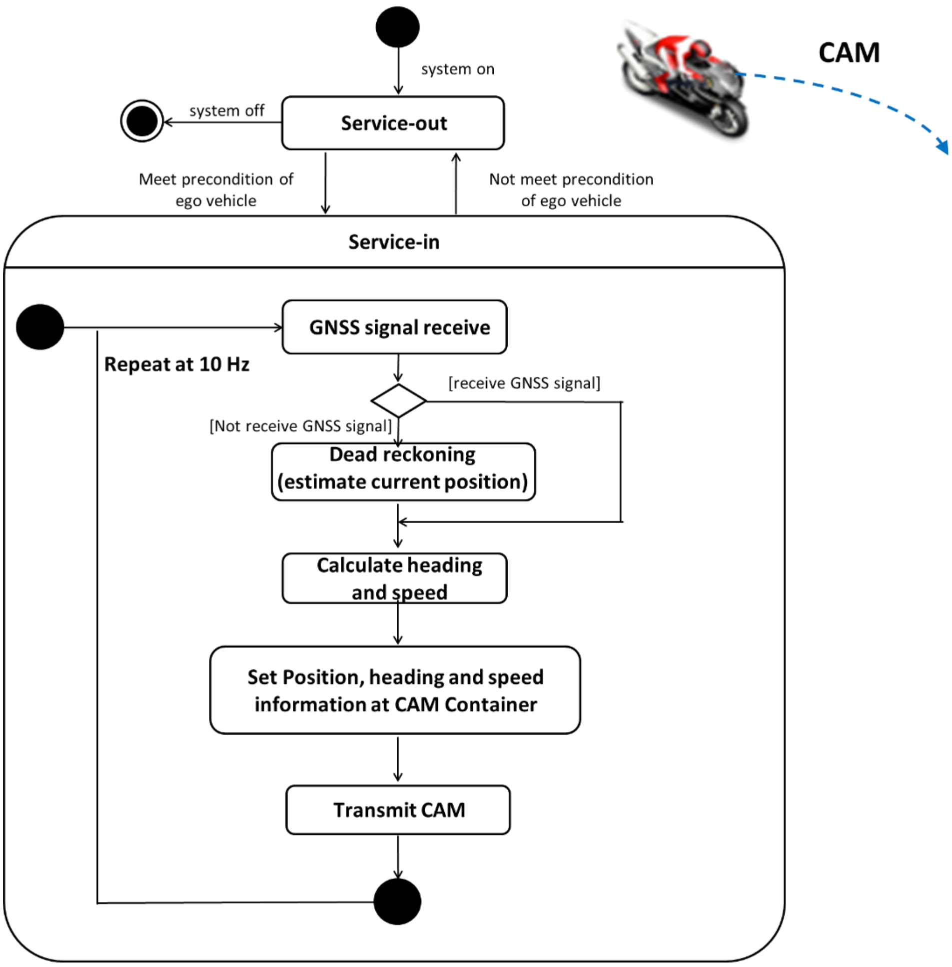


Figure 2.7: Flowchart of sending CAM - Level C

2.2.1.3 Receiving CAM

Cars and PTWs nearby receive the CAM. When receiving a CAM, cars and PTWs calculate the probability of a collision between the CAM sender and the own vehicle and issue a warning through a Human-Machine Interface (HMI) notification to the driver/rider if necessary. The function flowchart from Service-In to Service-Out of receiving CAM is indicated in the following Figure 2.8.

In the case of level C, the probability of a collision is calculated from the predicted position shown in Figure 2.5 or Figure 2.6 and so on.

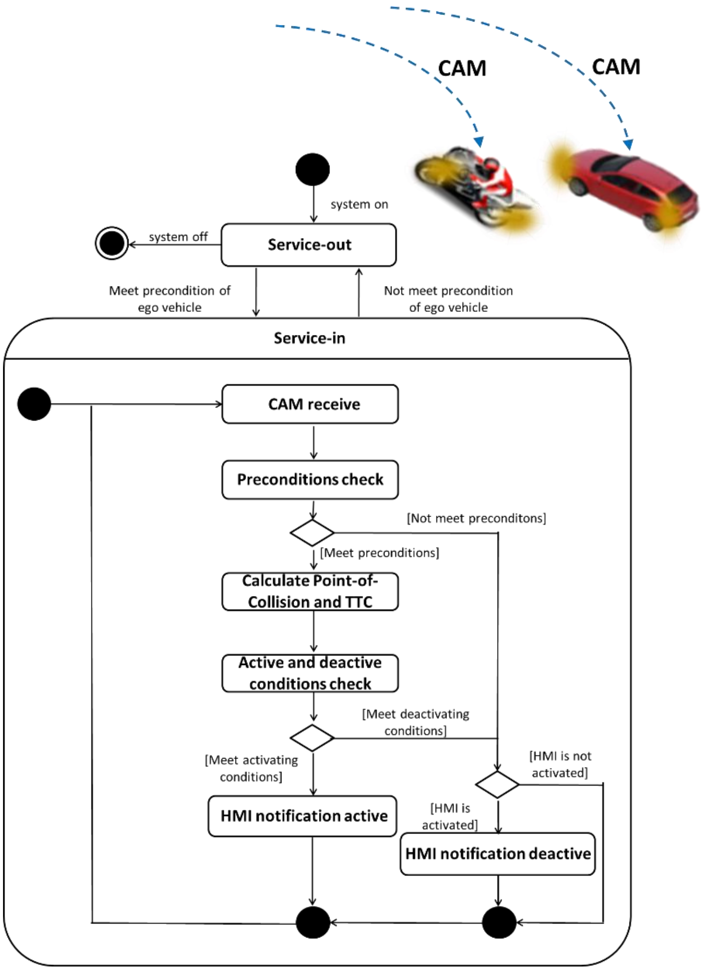


Figure 2.8: Flowchart of receiving CAM - Level C

2.2.1.4 Preconditions – Activating conditions – Deactivating condition.

As mentioned above (2.2.1.3), when a CAM is received, both car and PTW ITS application will check preconditions. If preconditions are met, Point-of-Collision and TTC are calculated. Then, activating and deactivating conditions are verified by the system and HMI notifications are displayed accordingly.

The tables below show preconditions, activating conditions and deactivating conditions of PTWs.

All of the following preconditions (PC_1 to PC_10) shall be satisfied every time before ”Calculate Point-of-Collision and TTC” step.

Table 2.1: Preconditions of ego vehicle (PTW receive CAM)

#	Item	Condition
PC_1	Ego vehicle	PTW
PC_2	Speed range	-
PC_3	Location	-
PC_4	Road type	-
PC_5	Time	-
PC_6	Weather	-
PC_7	Other condition	-
PC_8	Out of scope	-
PC_9	Left turn indication	Ego vehicle intends to turn left – by turning on the turn signal*
PC_10	Acceleration	Not decreasing as for stopping or sufficiently slowing down before turning left

*Left turn indication is necessary until the technology and reliability will increase in the future to detect intention to turn left without indication.

All of the preconditions of target (PC_11 and PC_12) shall be satisfied before proceeding to the activating and deactivating conditions check.

Table 2.2: Preconditions of target (PTW receive CAM)

#	Item	Condition
PC_11	Other vehicle	Other vehicle is on the same road as the ego vehicle approaching to this, and driving in the opposite direction
PC_12	Distance between ego vehicle and other vehicle	< 100m

The activation and deactivation requirements of PTW receive CAM LTA are stated below.
All activating conditions must be satisfied to trigger the warning.
To deactivate the warning, all deactivating conditions must be satisfied.

Table 2.3: Activating conditions of LTA (PTW receive CAM)

#	Activating conditions
AC_1	TTC is calculated and a warning is sent when below 6.5s

Table 2.4: Deactivating conditions of LTA (PTW receive CAM)

#	Deactivating conditions
DC_1	There is no calculated Point-of-Collision any more for 1 second or more

2.2.2 Level B

2.2.2.1 How to calculate Path Prediction of Level B

In level B, the vehicle's future trajectory is calculated from a radius of the curvature based on instantaneous state observed with vehicle sensors.
In the following, the case of the PTW turning left is being considered rather than the car turning left; this is because this paper is focusing on path prediction for PTWs and a PTW turning left represents a more difficult case for path predication compared to a PTW going straight. [3]
In order to calculate collision risk in the Level B "Ghost Vehicles" method, ghosts for notification and warning are calculated from the curvature in addition to current vehicle position, heading and speed (Figure 2.9). This figure shows the case of PTW turn left. If the curvature equals zero as special case, the "Ghost Vehicles" are located on a straight line.
This method can be enhanced in the same way as Level C (Figure 2.10). In other words, this method can be enhanced from discrete ghost to continuous arc ghost zone.

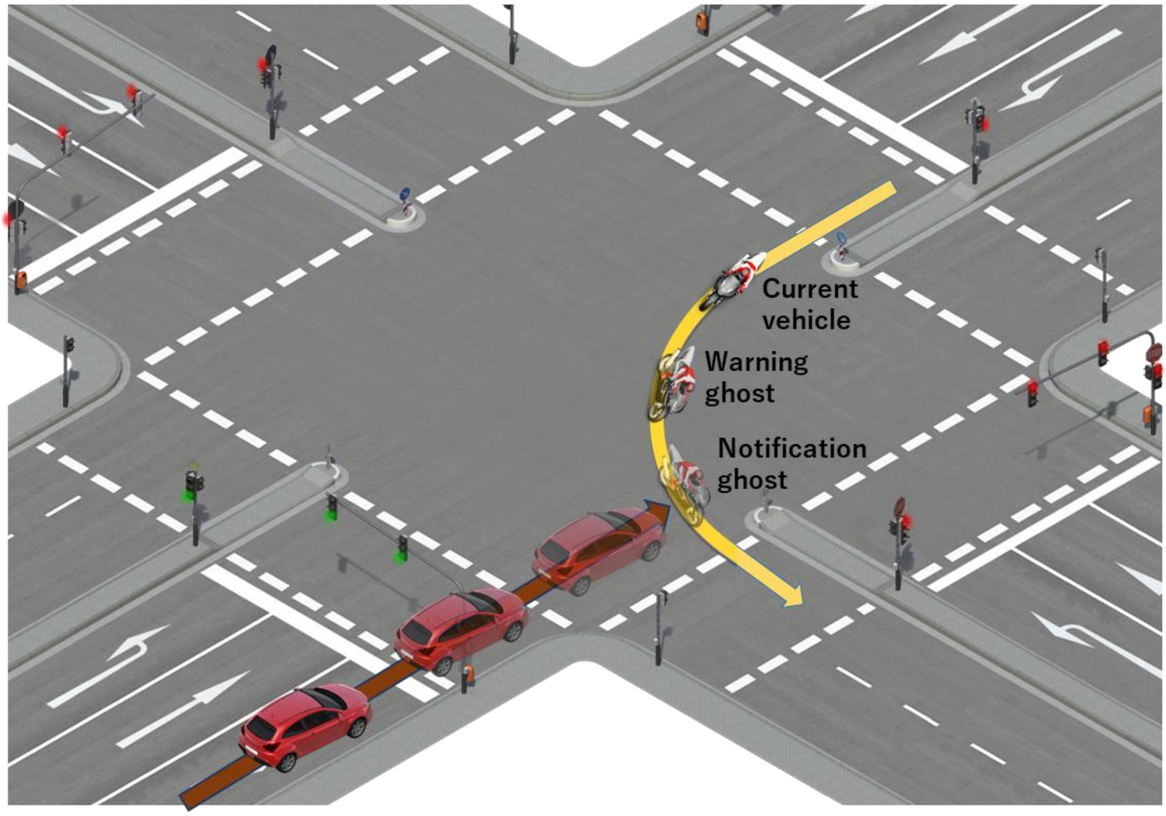


Figure 2.9: Methods of calculating collision risk - "Ghost Vehicles" with PTW curvature

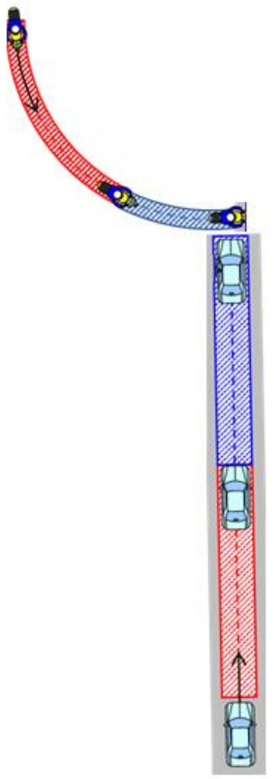


Figure 2.10: Methods of calculating collision risk – Enhanced ”Ghost Vehicles” with PTW curvature

2.2.2.2 Sending CAM – Flow chart

The flowchart of Sending CAM is the same as the flowchart of level C except related to the curvature shown in red (Figure 2.11**Error! Reference source not found.**).

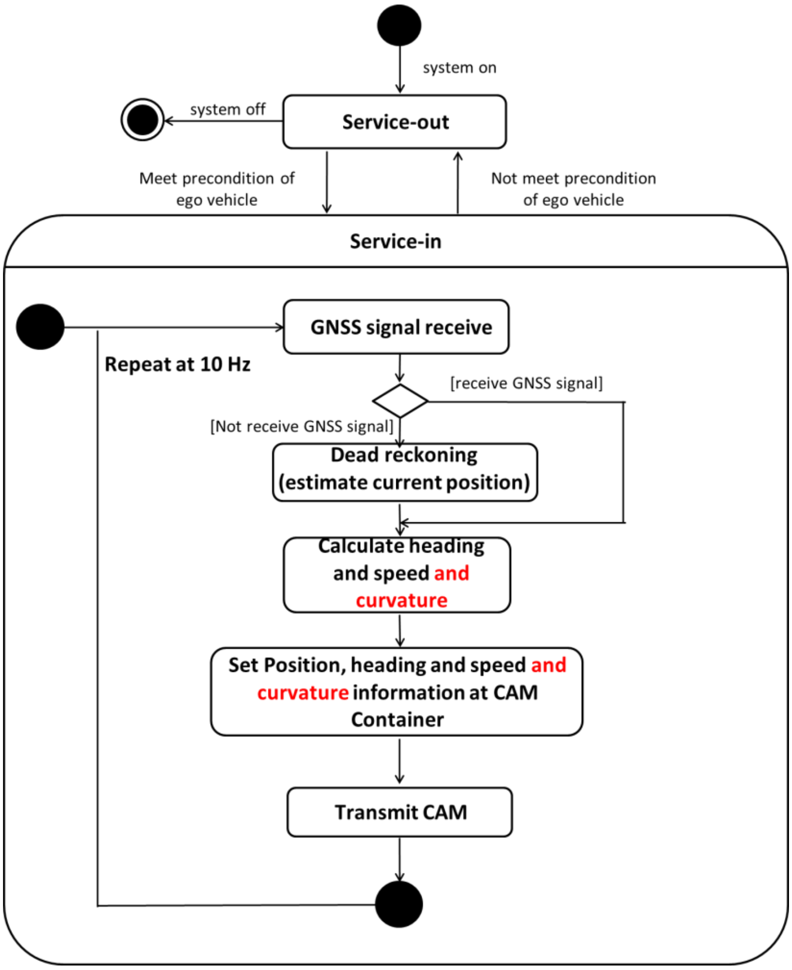


Figure 2.11: Flowchart of sending CAM - Level B

2.2.2.3 Receiving CAM – Flow chart

The flowchart of receiving CAM is the same as the flowchart of level C. The process related to the curvature is included in “Calculate Point-of-Collision and Time-To-Collision (TTC)” part of this flowchart (Figure 2.12**Error! Reference source not found.**).

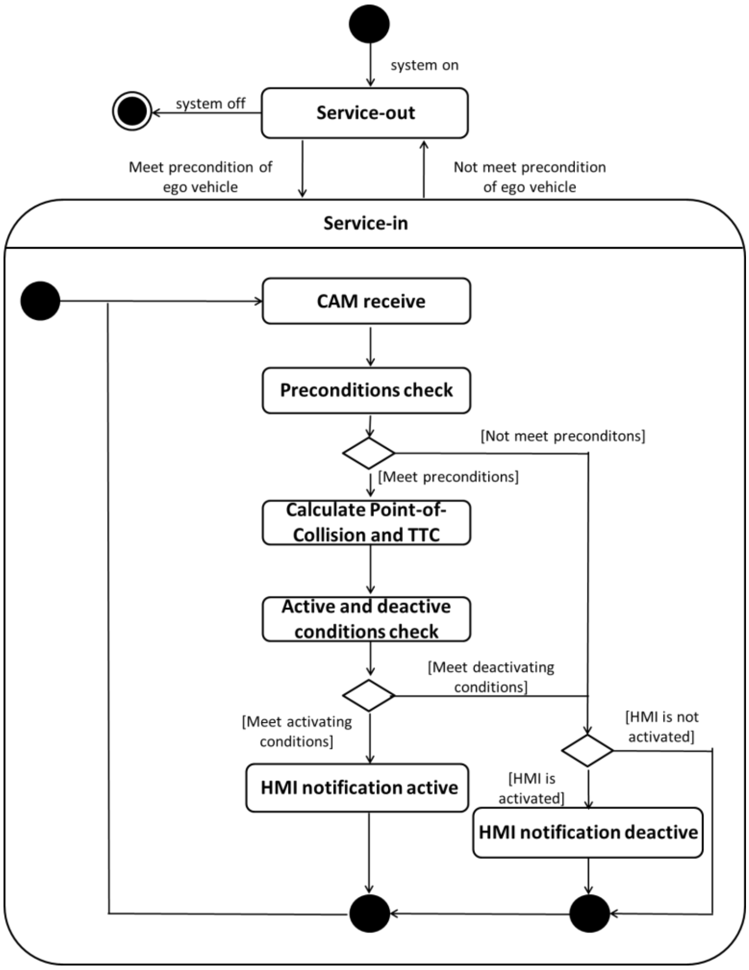


Figure 2.12: Flowchart of receiving CAM - Level B

2.2.2.4 Preconditions – Activating conditions – Deactivating condition

Preconditions, activating conditions and deactivating conditions of level B are the same as those of level C shown in 2.2.1.4.

3 Performance check by Simulation

3.1 Introduction

Predicting the path of a motorcycle is a desired technology, but there are many challenges. It is important to understand what performance is achieved with currently available technologies, and to what degree new technical solutions will be required.

This chapter is about simulations to understand the path prediction performance for a PTW. Simulated are the motion of a PTW in virtual space and the signals that would be obtained by sensors, and it is examined how far ahead (how many seconds ahead) the calculated predicted path is valid. Oncoming vehicles must be able to distinguish whether the PTW is going straight, turning right, or turning left; only evaluating the straight PTW case is not sufficient.

3.2 Simulation Method

3.2.1 Simulation Tool

For this study, “BikeSim” provided by Mechanical Simulation Corporation (known as the developer of “CarSim”) was used to reproduce multi-body-dynamics of a two-wheeled vehicle. BikeSim takes into account realistic motorcycle geometry, mass properties, rider interaction, reactions of tires and suspensions, and road surface effects. This tool simulates the output signals of the sensors (IMU, steering sensor, etc.) installed in a PTW.

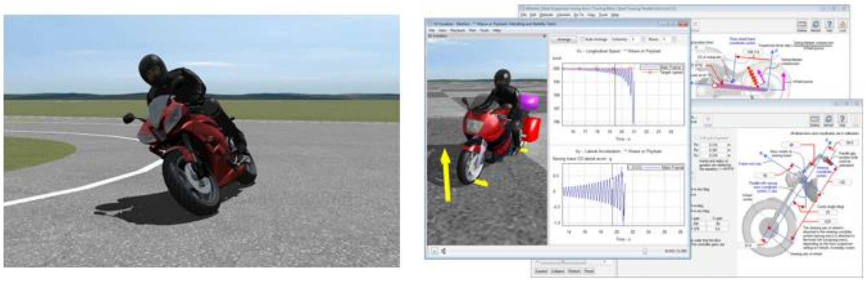


Figure 3.1: BikeSim

3.2.2 Test Patterns (Scenarios)

Twelve patterns of standard intersection passage scenarios were prepared. This intersection consists of a road running north-south, a road connected at a 90° angle from the west and a road at an angle of 45° from the east (Figure 3.2). Each scenario is a combination that passes from each direction (N-North, E-East, S-South, W-West) to each direction (left turn, straight, right turn). There are lane divisions on the north-south road.

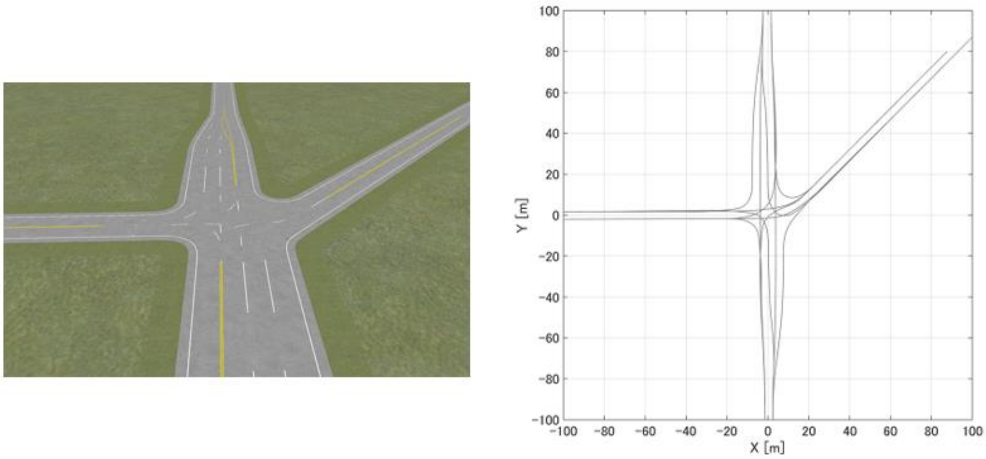


Figure 3.2: A standard intersection

The PTW approaches the intersection at a speed of 60 km/h from N, S and 40 km/h from E, W. The necessary deceleration for turning is also calculated.

Figure 3.3 shows an example of vehicle motion in a right turn scenario. The bottom speed is about 20 km/h because it needs to take a 90° angle corner.

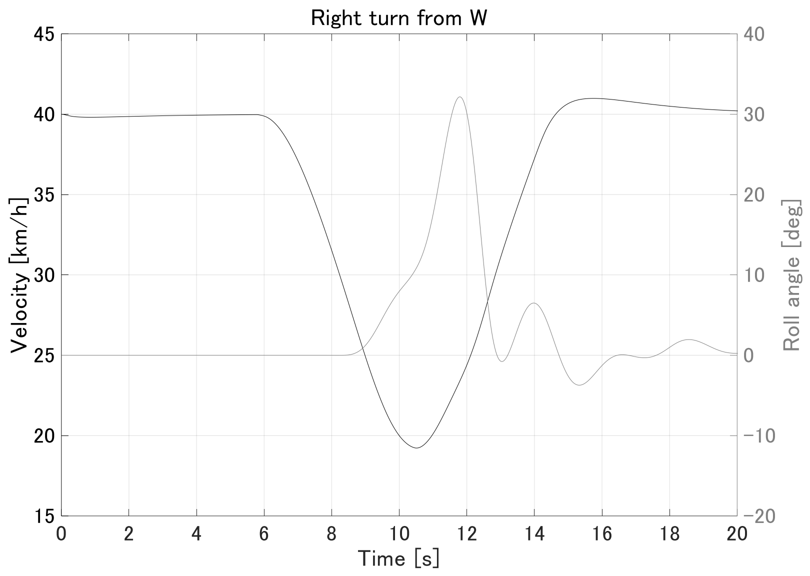


Figure 3.3: Time transition of velocity and roll angle (Right turn from W)

Figure 3.4 shows an example in a straight scenario. Therefore, the vehicle passes while maintaining speed. Due to the lane division on the north-south road, slight roll motion of less than 5° occurs even when going straight. This degree of roll motion will occur frequently due to occurring road conditions (puddles, pavement cracks, manhole covers), obstacles, disturbances and to keep distance from surrounding vehicles.

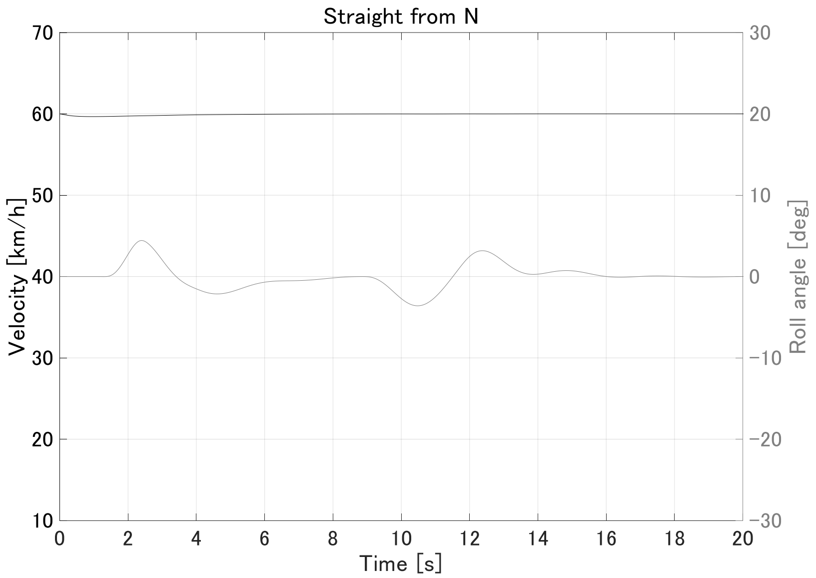


Figure 3.4: Time transition of velocity and roll angle (Straight from N)

All plots of the other patterns are presented in Chapter 3.5 below.

3.2.3 Algorithms

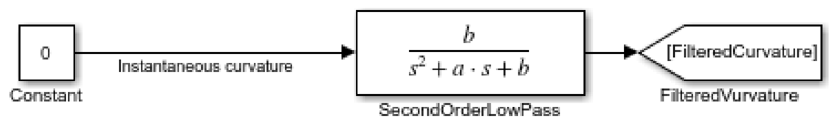
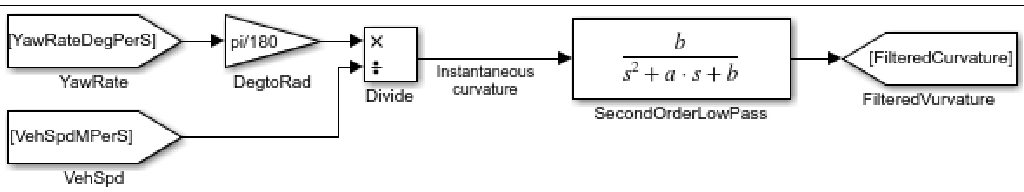
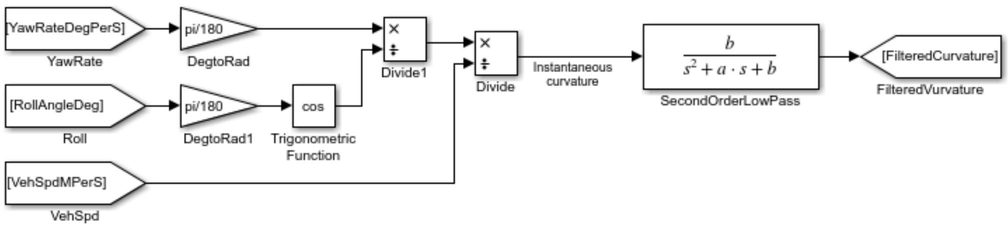
Five algorithms were tested in this study (Table 3.1). All algorithms require “Position” and “Heading angle (azimuth)”. These are already defined in the standard CAM specification as described in Chapter 5.1.

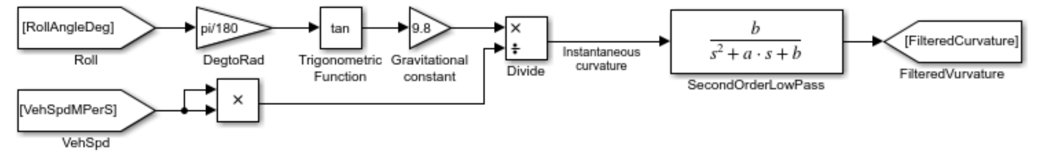
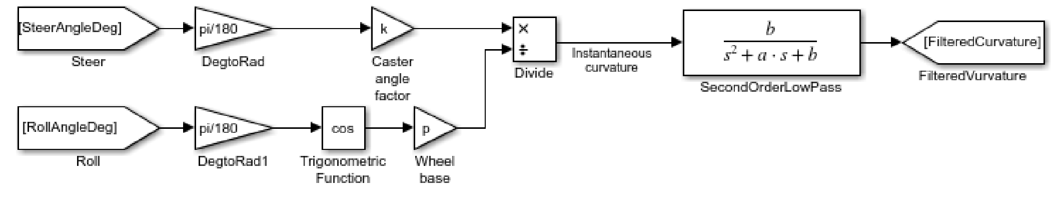
#0 is the "Level C" algorithm. This is as a reference to see the difference in performance between levels C and B. #1~4 are the "Level B" algorithms based on the curvature calculated with a combination of sensors expected to be installed in PTWs at the current industrial technology standards. #1 is calculated from the information mandatory in the standard CAM specifications, but # 2~4 will require additional sensors like an IMU or steering sensor.

Table 3.1: List of algorithms

#	Algorithm	Symbol colour	Level	Basic principle	Required sensors
0	Linear line		C	Linear interpolation of heading angle. The curvature is treated as zero.	Speed
1	SAE J2945/1		B	Calculate the curvature from the yaw rate information in the vehicle. This method is described as an example for four wheelers in the SAE J2945/1 appendix 6.	Speed Yaw rate
2	Modified SAE J2945/1		B	Calculate the curvature from the yaw rate which is corrected with the roll / tilt angle during PTW cornering.	Speed Yaw rate Roll angle
3	Centrifugal balance		B	Calculate the curvature that balances roll angle and velocity conditions.	Speed Roll angle
4	Kinematic steering angle		B	Calculate the curvature based on the steering angle projected onto the road surface.	Steering angle Roll angle

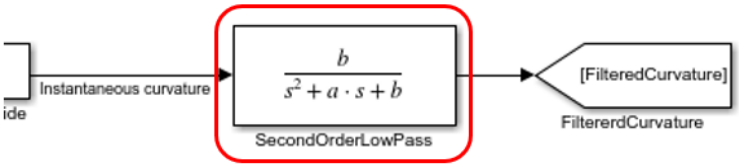
Table 3.2: Block diagrams

#	Algorithm	Calculation block diagram
0	Linear line	 The curvature is treated as zero.
1	SAE J2945/1	
2	Modified SAE J2945/1	

3	Centrifugal balance	
4	Kinematic steering angle	

3.2.4 Low Pass Filter

The SAE J2945/1 provides an example of how to calculate the curvature of a four-wheeled vehicle using yaw rate. In this case, a second order low pass filter is used, and the value of the cut off frequency is suggested to be 0.32-0.34 [Hz] (Figure 3.5). Following this, the same filter is used in this simulation as shown in Table 3.2. This is because it will be beneficial if the same path prediction (curvature calculation) method as four-wheeled vehicles can be used. The SAE filter is designed for steady state situations. However, this test scenario is at an intersection and requires a faster response, and the filter frequency seems to be too low for the test scenarios LTA and IMA. Different scenarios have different balance requirements for a trade-off between responsiveness and noise stability. In order to know the limitations with a faster filter, the result of a filter with the cut off frequency set to 1.0 [Hz] was also simulated. Figure 3.6 shows the difference between the results at 0.33 and 1.0 [Hz] as the cut off frequencies in the scenario of “Left turn from E”.



Calibration Parameter	Description	Default Value	Minimum Value	Maximum Value
Curvature Cutoff Frequency	Low-pass cutoff frequency for curvature filter.	0.33 Hz	0.32 Hz	0.34 Hz
Curvature Damping Factor	Curvature filter dampening factor.	1	0	2
Curvature Sampling Period	Sample time for discrete curvature filter.	100 ms	100 ms	400 ms

Figure 3.5: Low pass filter

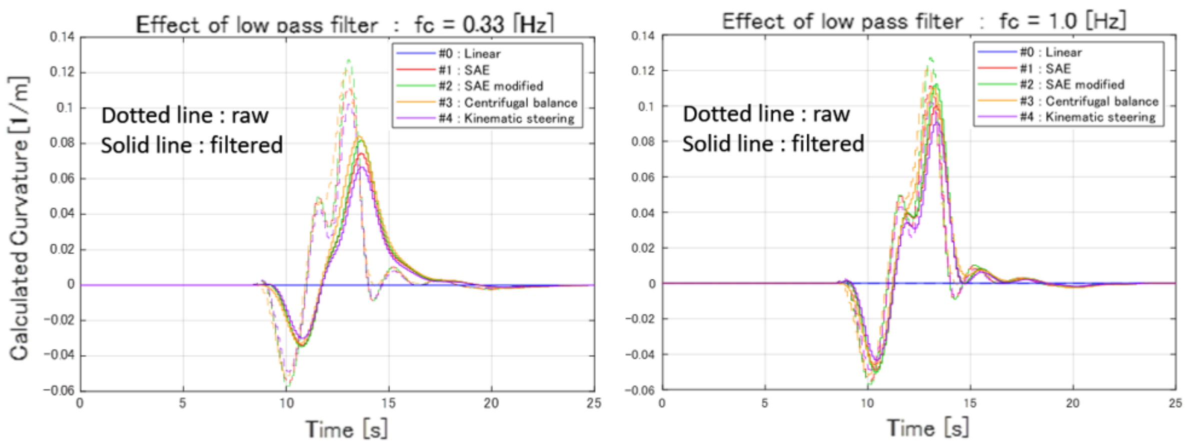


Figure 3.6: Effect of low pass filter (Left turn from E)

3.2.5 Evaluation Index

A quantification method is defined to evaluate the performance of the path prediction algorithm.

This is explained using Figure 3.7. It is evaluated whether the error distance between the predicted point and the actual point at a certain time in the future is smaller than the threshold (evaluation factor). In this case, the error distances are smaller than threshold at the time before t2. Therefore, the evaluation index of this case is defined as t2 [s]. This means that the path prediction performance of this algorithm is up to t2 [s] ahead.

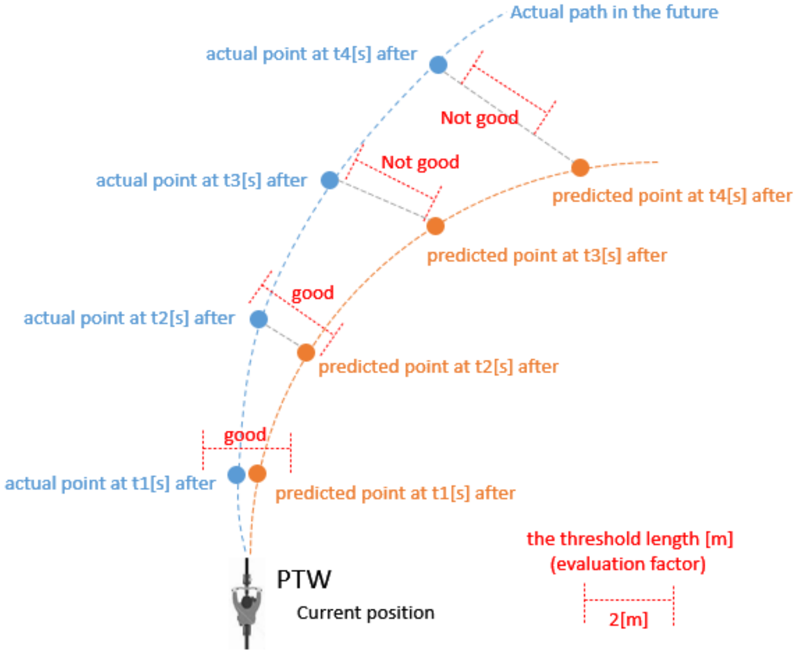


Figure 3.7: Performance evaluation of algorithms

With this evaluation method, the evaluation index depends on the situation and changes continuously. For example, in a steady situation (constant speed straight motion), the evaluation index will be a high value. On the other hand, in an unsteady situation (accelerating with changing direction) it will be a low value. In the same way, the evaluation index changes over time during an intersection scenario, e.g. when the scenario passes from a straight phase in a turn phase, as is shown in Figure 3.8.

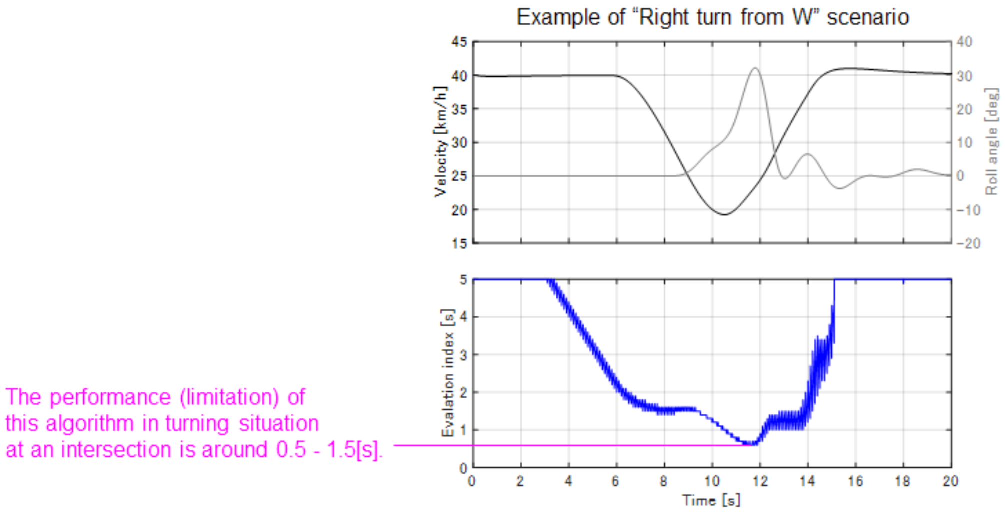


Figure 3.8: Example of evaluation index

The evaluation index also depends on the threshold length (evaluation factor). This threshold indicates a certain required spatial resolution. Increasing the threshold will improve the evaluation index (and vice versa), but it is equivalent to losing the spatial resolution.

In order to understand the sensitivity of the evaluation index by the threshold, a test was conducted in which the evaluation index was compared with several threshold values (Figure 3.9). Although the evaluation index slightly changes depending on the threshold, it is far below the TTC range assumed in Table 2.3. Considering the width of a typical car, the threshold was fixed to be 2.0 [m] in this study.

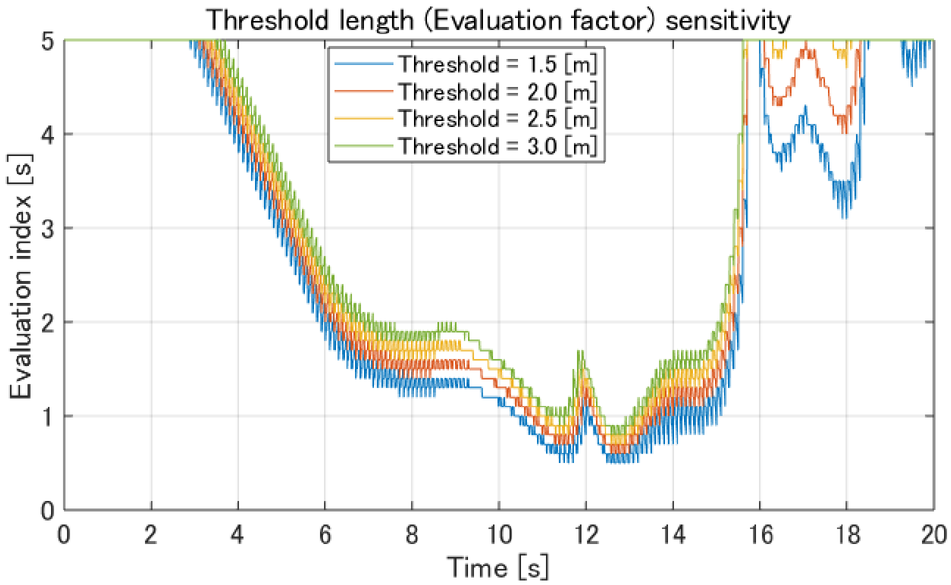


Figure 3.9: Sensitivity of evaluation index by threshold

3.3 Results

3.3.1 Exported data

The simulation results are available as Audio Video Interleave (AVI) video files (Figure 3.10). There are results of applying filters of 0.33 [Hz] and 1.0 [Hz], so 24 files in total.

LeftTurn_from_E_f033_k2.avi	Straight_from_E_f033_k2.avi	RightTurn_from_E_f033_k2.avi
LeftTurn_from_N_f033_k2.avi	Straight_from_N_f033_k2.avi	RightTurn_from_N_f033_k2.avi
LeftTurn_from_S_f033_k2.avi	Straight_from_S_f033_k2.avi	RightTurn_from_S_f033_k2.avi
LeftTurn_from_W_f033_k2.avi	Straight_from_W_f033_k2.avi	RightTurn_from_W_f033_k2.avi

LeftTurn_from_E_f10_k2.avi	Straight_from_E_f10_k2.avi	RightTurn_from_E_f10_k2.avi
LeftTurn_from_N_f10_k2.avi	Straight_from_N_f10_k2.avi	RightTurn_from_N_f10_k2.avi
LeftTurn_from_S_f10_k2.avi	Straight_from_S_f10_k2.avi	RightTurn_from_S_f10_k2.avi
LeftTurn_from_W_f10_k2.avi	Straight_from_W_f10_k2.avi	RightTurn_from_W_f10_k2.avi

Figure 3.10: List of AVI files

Figure 3.11 is an example of the screen of an AVI file.
In the left column and starting from the top, there are the plots of velocity & roll angle, curvature before and after applying the filter, and evaluation index. Each algorithm is drawn and compared in its own symbol colour.

The right-hand side column is a bird's-eye-view of the map. The black line represents the path that will actually pass in the 5 seconds future, and the line of each color is the result of the path prediction after 5 seconds estimated with each algorithm.

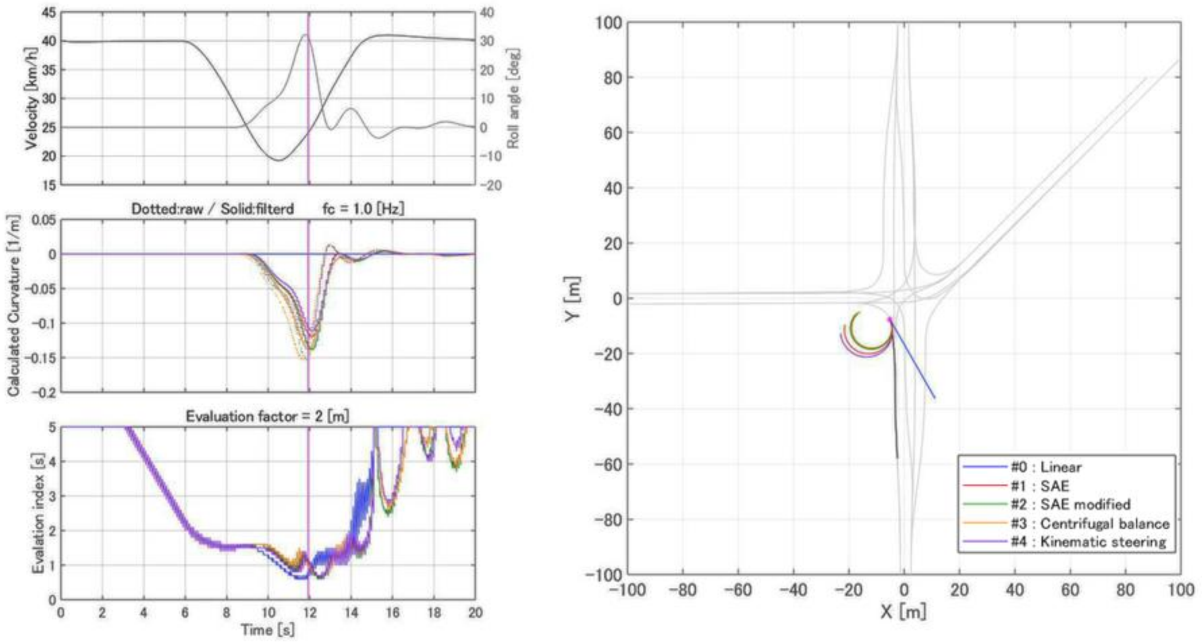


Figure 3.11: Example of simulation screen (Right turn from W)

3.3.2 Analysis

Figure 3.12 shows a bird's-eye-view of the map in a right turn scenario. And Figure 3.13 is the plot of the evaluation index of this scenario. As seen from Figure 3.13, there is no big difference between each algorithm, even the simplest #0 (linear line) method. In the turning situation at this standard intersection, a path prediction of only up to about 1 second ahead is the limit.

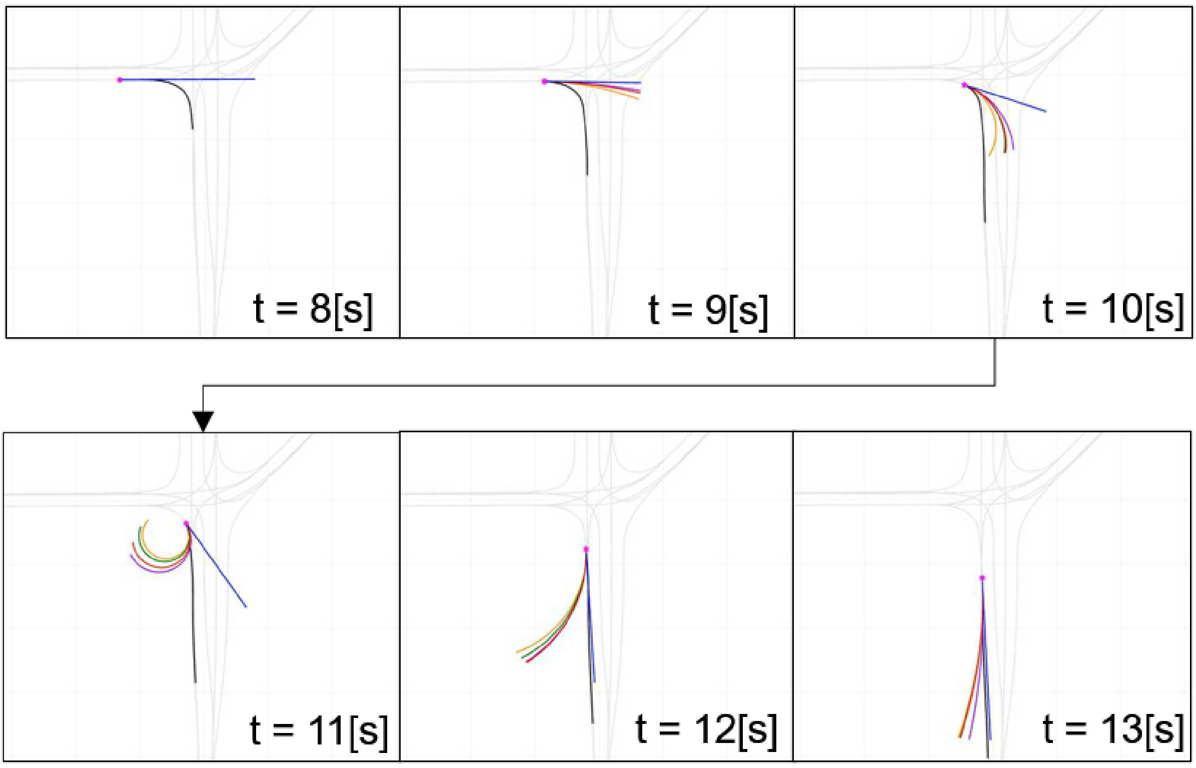


Figure 3.12: Bird's-eye-view of the map (Right turn from W)

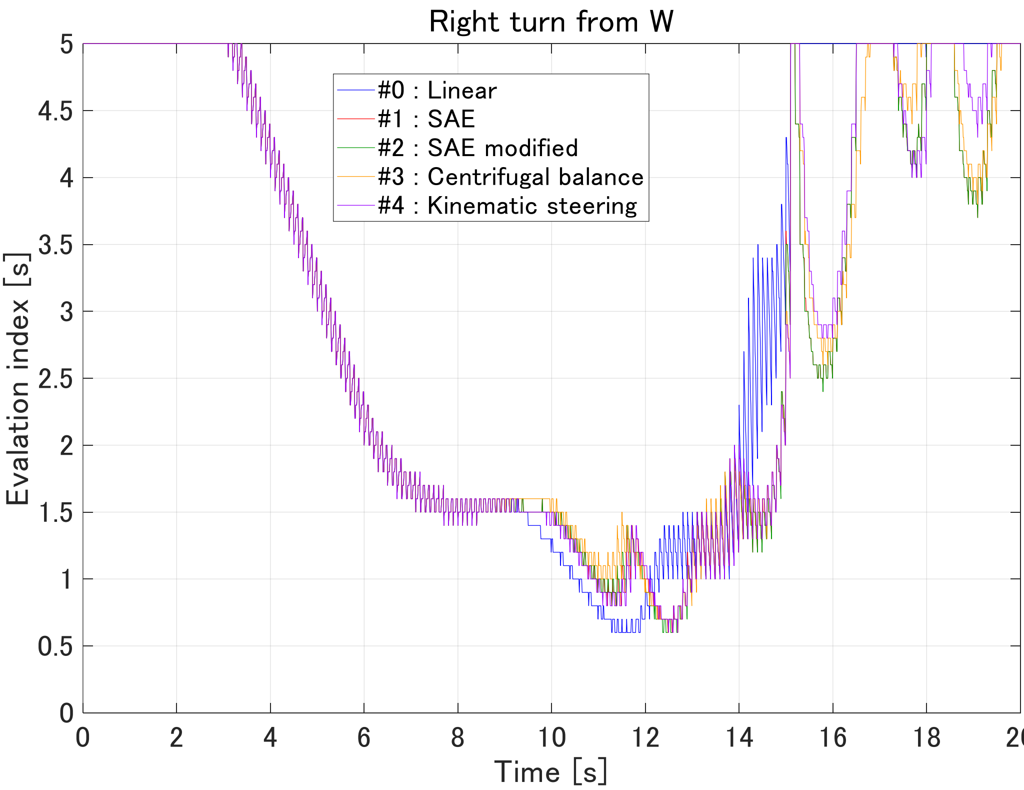


Figure 3.13: Evaluation result (Right turn from W)

Figure 3.14 and Figure 3.15 are the plots of a straight scenario. As already mentioned, a slight roll motion of less than 5° occurs when entering the central lane. Such behaviour is frequently caused by various disturbances in real traffic (e.g. avoiding manhole covers, puddles or cracks on the pavement). Even with this small degree of roll motion, the evaluation index is reduced to about 2-3 seconds. It is too sensitive to signals from additional sensors. The sensors cannot distinguish between turn motion and small (nonsensical) motion in straights. The level C algorithm can be said to be more primitive but stable. The level B algorithms are more advanced but unstable.

All plots of other patterns are presented in Chapter 3.5 below.

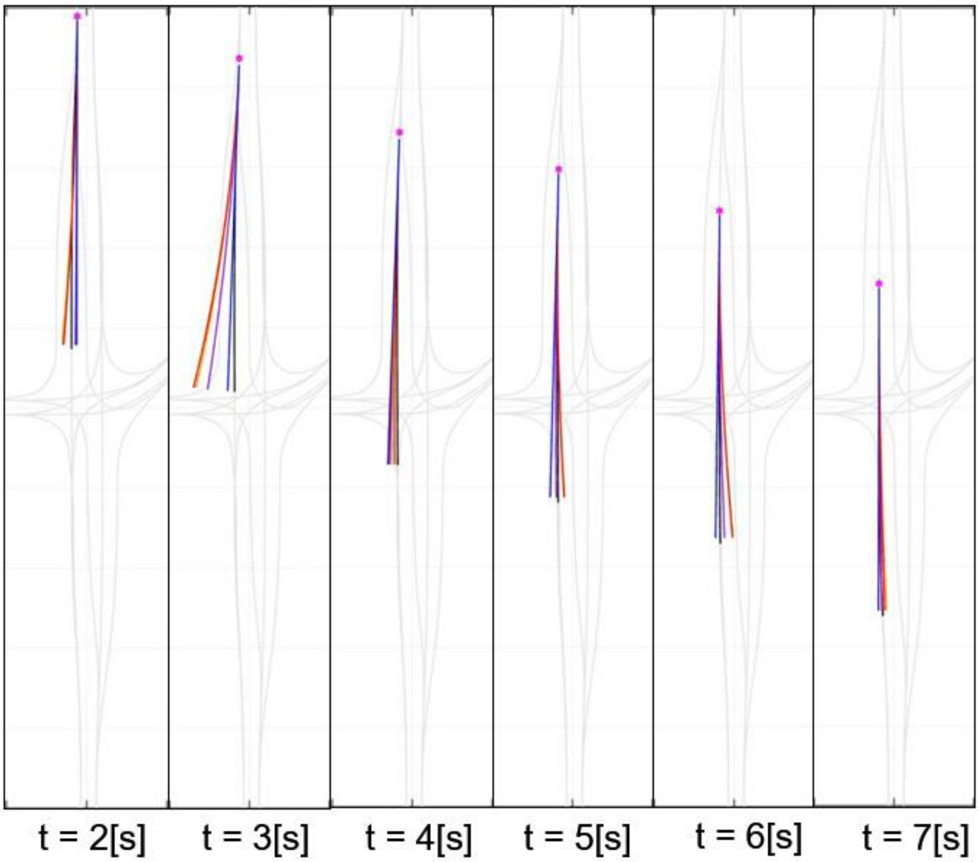


Figure 3.14: Bird's-eye-view of the map (Straight from N)

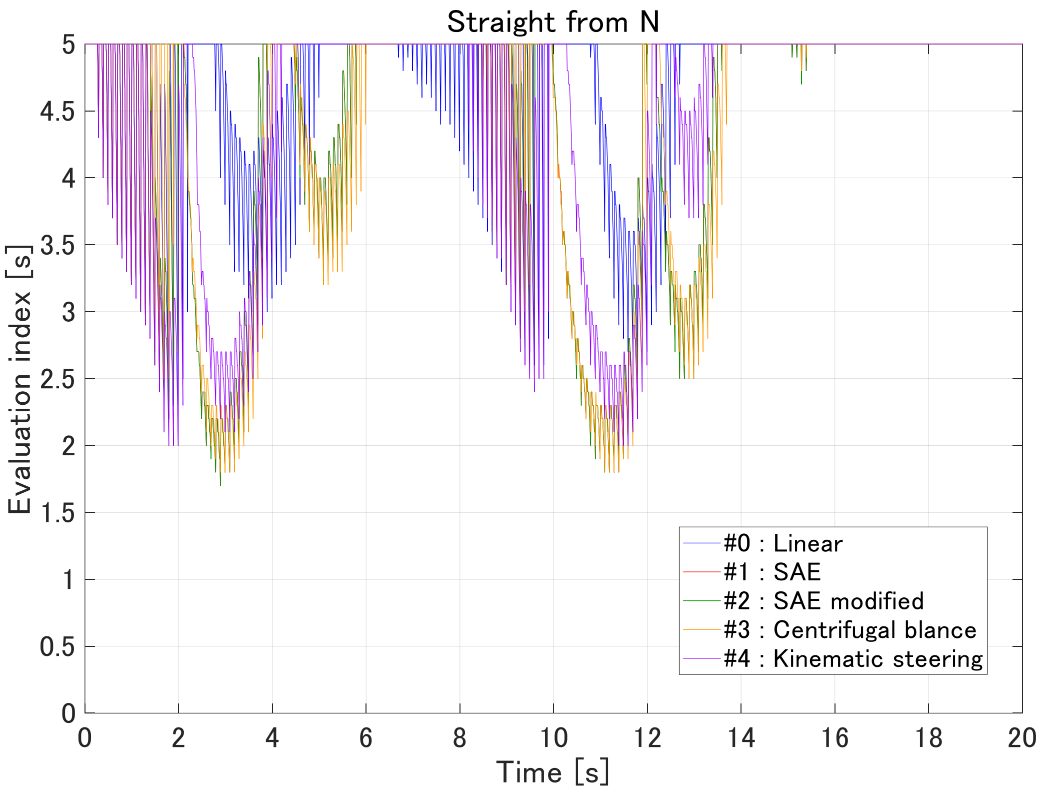


Figure 3.15: Evaluation result (Straight from N)

3.4 Considerations

As there was no large difference in results in terms of the evaluation indexes when algorithms with different combinations of sensors were used, and all of them showed a performance limitation of just 1-2 seconds, the above outcomes show that the algorithms need to be developed.
The oncoming vehicle must be able to know whether the PTW is going straight or turning a few (e.g. 2.5 – 6.5) seconds ahead of time, and currently this is not the case. The main factor of the performance limit is that it depends on the instantaneous curvature. In order to break through this performance limit, it is important how to incorporate future information.

As a method of predicting future information, the first thing that comes to mind is to calculate the amount of change (derivative) in the sensor value and reflect its trend of change. However, this method falls into the struggle against sensor noise. For PTWs, the sensor noise is more severe compared to cars, due to their inherently lighter weight and more exposed construction. Lightness causes vibrations and shocks from the engine and/or road surface to be transmitted without being absorbed by the mass of the frame and body. Exposure requires guarding the sensor, but the mass of a guard becomes a source of resonance and interferes with sensing. Although a PTW is going straight, the algorithm reacts to any slight fluctuation in the sensor and predicts a curvature, and at the next moment, it may predict in the opposite direction. Inserting a filter to suppress noise (even if it is a Kalman filter) is contrary to the purpose, and it is necessary to change the way of thinking.

The next improvement method we can think of is to read the rider’s intention. The state of the vehicle motion changes continuously, and even if a differential prediction is made, the situation will change completely in the next moment. On the other hand, the rider's intention of what manoeuvre to perform is likely to be very consistent (e.g. going straight, or turning), and therefore could be robust and effective information. It seems impossible even with the current state-of-the-art technology (brain wave detection, eye tracking), however, this information can be obtained easily. The most promising candidate is the turn indicator operation. In combination with lane accuracy in positioning, the algorithm will detect more smoothly if the rider intends to turn left / right or go straight. This will be possible by using a camera on the ITS station at an intersection and detect the lane that PTWs are in. Alternatively, it may be possible to implement it by giving the roadside station the role of a Real-Time Kinematic (RTK)-GNSS (D-GNSS) base station. The resolution for detecting the lane with conventional GNSS will not be enough, but it may be possible with RTK-GNSS.

Map information is also valid. Since PTWs move along the road, it is an effective method to limit the predicted position by map (possible trajectory). However, that leads to the question which device should process and update the map information. The hardware specs of the unit that will be installed in PTWs may be insufficient for map download and processing. From the idea that individual vehicles load and process maps, it may be useful to have the ITS-station that monitors an intersection collect information and report the location of possible collision points. This may lead to the idea of using the Local Dynamic Map (LDM) (described in Chapter 6.1.3).

As an idea for improving the algorithm, a method of multiplying the curvature after a few seconds by the forgetting coefficient can be considered instead of uniformly predicting the curvature. This is because the vehicle will not make constant turns on public roads. Determining the forgetting coefficient can be done by contextually interpreting the movement of the vehicle up to that time/point. This may be possible by using Recurrent Neural Networks (RNN).

3.5 Simulation results

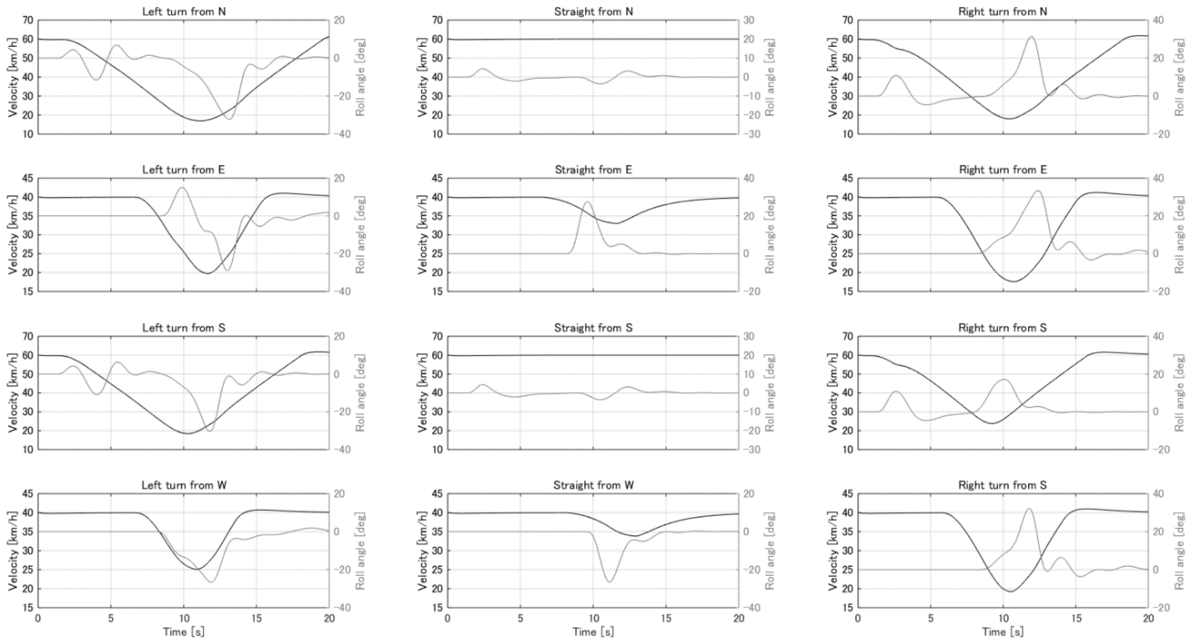


Figure 3.16: Time transition of velocity and roll angle (All patterns)

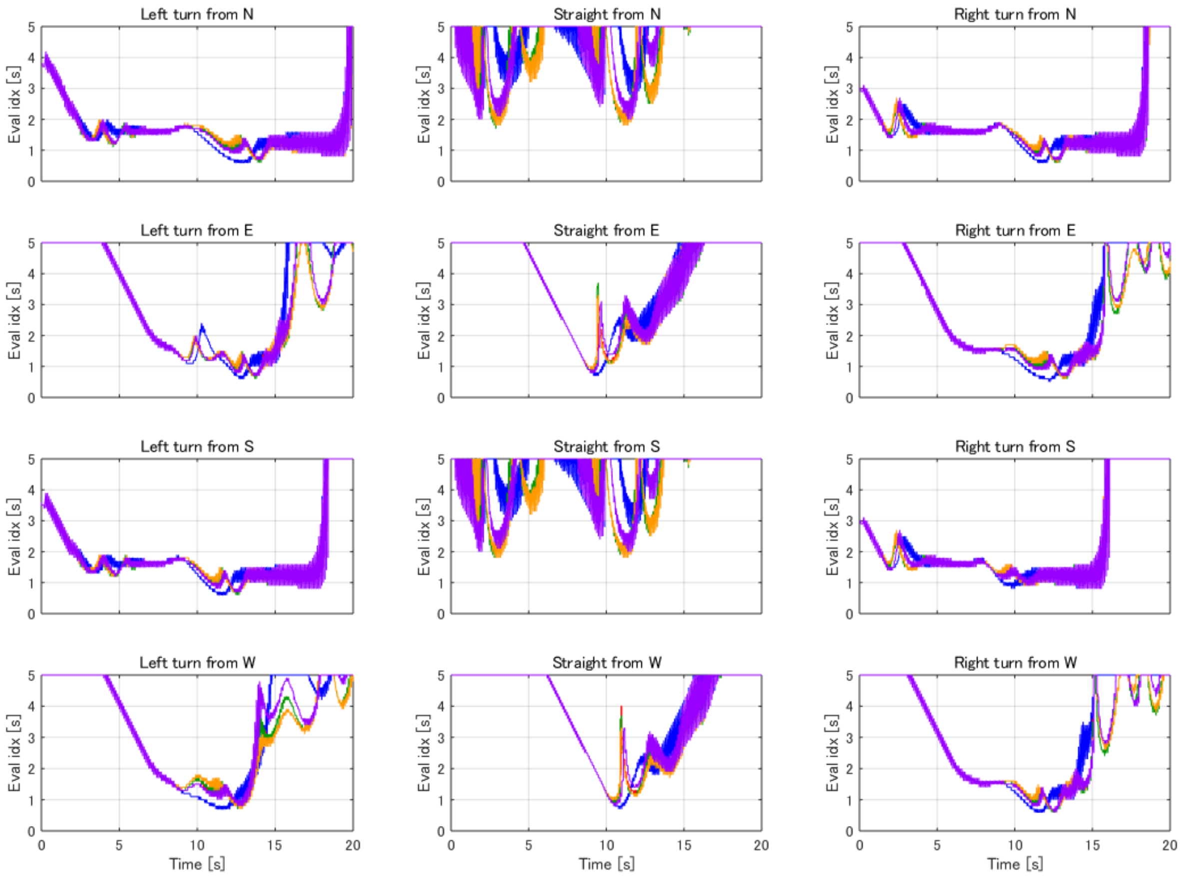


Figure 3.17: Evaluation indexes with 1.0[Hz] filter (All patterns)

4 Next steps in Path Prediction

Path prediction (or most commonly referred in the literature trajectory prediction) is mainly studied in development of ADAS-Systems The target in ADAS development is to predict other vehicles trajectories in order to avoid collisions.
Path prediction can be specified as the inference of future vehicle trajectory for a couple of seconds. By its definition, it is a regression problem for the longitudinal and lateral position of the vehicle relative to time. Machine learning methodologies as Hidden Markov models, dynamic Support Vector Machines, Gaussian Processes, Bayesian networks and Long-Short-Term Memory (LSTM) architectures are used for this purpose [4]. LSTM, which is a specialized type of Recurrent Neural Networks (RNN), is most commonly used the last years. RNN are neural networks used for the forecast of sequential data, in our case the trajectory of the motorcycle. The advantage of LSTM over the RNN is that they have the ability to “forget”, past positions of the trajectories that are irrelevant for the current prediction.

The problem definition for ADAS systems differs from the motorcycle path prediction in three ways:

1. The motorcycle lean angle or steering angle can be a known input while no such information is available in the ADAS systems
2. The motorcycle lateral dynamics are quite different and more complex than the car dynamics.
3. The road centerline position in the case of motorcycle path prediction is not considered known while it is known in some of ADAS articles in the literature.

The training of the LSTM network can be performed with datasets containing vehicle trajectory data of motorcycles. These trajectories could be available from datasets such as the nuScenes dataset [5] that contains recordings from sensors onboard vehicles, or from datasets with recordings from sensors on drones or traffic cameras as the datasets from NGSIM (“A Model Endeavor | [6] and HighD [7] or even from instrumented motorcycles or motorcycle vehicle dynamics simulation software.

The path prediction problem can also be specified as a classification problem for the determination of the vehicle maneuver. The motorcycle maneuver can be classified in the following lateral and longitudinal kinematics classes: keep lane, change lane left/right, turn left/right on an intersection, and normal driving and braking respectively. The input for this classification can be the motorcycle controls and previous kinematics as well as the position of the motorcycle relative to the road lanes. In [7] an Artificial Neural Network was trained with input vehicle kinematics parameters, as the yaw angle relative to the road centerline, yaw rate and velocity. In [8] the relative distance of the vehicle from the centerline of the lane was used instead. The trajectory prediction and the maneuver prediction can be used together to improve model accuracy with a Kalman filter. While another approach is to use the maneuver prediction as an additional input for the trajectory prediction [7].

The prediction of the trajectory can be represented in an Occupancy Grid Map (OGM) with probabilities for the future position of the motorcycle. The OGM is extensively used in robotics for path planning. The OGM can be used to formalize the communication of the motorcycle position to the other vehicles by communicating the different probabilities for the motorcycle to take certain positions at specific times [9], [10], [11].

5 Summary

Path prediction can be specified as the inference of future vehicle trajectory for a couple of seconds. Path prediction for PTWs is quite a challenge, given the complex vehicle dynamics which differ fundamentally from cars.

CMC has investigated the level of path prediction based on two levels: level B (by means of radius of curvature calculation based on the instantaneous state of the vehicle, as measured with a combination of vehicle sensors) and level C (based on information provided by CAMs, such as ‘referencePosition’, ‘heading’ and ‘speed’).
Level A (more advanced and complex methods such as contextual behaviour predictions using machine learning) was seen as exceeding currently available technology standards and therefore not tested within the scope of this whitepaper.

A left turn situation was taken as scenario; in both cases the path predication and collision risk can be calculated based on a ‘Ghost Vehicle’ method.
In this whitepaper, CMC examined how far ahead (how many seconds ahead) the calculated predicted path is valid, within a threshold of 2 meters.

Various intersection passage scenarios were investigated with a BikeSim tool and five different algorithms were tested.
In most cases, there were no big differences between the algorithms. During an actual manoeuvre, a path predication of only up to about 1-2 seconds ahead was the limit of these systems. On straights, the path predication can be valid for more seconds, but as soon as there is a small degree of roll motion (which is natural for PTWs, even on straights) this is already reduced to 2-3 seconds.

The main factor behind the performance limit is that it depends fully on the instantaneous curvature. In order to improve that, consideration could be given to items like rider intention, map information, and next algorithm improvements; in the future also machine learning / artificial neural networks.

For the moment, however, this means that an oncoming vehicle driver cannot be informed early enough whether the PTW is going straight or turning and path prediction for PTWs is not sufficiently solved.

6 Appendix

6.1 Available Parameters

6.1.1 Introduction

Most V2X (Vehicle-to-Everything) applications rely on broadcasting awareness messages known as CAM in ETSI standards. A CAM is sent only when a set of rules is met so that the resulting CAM send frequency can be 1 to 10 Hz. These ETSI rules specify when vehicles, (or in general ITS stations) should generate CAMs, and what should be their content [2]. These messages are independent of the wireless technologies (e.g., ITS-G5, LTE-V2X or 5G V2X) and include the position, the speed and other basic status information of the transmitting node, as we will see in more details in the following chapter. The received CAMs are aggregated in a so-called Local Dynamic Map (LDM). From this database, information about the traffic situation can be extracted. Moreover a extra container with additional PTW specific information for motorcycles, according to ETSI TS 103 300-3 could be integrated into the CAM.
The goal of this chapter is to highlight all the available parameters included in these messages in order to have a clear idea regarding what can be used to perform different levels (C, B, A) of path prediction.

6.1.2 Cooperative Awareness Message (CAM)

The Cooperative Awareness (CA) basic service is a facility layer that operates the CAM protocol. It provides two services: sending and receiving of CAMs.
As specified in [2], CAMs are generated periodically with a frequency controlled by the CA basic service in the originating ITS-S. The generation frequency is determined taking into account the change of own ITS-Ss status, e.g. change of position or speed as well as the radio channel load. In particular, a vehicle should generate a new CAM if any of the following triggering conditions is satisfied:

- The distance between the current position of the vehicle and the position included in its previous CAM exceeds 4 m.
- The absolute difference between the current speed of the vehicle and the speed included in its previous CAM exceeds 0.5 m/s.
- The absolute difference between the current heading of the vehicle and the heading included in its previous CAM exceeds 4°.
- The time elapsed since the last CAM was generated is equal to or higher than 1 s. [12]

In the case of ITS-G5, the minimum time interval between two consecutive CAM is provided by the Decentralized Congestion Control (DCC) cross-layer, in order to reduce the CAM generation according to the channel usage requirements [13].
In the case of LTE-V2X, DCC is not applicable, since channel congestion control is managed by the access layer defined in [14].
A CAM message includes one ITS Protocol Data Unit (PDU) header and multiple mandatory or optional containers [2]. The header includes Data Elements (DE) such as the protocol version, the message type and the ID of the vehicle or Road-Side Unit (RSU) that transmits the CAM. Each container includes a series of optional and mandatory DEs:

- The basic container is mandatory and includes information of the transmitting vehicle (e.g., the type of vehicle or its position).
- The high frequency container is mandatory in every CAM and contains highly dynamic information of the transmitting vehicle (e.g., its acceleration, heading or speed).
- The low frequency container is optional (i.e., not included in every CAM, it is included with a lower frequency) and contains static and dynamic information of the transmitting vehicle (e.g., the status of the exterior lights and the vehicle's path history).
- The special vehicle container is optional and is transmitted by specific vehicles such as motorcycles, public transport, emergency vehicles, or vehicles transporting dangerous goods.

The general structure of a CAM is illustrated in Figure 6.1:

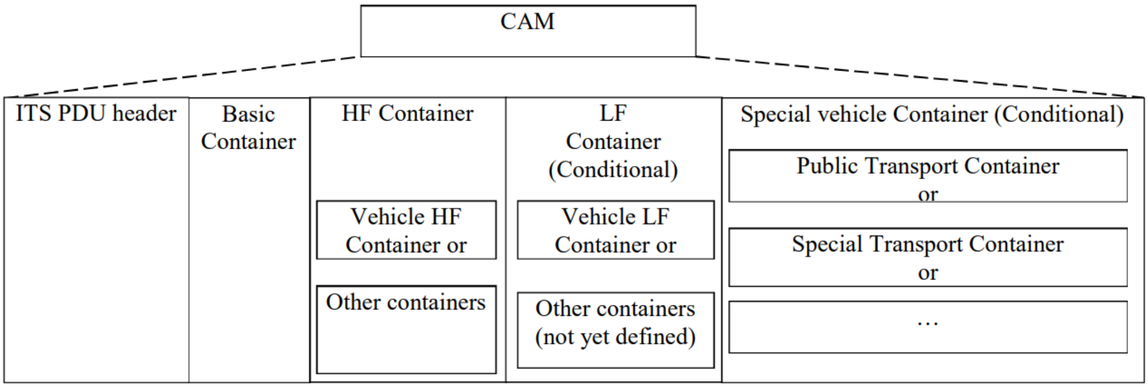


Figure 6.1: Cooperative Awareness Message structure

As well explained in [12], the size of CAMs depends on the optional containers and the DEs included.
The ITS PDU header and the basic container are mandatory and have a fixed size.

6.1.2.1 Basic Container

- stationType
- referencePosition
  - latitude
  - longitude
  - positionConfidenceEllipse
  - altitude

Parameters of basic container have to be explained (descriptions can be found in Annex A of [7])

The high frequency container is mandatory. However, 7 of its 16 DEs are optional. The size of this container is hence variable and depends on the manufacturer and the context conditions of the vehicle [15].

6.1.2.2 High Frequency Container

- Basic Vehicle Container HF
    - Heading
    - Speed
    - driveDirection
    - vehicleLength
    - vehicleWidth
    - longitudinalAcceleration
    - curvature
    - yawRate

Parameters of high frequency container have to be explained (descriptions can be found in Annex A of [16])

The low frequency container is optional and is transmitted less frequently than the high frequency container. It has three mandatory DEs including the PathHistory. This DE describes the path that a vehicle has followed. PathHistory is not fixed and uses between 0 and 40 path entries, of. The number of path entries depends on the driving conditions and the implementation [16]. For example, the Car-to-Car Communication Consortium (C2C-CC) profile 1.3 [15] states PathHistory should cover 200-500 meters. However, other implementations like SCOOP release 1.2 [15] propose up to 40 points.

6.1.2.3 Low Frequency Container

- vehicleRole
- exteriorLights
- pathHistory
    - pathPoint
        - pathPosition
            - deltaLatitude
            - deltaLongitude
            - deltaAltitude
        - pathDeltaTime

Parameters of low frequency container have to be explained (descriptions can be found in Annex A of [16])

Before talking about the special vehicle container, it is worth mentioning the position of ETSI with respect to Vulnerable Road User Awareness Message (VAM) and motorcycles. In particular, [17] specifies that VRU profile 3 (motorcycles) should not transmit the VAM, as it includes data which is already transmitted in the CAM. Therefore, Annex D in [17] recommends the inclusion of a special vehicle container into the CAM. Indeed, in case of a motorcycle, the VRU basic service needs to inform the Cooperative Awareness basic service that the vehicle is a VRU profile 3 and trigger the dedicated container when transmitting CAMs. It also needs to provide associated DEs to put in the VRU special container.

Table 6.1 below provides the list of the data to be included in the specialVehicleContainer.MotorcyclistContainer proposed to be added to the CAM standard, as specified in [17].

Table 6.1: Proposed list of data to be included to the CAM

Description	Data
Motorcyclist special vehicle container [this container should be present if the DE vehicleRole, contained inside the low frequency container, is set to VRUMotorcyclist[16]	• vruSubProfileMotorcyclist (mandatory) • vruSizeClass (mandatory) • rollAngle (optional) • vruOrientation (optional) • vruSafeDistance (optional) • pathPrediction (optional) • stabilityChangeIndication (optional)

Each data is described in the Annex B of [17].

vruSubProfileMotorcyclist = The sub-profile of the ITS-S that originates the message.

vruSizeClass = It contains information of the size of the motorcycle. The size class is interpreted in combination with the profile type to get the range of dimensions of the VRU.

Profile	Profile Value	VruSizeClass Value	VruSizeClass description
Motorcyclist	3	0	Unavailable
		1	Low → 1,5 m or less height
		2	medium → more than 1,5 m in height, 1 m or less front-to-back
		3	high → more than 1,5 m in height, more than 1 m front-to-back

rollAngle = The Data Frame (DF) rollAngle provides the angle and angle accuracy between the ground plane and the current orientation of a vehicle's y-axis with respect to the ground plane about the x-axis according to the ISO 8855 [9]. The DF shall include the following information:

- rollAngleValue
- rollAngleConfidence

vruOrientation = The VruOrientation DF complements the dimensions of the vehicle by defining the angle between the vehicle longitudinal axis with regards to the WGS84 north.

vruSafeDistance = This DF provides an indication of whether the VRU is at a recommended safe distance laterally, longitudinally and vertically from up to 8 other stations in its vicinity. The simultaneous comparisons between Lateral Distance (LaD), Longitudinal Distance (LoD) and Vertical Distance (VD) and their respective thresholds, Minimum Safe Lateral Distance (MSLaD), Minimum Safe Longitudinal Distance (MSLoD), and Minimum Safe Vertical Distance (MSVD) as defined in [18], shall be used for setting the VruSafeDistanceIndication DF. Other ITS-S involved are indicated as StationID DE within the VruSafeDistanceIndication DE.

pathPrediction = This DF provides the set of predicted locations of the ITS-S, confidence values and the corresponding future time instants. It is a sequence of VruPathPoint (up to 40 predicted path points).

A way to predict future points based on "learned paths" is given in **Annex G.3 VRU Behaviour learning aspects** in [18].

stabilityChangeIndication = It contains ego-vehicle's change in stability for a time period. The DE StabilityLossProbability shall give the probability indication of the stability loss of the ego-vehicle. The DE ActionDeltaTime shall indicate the time duration.

6.1.3 Local Dynamic MAP (LDM)

The LDM is a technology incorporated in C-ITS with the purpose of registering spatio-temporal data of static and dynamic traffic-related information on a detailed geographic map. LDM is standardised from ETSI for managing the vehicle sensor and map data. The LDM functions within roadside ITS stations or inside Traffic Management Centres (TMC) and is making the information available to the vehicle / driver through V2X communication. With the advancement of embedded systems, the LDM will be also functioning onboard self-driving vehicles [19]. LDMs are implemented with the use of a database while semantically enriched LDMs with spatial stream databases are in development. These enhanced LDMs provide in-depth and easier to access information, in real-time after the data is created, with the query capabilities of the databases.

The LDM contains real-world data from real objects (vehicles, road events, etc.) registered on a highly accurate geographic map. The ITS reference architecture according to ETSI specifies two main components of the LDM. The LDM Management and the Data Store. The LDM Management receives incoming information such as CAMs and Decentralized Environmental Notification Messages (DENMs) and performs plausibility checks. The Data Store is used to store the validated data.

LDM has a mechanism of dividing up the data in types based on its time of validity. Real-world data can be categorized into four types [19] (Figure 6.2, [20])

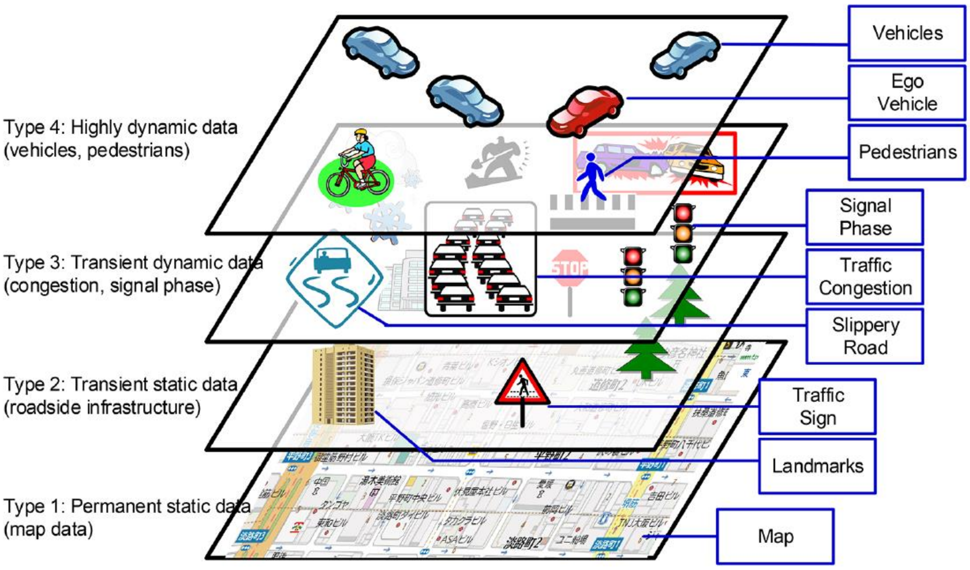


Figure 6.2: Four types of the LDM [22; p.103]

Type 1

The first type, the static one, is usually provided by a map data supplier, which includes e.g., road topography, road attributes and points of interests.

Type 2

Including transient static data e.g., information about roadside infrastructure such as position of gantries and traffic signs.

Type 3

The 3rd type contains transient dynamic data like information about road works such as position, lane width, speed limits etc.

Type 4

The last type consists of highly dynamic data provided in defined time interval and manages all short-term changes in the traffic situation, in the range of a few seconds. This includes information on traffic signal phases, traffic density, speed levels and the positions of rescue and emergency vehicles. In addition, vehicle data is sent from the CAM with information about the vehicle type, position and state. From these, information such as position, speed, direction, ABS (Anti-Lock Braking System), ESP (Electronic Stability Program), crash, etc. can be read out. The combination of these data enables a good estimation of the current road situation and makes it possible to warn affected participants or to generate predictions for the following traffic.

For an LDM it is not necessary to use Type 1 from the real-world data.

Some of the applications of LDM concern the following:

- Congestion warning
- Traffic jam reduction / avoidance
- Hazard alerts (e.g. emergency braking of the vehicle ahead)
- Reduction of the probability of accidents
- Dynamic traffic guidance (e.g. adapting traffic lights with the traffic situation)

6.1.4 References

[1] = CMC, 2022.10., Rider Reaction Time - Connected Motorcycle Consortium (cmc-info.net)

[2] = ETSI EN 302 637-2 V1.4.1, Jan. 2019, "Intelligent Transport Systems; Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service,"

[3] = SAE International, J2945/1_202004: On-Board System Requirements for V2V Safety Communications

[4] = Altché, F., de La Fortelle, A., 2017. An LSTM network for highway trajectory prediction, in: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). pp. 353–359. https://doi.org/10.1109/ITSC.2017.8317913

[5] = Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O., 2020. nuScenes: A Multimodal Dataset for Autonomous Driving, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Seattle, WA, USA, pp. 11618–11628. https://doi.org/10.1109/CVPR42600.2020.01164

[6] = A Model Endeavor | FHWA [WWW Document], n.d. URL https://highways.dot.gov/public-roads/janfeb-2007/model-endeavor (accessed 7.29.22).

[7] = Benterki, A., Boukhnifer, M., Judalet, V., Maaoui, C., 2020. Artificial Intelligence for Vehicle Behavior Anticipation: Hybrid Approach Based on Maneuver Classification and Trajectory Prediction. IEEE Access 8, 56992–57002. https://doi.org/10.1109/ACCESS.2020.2982170

[8] = Houenou, A., Bonnifait, P., Cherfaoui, V., Yao, W., 2013. Vehicle trajectory prediction based on motion model and maneuver recognition, in: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 4363–4369. https://doi.org/10.1109/IROS.2013.6696982

[9] = Khakzar, M., Rakotonirainy, A., Bond, A., Dehkordi, S.G., 2020. A Dual Learning Model for Vehicle Trajectory Prediction. IEEE Access 8, 21897–21908. https://doi.org/10.1109/ACCESS.2020.2968618

[10] = Kim, B., Kang, C.M., Kim, J., Lee, S.H., Chung, C.C., Choi, J.W., 2017. Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network, in: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). pp. 399–404. https://doi.org/10.1109/ITSC.2017.8317943

[11] = Deo, N., Trivedi, M.M., 2018. Multi-Modal Trajectory Prediction of Surrounding Vehicles with Maneuver based LSTMs, in: 2018 IEEE Intelligent Vehicles Symposium (IV). pp. 1179–1184. https://doi.org/10.1109/IVS.2018.8500493

[12] = R. Molina-Masegosa, M. Sepulcre, J. Gozalvez, F. Berens and V. Martinez, "Empirical Models for the Realistic Generation of Cooperative Awareness Messages in Vehicular Networks," in IEEE Transactions on Vehicular Technology (Early Access). Date of Publication: 09 March 2020. DOI: 10.1109/TVT.2020.2979232 https://ieeexplore.ieee.org/document/9027961

[13] = ETSI TS 102 724 (V1.1.1): "Intelligent Transport Systems (ITS); Harmonized Channel Specifications for Intelligent Transport Systems operating in the 5 GHz frequency band".

[14] = ETSI TS 103 574 (V1.1.1): "Intelligent Transport Systems (ITS); Congestion Control Mechanisms for C-V2X PC5 interface; Access layer part".

[15] = CAR 2 CAR Communication Consortium, "Survey on ITS-G5 CAM statistics," TR2052, V1.0.1, Dec. 2018.

[16] = ETSI TS 102 894-2 V1.3.1 (2018-08): "Intelligent Transport Systems (ITS); Users and applications requirements; Part 2: Applications and facilities layer common data dictionary".

[17] = ETSI TS 103 300-3 V2.1.1 (2020-11): "Intelligent Transport Systems (ITS); Vulnerable Road Users (VRU) awareness; Part 3: Specification of VRU awareness basic service; Release 2".
[18] = ETSI TS 103 300-2 V2.1.1 (2020-11): "Intelligent Transport System (ITS); Vulnerable Road Users (VRU) awareness; Part 2: Functional Architecture and Requirements definition; Release 2";
[19] = ETSI EN 302 895 V1.1.1 (2014-09): "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM)".
[20] = Shimada, Hideki & Yamaguchi, Akihiro & Takada, Hiroaki & Sato, Kenya. (2015). Implementation and Evaluation of Local Dynamic Map in Safety Driving Systems. Journal of Transportation Technologies. 05. 102-112. 10.4236/jtts.2015.52010, link: https://www.researchgate.net/publication/276499922_Implementation_and_Evaluation_of_Local_Dynamic_Map_in_Safety_Driving_Systems.

7 List of abbreviations / Glossary

ABS	Anti-Lock Braking System
ADAS	Advanced Driver Assistance Systems
C2C-CC	Car-to-Car Communication Consortium
CA	Cooperative Awareness
CAM	Cooperative Awareness Message
C-ITS	Cooperative Intelligent Transport Systems
CMC	Connected Motorcycle Consortium
DCC	Decentralized Congestion Control
DE	Data Elements
DENM	Decentralized Environmental Notification Message
DF	Data Frame
ESP	Electronic Stability Program
ETSI	European Telecommunications Standards Institute
GIDAS	German In-Depth Accident Study
GNSS	Global Navigation Satellite System
HMI	Human-Machine Interface
HUD	Head-Up Display
IMU	Inertial Measurement Unit
ITS	Intelligent Transport System
LaD	Lateral Distance
LDM	Local Dynamic Map
LoD	Longitudinal Distance
LSTM	Long-Short-Term Memory
LTA	Left Turn Assist
MSLaD	Minimum Safe Lateral Distance
MSLoD	Minimum Safe Longitudinal Distance
MSVD	Minimum Safe Vertical Distance
OGM	Occupancy Grid Map
PDU	Protocol Data Unit
PTW	Powered Two-Wheeler

RNN	Recurrent Neural Network
RSU	Road-Side Unit
RTK	Real-Time Kinematic
TMC	Traffic Management Centres
V2X	Vehicle-to-Everything
VAM	Vulnerable Road User Awareness Message
VD	Vertical Distance