Safe and Efficient Electrical Vehicle

7th Framework programme / FP7-2010-ICT-GC /
ICT for the fully electrical vehicle / Grant Agreement Number 258133

D620.61 Virtual Co-Pilot

Authors

Name	Company
GLASER, Sébastien	IFSTTAR
ORFILA, Olivier	IFSTTAR

Revision chart

Version	Date	Reason
0.1	30.06.2011	Basis document
0.2	26.08.2011	Document Ready for Peer Review
1.0	30.09.2011	Document Reviewed

Executive Summary

This deliverable addresses the definition of the algorithm of the co-pilot subsystem. However, the task "620 - Co-Pilot" ends at T0+24. Then, this document presents the first achievement for the co-pilot. It will be updated at the end of the task with final version of the algorithm.

The aim of this document is to provide the other partners with the description and performance of the algorithm for the co-pilot subsystems. It will allow the test, evaluation and integration of the functions in the simulator and the micro controller. The document is organized in three sections. First section presents the general methodology for the algorithm development. Second section specifically develops the longitudinal ADAS and third section handles the lateral ADAS. The description of the function does come from the deliverable D220_25.

Table of Content

Introduction

This deliverable addresses the definition of the algorithm of the co-pilot subsystem. However, the task “620 - Co-Pilot” ends at T0+24. Then, this document presents the first achievement for the co-pilot. It will be updated at the end of the task with final version of the algorithm.

The aim of this document is to provide the other partners with the description and performance of the algorithm for the co-pilot subsystems. It will allow the test, evaluation and integration of the functions in the simulator and the micro controller. The document is organized in three sections. First section presents the general methodology for the algorithm development. Second section specifically develops the longitudinal ADAS and third section handles the lateral ADAS. The description of the function does come from the deliverable D220_25.

1 Description of the methodology

This deliverable is the first version of the co-pilot module. In the global architecture, the co-pilot aims at delivering to the Decision Unit 1 (DU1) module the motion vectors that are defined by various driver assistance using the inputs from the driver (driver desired speed, driver constraints...) and from the data fusion (lane position, obstacle position, vehicle speed...).

The objective is to bring the full electric vehicle developed in this project as close to the existing conventional car in term of driving assistance. Then, most of the driving assistances presented in this deliverable are already available. However, we bring some innovation by integrating the energy management directly in the definition of the motion vector at the co-pilot level, especially for the longitudinal driving assistance, with the Smart And Green Acc (SAGA).

In the following, we have separated the longitudinal driving assistance and the lateral driving assistance, as they do not address the same output (speed and acceleration for the longitudinal driving assistance, trajectory curvature or steering angle for the lateral driving assistance). For each mode, we start by presenting the different variables that are common for each driving assistance in this mode; next we present each driving assistance.

The driving assistances are presented using the following sections:

- High level description: this section reminds the inputs and outputs of the function, as stated in the D210_23 deliverable.
- Low level description: this section presents the architecture of the function itself and the various subfunctions.
- Algorithm description: this section explains the behavior of each sub functions
- Validation: this last section presents the behavior of the function on selected scenario and compares the impact of the driving assistance, if it is possible.

2 Longitudinal ADAS

The family of longitudinal ADAS is composed of several systems. However, each subsystem relies on the data presented on the following figure.

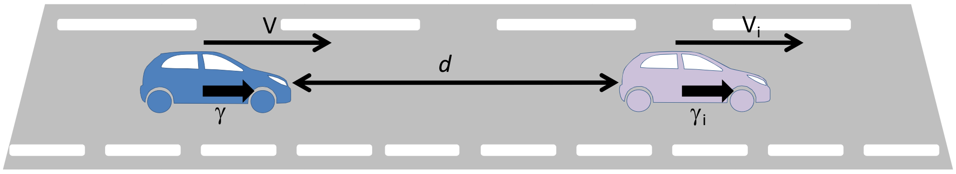

Figure 1: Notations for the *ACC* function

Table 1: Vehicle dynamic related variables for the *ACC* function

Variable	Type	Units
d	Distance to the front vehicle	m
V	Speed of the ego vehicle	m/s
γ	Acceleration of the ego vehicle	m/s^2
T	Time headway, expressed as $T=d/V$	s
V_i	Speed of the front vehicle	m/s
γ_i	Acceleration of the front vehicle	m/s^2
ΔV	Relative speed, expressed as $\Delta V=V_i-V$	m/s

Moreover, the driver may also interact with the function, in order to define if it is operating or not and to define several working states, as described in the following table.

Table 2: Driver desired states for *ACC* function

Variable	Type	Units
V_{driver}	Driver desired speed	m/s
T_{driver}	Driver desired time headway	s
d_{driver}	Driver desired minimal distance between vehicle, at $V=0$	m

2.3 Cruise Control

2.3.1 High level description

The aim of the cruise control is to regulate the vehicle speed around a driver desired speed. It does not take into account the environment and the possibilities in term of acceleration and deceleration is limited.

2.3.2 Low level description

The only functionality of the cruise control is to regulate the speed of the vehicle around a driver desired speed, upon its activation. We enhance this simple function with a block that handles the error flags, the functional limit and the dynamic limit according with the vehicle state.

2.3.3 Algorithm description

2.3.3.1 Speed Regulation subsystem

The vehicle speed regulation subsystem aims at regulating the speed around a driver. This subsystem achieves:

- The control of the vehicle speed with respect to driver selected speed :

$$e_s = V_{driver} - V$$

The acceleration is then expressed as a function of this error:

$$\gamma_S = K_S e_S$$

2.3.3.2 Validation subsystem

The validation subsystem achieves the control of the output with respect to the dynamic limit and functional limit. It raises an error flag as soon as the subsystem has to limit the output acceleration according to the previous limitation.

2.3.4 Validation

In order to assess the proposed CC, we demonstrate the performance on one scenario. The sensors are supposed to deliver the data defined in the deliverable D25. We also consider noises on the sensors for our simulation.

Table 3 Simulation parameters

Variable	Type	Range	Accuracy	Units
V_{driver}	Driver desired speed	[15; 40]	0.1	m/s

V	Speed of the ego vehicle	[0; 50]	0.05	m/s

For the parameter of the CC itself, we use the following values:

Table 4 CC parameter

Variable	Values
K_s	0.4

2.3.4.1 Scenario 1

We evaluate the output of the CC for a large range of variation of driver desired speed. We suppose that the vehicle speed at the origin is at 20m/s with a driver desired speed of 30m/s. After 15s, we change, each 10s for a new driver desired speed. The performance of the cruise control is however limited because of the small possible acceleration. We decide on this limit to make a clear distinction between the ACC and the CC, which is more limited in the acceleration domain. The ACC limit directly comes from the cruise control ISO norm[1].

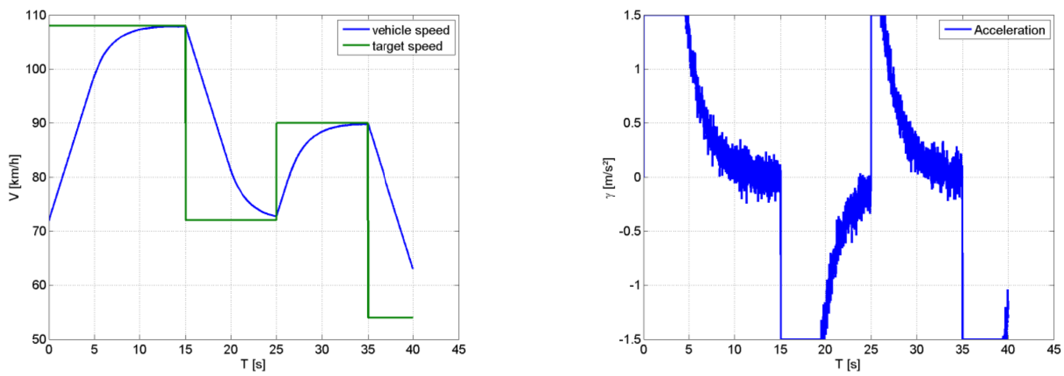

Figure 2 CC resulting speed and acceleration from variation of the driver desired speed

2.4 Adaptive Cruise Control

2.4.1 High level description

When *ACC* is active, the basic control strategy is that the vehicle speed shall be controlled automatically either to maintain a clearance to a forward vehicle, or to maintain the set speed, whichever speed is lower. The transition between these two control modes is made automatically by the *ACC* system. The I/O of the ACC is presented on the Figure 3.

[1] ISO 15622:2010, Intelligent Transport System – Adaptive Cruise Control Systems – Performance requirements and test procedure.

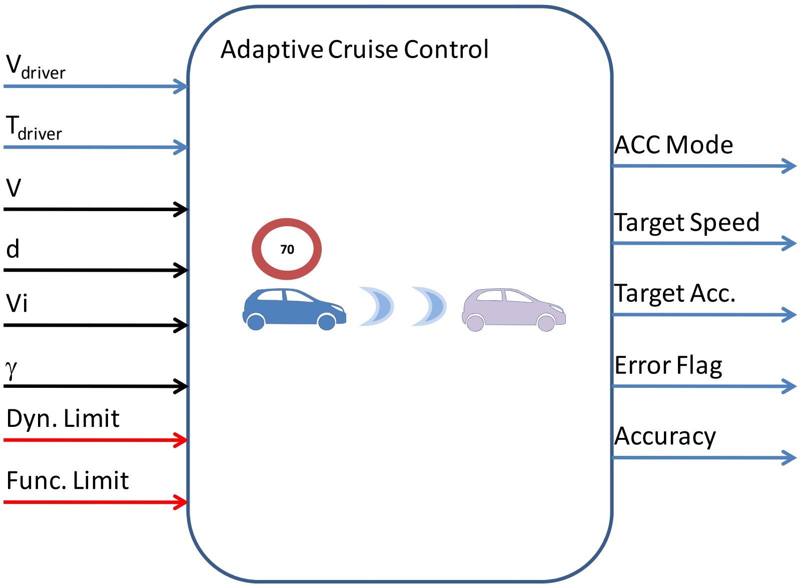

Figure 3: *ACC* I/O representation

2.4.2 Low level description

The common ACC systems use two low level functions:

- Speed Regulation: the vehicle speed must control its speed around a driver-desired speed. This function is enable when no vehicles are detected in front of our vehicle
- Vehicle Following: the vehicle controls its speed according to the speed of the lead vehicle.

The switch between each mode is decided using the following criteria:

- The clearance to the following vehicle is below the driver selected clearance
- The speed of the lead vehicle
- The speed of the ego vehicle

We enhance these two blocks with a validation subsystem that handles the error flags, the functional limit and the dynamic limit according with the vehicle state (see Figure 4).

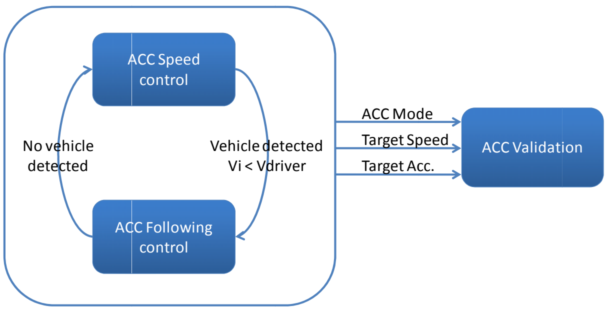

Figure 4 ACC sub functions

2.4.3 Algorithm description

2.4.3.1 Speed Regulation subsystem

Speed regulation is presented in the previous section. Only the functional limit changes in the case of an ACC: under conventional usage, the acceleration is limited at 3m/s².

2.4.3.2 Vehicle following subsystem

The vehicle following subsystem aims at controlling the vehicle when another vehicle is detected in front of our vehicle. This subsystem achieves:

- The respect of the clearance selected by the driver:

$$e_c = d - T_{driver}V$$

- The control of the vehicle speed with respect to the leading vehicle :

$$e_{sf} = V_i - V$$

The acceleration is then expressed as a function of these two errors:

$$\gamma_f = K_c e_c + K_{sf} e_{sf}$$

2.4.3.3 Validation subsystem

The validation subsystem achieves the control of the output with respect to the dynamic limit and functional limit. It raises an error flag as soon as the subsystem has to limit the output acceleration according to the previous limitation.

2.4.4 Simulation

In order to assess the proposed ACC, we demonstrate the performance on two scenarios. First one aims at speed regulation test, while the second focus on vehicle following. The sensors are supposed to deliver the data defined in the deliverable D25. We also consider noises on the sensors for our simulation.

Table 5 Simulation parameters

Variable	Type	Range	Accuracy	Units
V_{driver}	Driver desired speed	[15; 40]	0.1	m/s
T_{driver}	Driver desired time headway	[1.5; 2.5]	0.1	s
d	Distance to the front vehicle	[10; 150]	0.1	m
V	Speed of the ego vehicle	[0; 50]	0.05	m/s
V_i	Speed of the front vehicle	[0;50]	0.1	m/s

For the parameter of the ACC itself, we use the following values:

Table 6 ACC parameter

Variable	Values
K_c	0.8
K_{sf}	0.25

2.4.4.1 Scenario 1

In this scenario, we suppose that the ego vehicle is only in 'Speed following' mode. We evaluate the output of the ACC for a large range of variation of driver desired speed. We suppose that the vehicle speed at the origin is at 20m/s with a driver desired speed of 30m/s. After 15s, we change, each 10s for a new driver desired speed. The impact of the noise on the acceleration request is especially strong; however, the resulting speed is good. The performance here is better than the CC as the ACC has a larger possible acceleration.

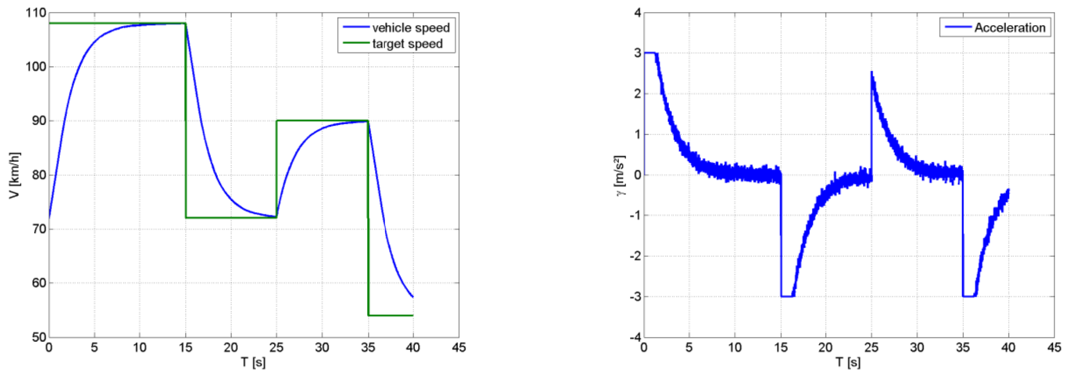

Figure 5 ACC resulting speed and acceleration from variation of the driver desired speed

2.4.4.2 Scenario 2

In this scenario, the driver desired speed remains at 30m/s. A leading vehicle, starting at 200m in front of our vehicle drive at a speed of 20m/s. After 15s, the lead vehicle decides to accelerate up to a speed of 25m/s. After 25s, the lead vehicle decelerates down to 18m/s. Then, after 35s, it accelerates up to 25m/s, then, after 45s, it accelerates up to 35m/s.

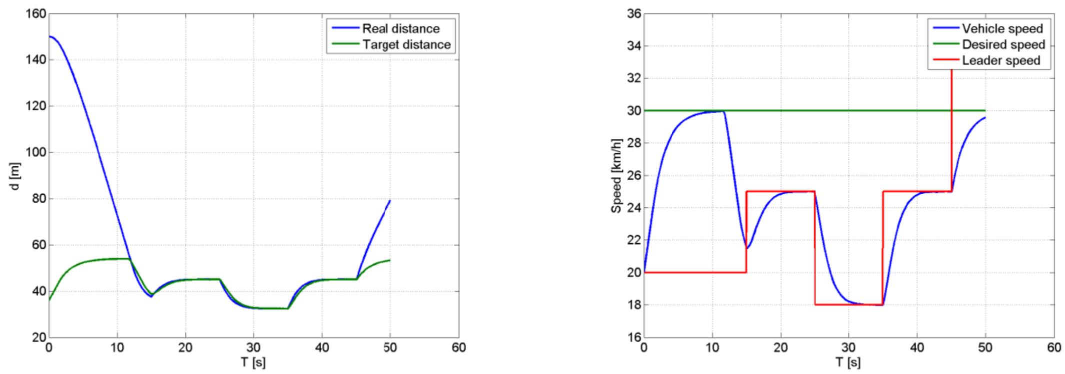

Figure 6 ACC resulting distance and speed from variation of the lead vehicle speed

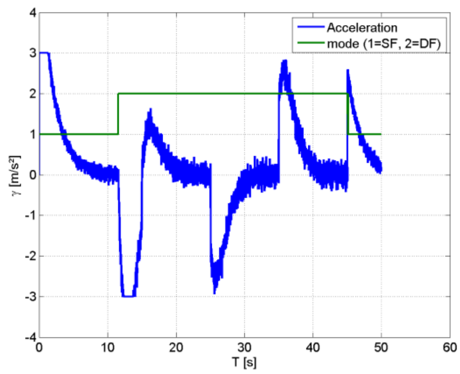

Figure 7 ACC resulting acceleration and mode selection (1 speed following, 2 distance following)

Even for large variation of the vehicle speed, the target distance can be achieved. The difference between target distance and real distance remains small.

2.5 Low Speed Following

This function is not yet finalized.

2.5.1 High level description

LSF is the counterpart of the ACC at low speed. The basic control strategy is that the vehicle speed shall be controlled automatically either to maintain a clearance to a forward vehicle, to hold the vehicle with a given minimal distance, or to maintain a maximal speed. The transition between these three control modes is made automatically by the *LSF* system. The I/O of the ACC is presented on the Figure 3.

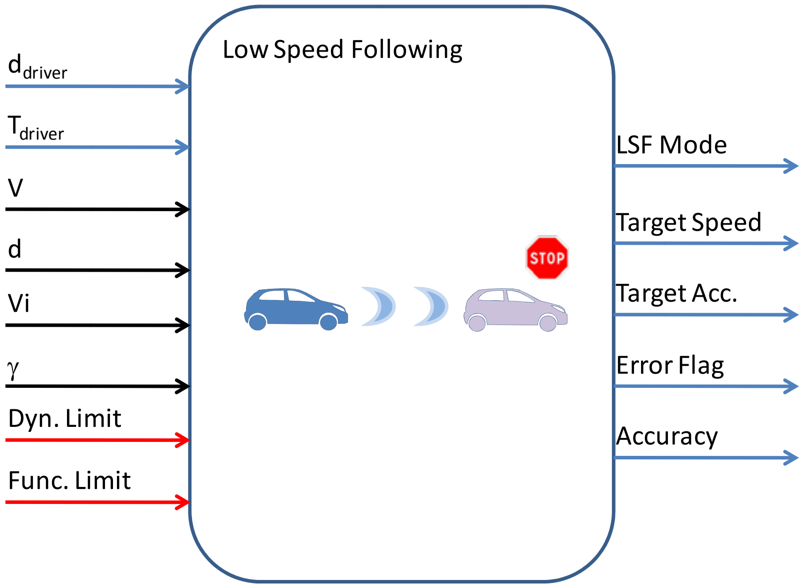

Figure 8: LSF I/O representation

2.6 Full Speed Range ACC

This function is not yet finalized.

2.6.1 High level description

The aim of the FSRACC is to close the gap between the LSF and the ACC and provides a continuous support to the driver.

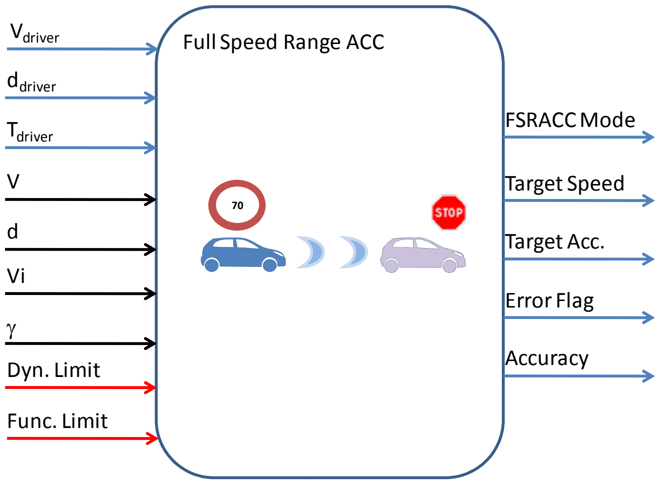

Figure 9: FSRA I/O representation

2.6.2 Low level description

The FSR ACC system embeds the ACC and LSF to provide a continuous support to the driver. However, the limitation, in term of acceleration, is not the same for each function. In order to fill the gap between each function, a ramp is created as shown in Figure 10.

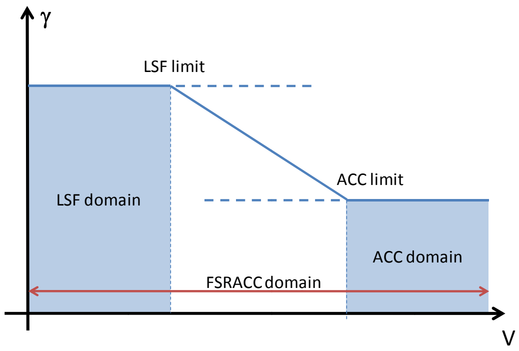

Figure 10 FSR ACC operating domain

2.7 Enhanced ACC

2.7.1 High level description

The aim of the E-ACC is to supervise the speed limit of an ACC system using both the driver desired speed, as a conventional ACC, and a safe speed that is evaluated using the digital map. This safe speed is computed to limit the lateral and longitudinal acceleration of the vehicle while driving through an infrastructure difficulty, as a curve, or a succession of curves.

2.7.2 Low level description

We only present here the computation of the safe speed limit, which is used to supervise the ACC. The remaining of the function is the same than the conventional ACC.

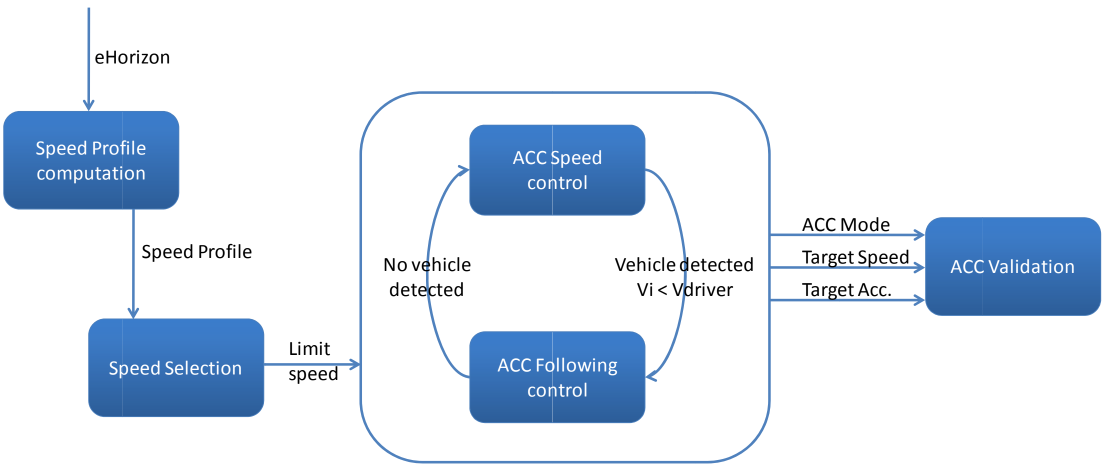

Figure 11 Low level description of the E-ACC

2.7.3 Algorithm description

The aim of the supervision module is to provide a recommended speed that can overrule the driver desired speed for safety reason. In fact, the supervision does limit the speed in order to mobilize only a limited amount of the road friction. The mobilized road friction can be expressed as:

$$\mu = \frac{F_{xy}}{F_z} = \sqrt{\frac{F_x^2 + F_y^2}{F_z^2}}$$

Where F_x, F_y and F_z are respectively the forces at the tire road interface in longitudinal, lateral and vertical direction, which can be either on the front (f) or rear (r) axle. The tire road forces can be defined using the desired acceleration (longitudinal γ_{lon}, lateral γ_{lat} and vertical γ_{vert}) and the road attributes (slope ϕ_r, banking θ_r, and curvature):

$$\begin{cases} M\gamma_{lat} = F_{yf} + F_{yr} - M\,g\sin\phi_r \\ I_z\ddot{\psi} = L_f F_{yf} - L_r\,F_{yr} \end{cases}$$
$$\begin{cases} M\gamma_{vert} = F_{zf} + F_{zr} + M\,g\cos\theta_r \\ I_y\ddot{\theta} = H\,F_m - L_f F_{zf} + L_r F_{zr} \end{cases}$$
$$\{M\gamma_{lon} = F_m + M\,g\sin\theta_r$$

M is the vehicle mass and F_m is the motor force to achieve a given acceleration, I_y and I_z are the inertia of the vehicle body along the *y* and *z* axle respectively. Moreover, we can express the lateral acceleration to follow the trajectory defined by the road and the longitudinal acceleration as the desired speed variation:

$$\begin{cases} \gamma_{lat} = \rho_r V^2 \\ \gamma_{lon} = V\dfrac{dV}{ds} \end{cases}$$

By limiting the longitudinal and lateral maximal mobilized friction to $\lambda_{lon}\mu_{max}$ and $\lambda_{lat}\mu_{max}$, the expression of the speed is given by the following system:

$$\begin{cases} V^2 = \dfrac{g}{\rho_r}\left(\left(1 + \dfrac{H}{L_r}\theta_r\right)\sqrt{1 - \left(\dfrac{\theta_r}{\lambda_{lon}\mu_{max}}\right)^2}\,\lambda_{lat}\mu_{max} - \phi_r\right) \\ 1 = \left(\dfrac{1}{\lambda_{lat}\mu_{max}}\dfrac{\dfrac{\rho_r V^2}{g} + \phi_r}{1 - \dfrac{H}{L_r}\left(\dfrac{V}{g}\dfrac{dV}{ds} - \theta_r\right)}\right)^2 + \left(\dfrac{\dfrac{V}{g}\dfrac{dV}{ds} - \theta_r}{\lambda_{lon}\mu_{max}}\right)^2 \end{cases}$$

The first equation gives the speed in the worst part of the difficulty, while the second one defines the speed to reach safely the difficulty.

Table 7 Variables definition for E-ACC

Variable	Type	Units

g	Gravity acceleration	m/s²
θ_r	road slope	rad
ϕ_r	road banking	rad
ρ_r	road curvature	1/m
H	CoG height	m
L_f, L_r	Distance from CoG to front and rear axle	m
μ_{max}	Maximal available road friction	--
$\lambda_{lat}, \lambda_{lon}$	Maximal mobilization	--

Figure 12 shows the computation of the safe speed for a succession of curves, the second one being sharper than the first.

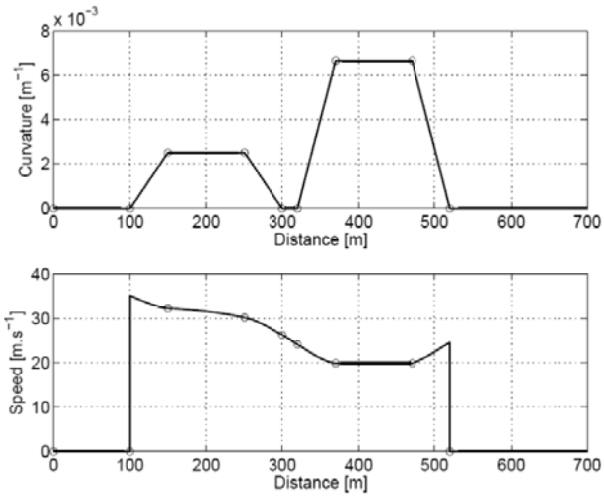

Figure 12 Computation of the safe speed for E-ACC

2.7.4 Validation

As we can limit the longitudinal acceleration, the vehicle under ACC could easily follow the speed profile that is generated as soon as the limitation is under the functional limit of the ACC. Then, this last parameter is used to limit the variable λ_{lon}.

We have computed the safe speed profile on the Satory test track for various values of the λ_{lon} and λ_{lat} parameters (ranging from 0.3 to 0.5).

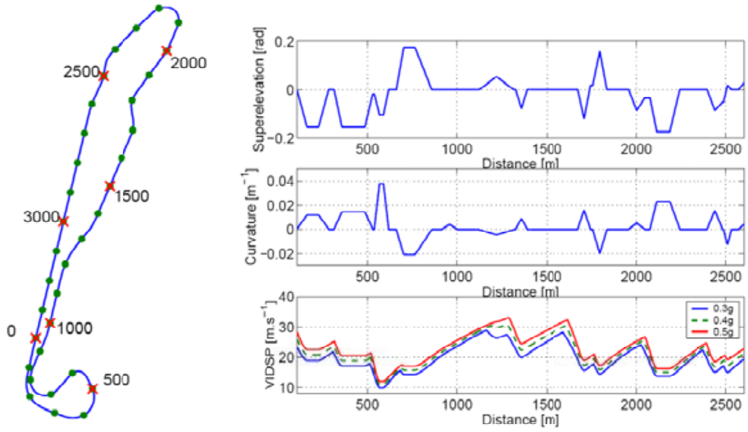

Figure 13 Speed profile computed for various parameters on Satory test track

2.8 Smart And Green ACC

This function is not yet developed.

The Smart And Green ACC function (*SAGA*) adapts the vehicle speed over all its speed range according to a forward vehicle and to road events in a near horizon (legal speed change, grade, presence of crossings) in the aim to reduce the energy consumption.

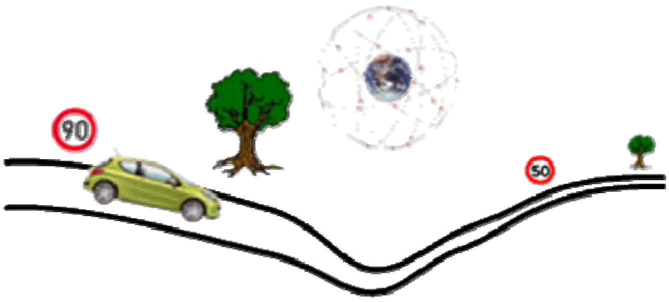

Figure 14: SAGA illustration

The basic control strategy is to adjust automatically the vehicle speed either to maintain a clearance to a forward vehicle, to maintain the set speed or to maintain a speed reducing the energy consumption, whichever speed is lower. The change between the three modes is automatic. During the hold state, the automatic brake control is used to keep the subject vehicle stationary.

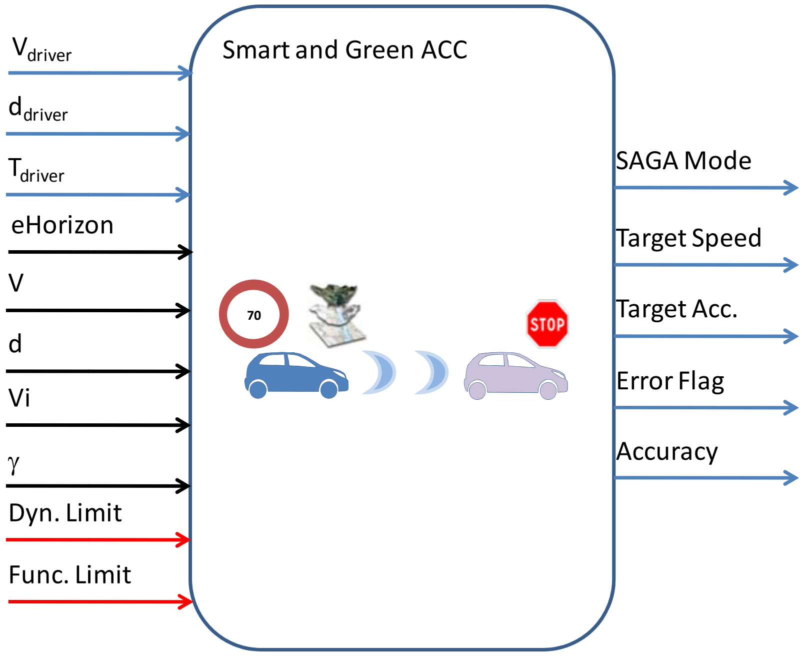

Figure 15: SAGA I/O representation

2.9 Autonomous Emergency Braking

2.9.1 High level description

Autonomous Emergency Braking or Collision Mitigation Systems is a function that applies the brakes in critical situations, independently of the driver, to avoid or mitigate the accident. The variables used by AEB are almost the same as by the ACC or the Stop&Go function.

The basic control strategy is to adapt the deceleration of the vehicle in order to avoid or mitigate the crash. The distance to the front object is measured by the perception functions and if this distance reaches a minimum safety distance (related to the ego vehicle speed), the brakes are autonomously activated.

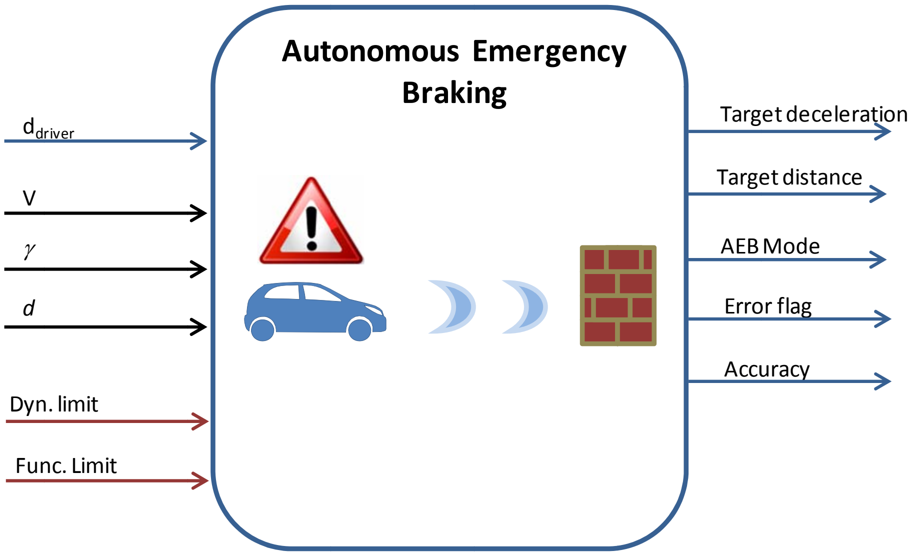

Figure 16: AEB I/O representation

2.9.2 Low level description

Most of the AEB developed or under-development are based on a static definition of mitigation strategy. For instance (Figure 17), R. Labayrade[2] generates a strong braking (around 0.8g) if a collision situation is detected with a TTC, time to collision, below 1s; Honda and Mercedes propose on the high-end vehicle an advanced AEB that applies the following strategy:

1. Warn the driver if a collision situation is detected within a 1.6s time range
2. Start a slow deceleration (around 0.1g) if no correction occurs at 1s, and act on the seat belts.
3. Generate a strong deceleration (between 0.6g and 0.8g) if the TTC drops below 0.6s.

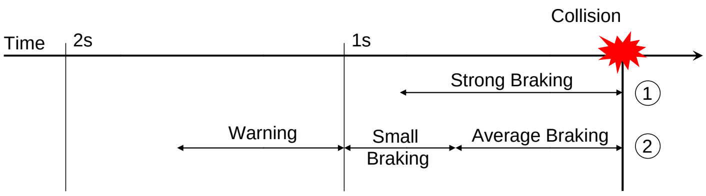

Figure 17 AEB strategy

Whatever the strategy selected is, it uses the TTC which is defined as the time remaining for two vehicles to collide if they continue on the same path at the present speed:

$$T_{tc} = \frac{d}{V - V_i}$$

[2] Labayrade R., Royere C., Aubert D. Experimental assessment of the RESCUE collision-mitigation system, IEEE transactions on vehicular technology, 2007, vol. 56, no1, pp. 89-102

Figure 18 presents the low level description of the AEB. It includes the evaluation of the TTC itself, as defined with the previous formulae, the AEB management, which aims at defining the strategy and the mode and the braking strategy that defines the braking request.

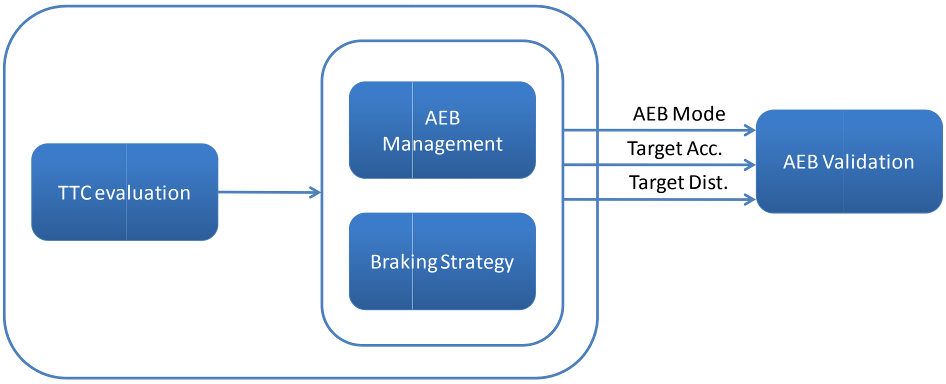

Figure 18 Low level description of the AEB

2.9.3 Algorithm description

2.9.3.1 AEB Management

The aim of the AEB management is to determine the risk related to the current situation, and to avoid overwarning. The warning is mainly based on the TTC value, however, we also use the variation of the TTC and the acceleration of the vehicle to assess exactly the state of the AEB and avoid frequent switching.

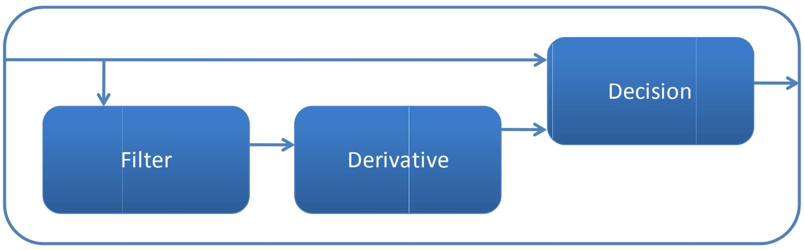

Figure 19 AEB Management architecture

The decision part is based on the following graph. The decision specifically looks for the area in red.

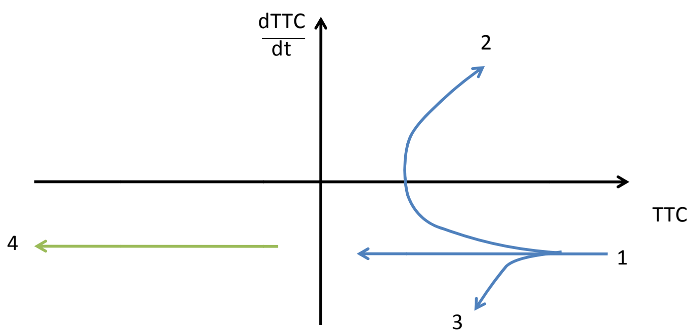

Figure 20 Possible trajectory in the (TTC,dTTC) plane

Figure 20 shows the possible trajectories in this plane. The trajectory 4 is the trajectory of a lead vehicle that has a speed higher than the ego vehicle; its TTC is negative and increasing. For the trajectory 1, 2 and 3, the lead vehicle speed is lower than the ego vehicle speed. For trajectory 1, no action is done. For the trajectory 2, the lead vehicle driver accelerates and for the trajectory 3, the lead vehicle driver accelerates.

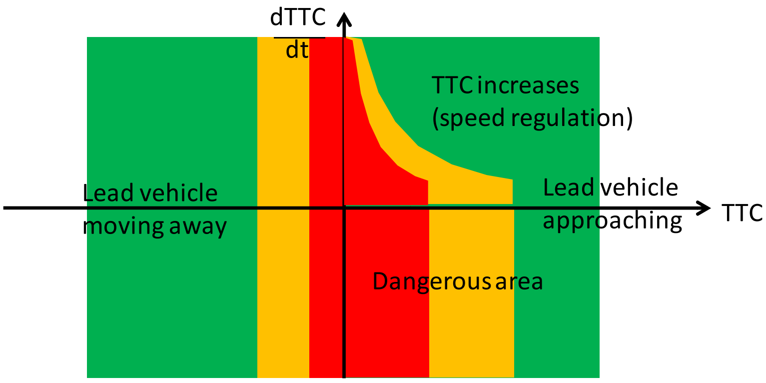

Figure 21 Decision process

The decision reacts accordingly to these rules:

- If TTC>0
    - TTC < 1.6s : a warning is emitted
    - TTC < 1s and TTC*dTTC/dt < K1 : small braking
    - TTC < 0.6s and TTC*dTTC/dt <K2 : strong braking
- If TTC<0
    - |TTC| <1s : warning
    - |TTC| < 0.6s : small braking

K1 and K2 are the parameters of the decision unit.

2.9.3.2 Braking strategy

The braking strategy only verifies the information from the AEB management block in order to apply the braking. It also verifies that the current braking request from the driver is below the braking request from the AEB before applying it. Two levels of braking are possible:

- Small braking : 0.1g
- Strong braking : 0.6g

This sub function is also responsible to decrease the breaking demand with respect to variation of mode.

2.9.4 Validation

2.9.4.1 Scenario 1

In this scenario, our vehicle approaches a slow moving vehicle. At the beginning of the simulation, our vehicle drives at 50km/h, the front vehicle is at 50m and its speed is 30km/h. The front vehicle neither accelerates nor decelerates.

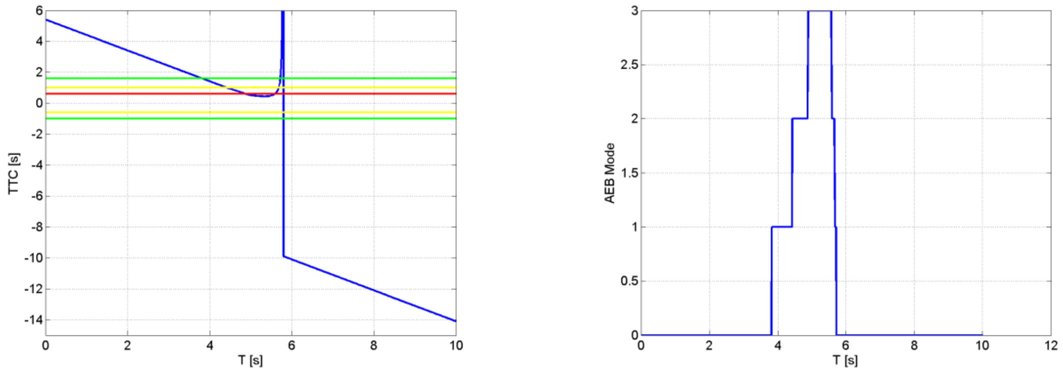

Figure 22 TTC and resulting mode for the AEB

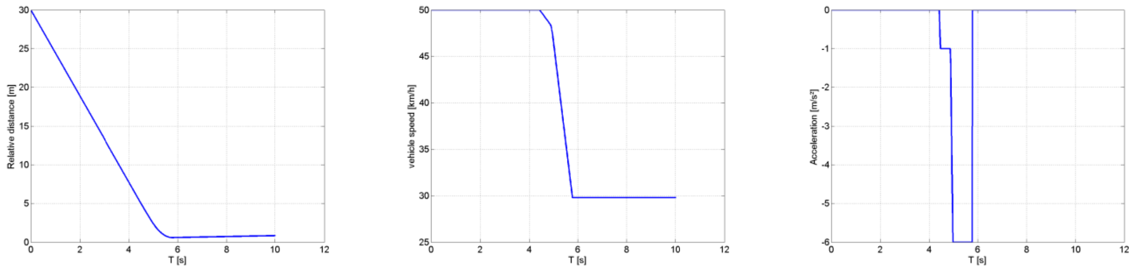

Figure 23 Relative distance, speed and acceleration of the ego vehicle

In the first graph, we can see that the TTC decrease linearly as no action is done. When it decreases below 1.6s, the function switches to mode 1 (warning), then at 1s it goes to mode 2 (small braking demand is realised). Finally when TTC goes below 0.6s, a strong braking is generated. The deceleration impacts the TTC which increases up to two seconds. However, the braking demand remains constant, as the speed of the ego vehicle is too high with respect to the obstacle. If the braking was lower, then the ego vehicle would still approach the front vehicle, decreasing the TTC, and asking for another strong braking. In order to avoid this oscillation, the demand remains constant until the ego vehicle speed is under the front vehicle speed. As the ego vehicle speed is lower than the front vehicle speed, the TTC becomes negative.

2.9.4.2 Scenario 2

Our vehicle approaches a static obstacle. The speed at the beginning of the simulation is 50km/h. The function acts as in the previous scenario. However, the threshold set for the strong deceleration is too low to allow the ego vehicle to stop before the obstacle.

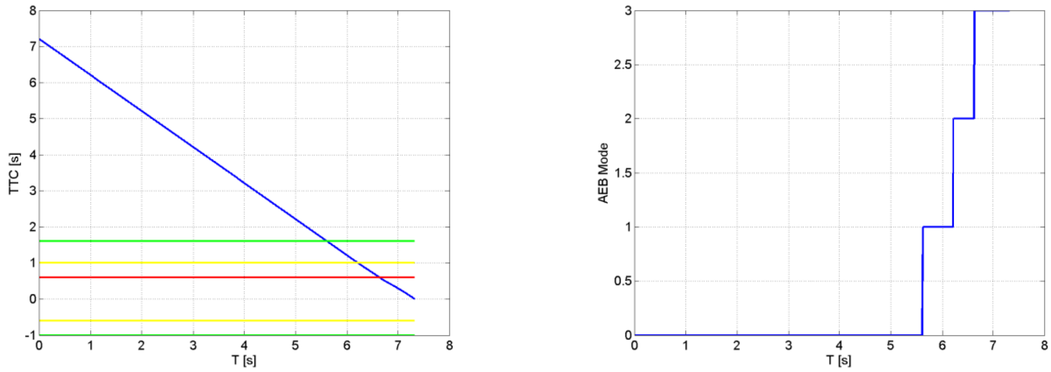

Figure 24 TTC and resulting mode for the AEB

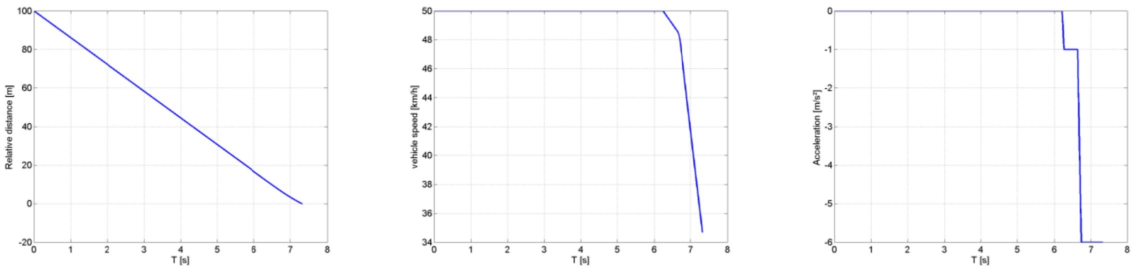

Figure 25 Relative distance, speed and acceleration of the ego vehicle

However, the speed at the time of the collision is largely decreased: instead of having a vehicle speed of 13.9m/s, the vehicle speed is 9.6m/s. According to Nilsson[3], this leads to a drop of car accident fatalities of 75%.

3 Lateral ADAS

Most of lateral ADAS presented in the following sections relie on several variables and the main ones are listed in Table 8: Road inputs and Table 9: Vehicle inputs.

[3] Nilsson, G. (1982). The effects of speed limits on traffic accidents in Sweden. In: Proceedings of the international symposium on the effects of speed limits on traffic accidents and transport energy use, 6-8 October 1981, Dublin. Organisation for Economic Co-operation and Development OECD, Paris, p. 1-8.

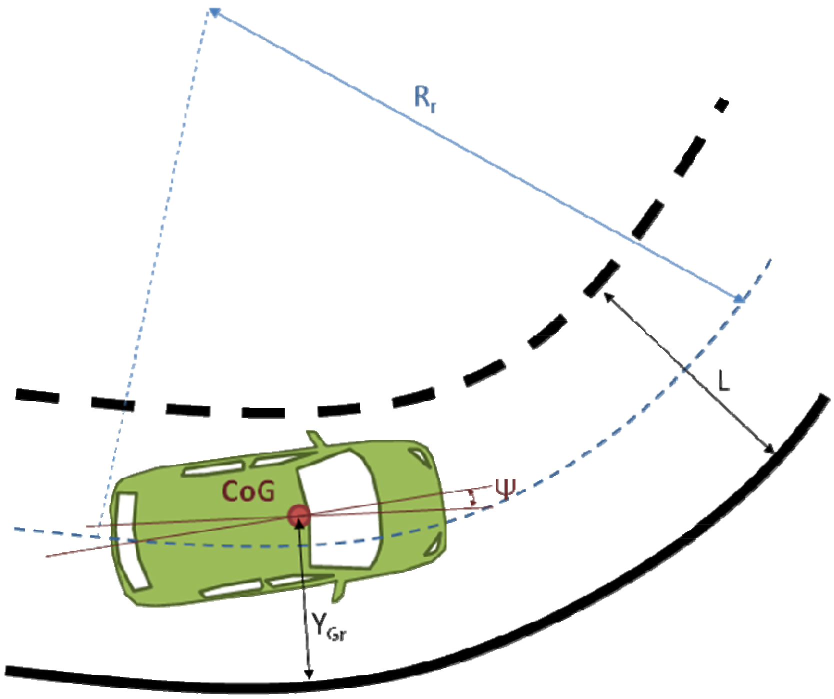

Figure 26: Notations for the LDW function

Table 8: Road inputs

Variable	Type	Units
L	Lane width	m
R_r	Road radius of curvature	m

Table 9: Vehicle inputs

Variable	Type	Units
V	Speed of the ego vehicle	m/s
γ	Acceleration of the ego vehicle	m/s²
Y_{Gr}	Distance from the right lane boundary to vehicle CoG	m
Ψ	Vehicle relative yaw angle	radians
δ_f	Steering angle	radians

Moreover, the driver may also interact with the function, in order to define if it is operating or not and to define several working states, as described in the following table.

Table 10: Driver inputs

Variable	Type	Units
St	Function state (0 if off, 1 if on)	Boolean
B	Blinkers activated (O or 1)	Boolean
Sy	Function sensitivity	--

3.1 Lane Departure Warning

3.1.1 High level description

The Lane Departure Warning warns the driver if the vehicle is predicted to leave its lane by detecting the road markings and measuring the current vehicle dynamic state.

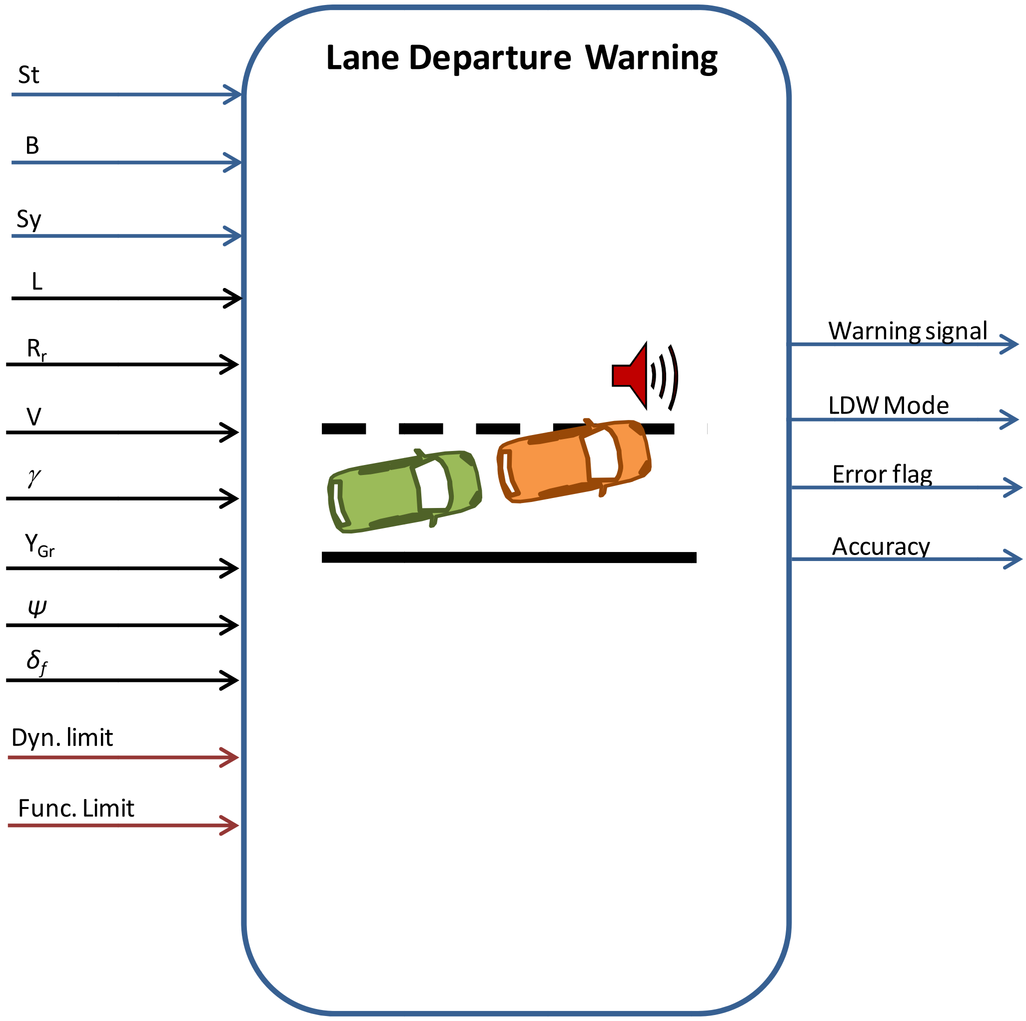

Figure 27: LDW I/O representation

3.1.2 Low level description

The LDW module uses the information of the position inside its lane to compute a Time to Lane Crossing to provide a warning to the driver in case of unintentional lane departure.

Figure 28: Low level description of LDW

3.1.3 Algorithm description

The algorithm is working in several steps:

- estimation of TLC from the relative yaw angle and the lateral deviation;
- if the TLC is under a predefined threshold, a signal is transmitted to the driver (sonorous and luminous signal). This threshold varies according to the sensitivity set by the driver;
- when the TLC goes above the threshold, the signal is deactivated;
- the information on blinkers can be used to deactivate temporarily the LDW function.

The TLC is computed using the remaining distance to cross the lane border following a constant trajectory (constant speed, constant yaw rate) divided by the current vehicle speed. Then the exact computation of the TLC directly depends on the road detection.

3.1.4 Validation

The scenario tested for the validation of LDW consists in a straight line with a vehicle travelling at 100 km/h and an initial yaw angle of 4°. This value is maximum for road departure and the trajectory plotted in Figure 29 shows that the vehicle has barely the time to stay on the road. A virtual driver, inactive while the signal has not been emitted once, is added in this scenario. A delay corresponding to the reaction time of the driver has also been set to represent his reaction.

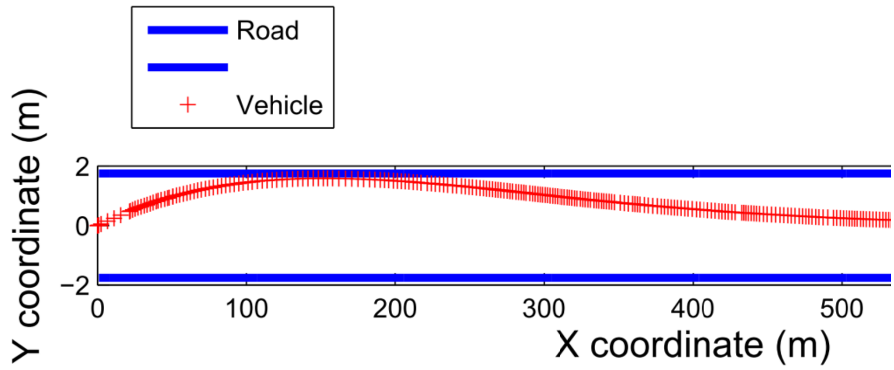

Figure 29: LDW vehicle trajectory

Figure 30 shows that, at the beginning of the simulation, the TLC is diminishing. Then, it reaches the predefined threshold and the sonorous signal is emitted (blue line going from 0 to 1). Then, the driver reacts after his reaction time (set to 0.8s in this case) and as the vehicle is getting closer to the centre line, the TLC increases. The figure on the right presents the steering angle generated by the driver. On this figure it can be seen that this angle is 0 until the driver reacts.

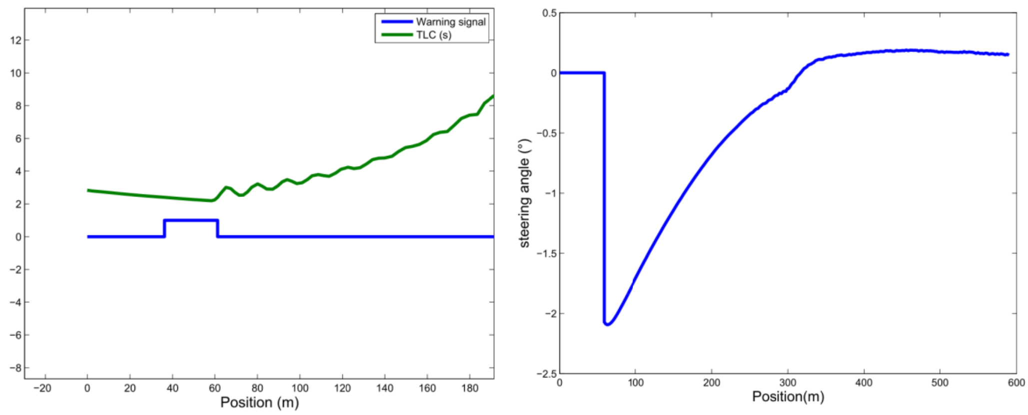

Figure 30: LDW variables (left) and steering angle (right)

3.2 Lane Keeping Assistance System

3.2.1 High level description

The Lane Keeping Assistance System (LKAS) acts on the vehicle commands to keep it in its lane by detecting the road markings and measuring the current vehicle dynamic state. In the eFuture project the copilot acts on the front wheels torques, generating a yaw rate, to make the vehicle turn.

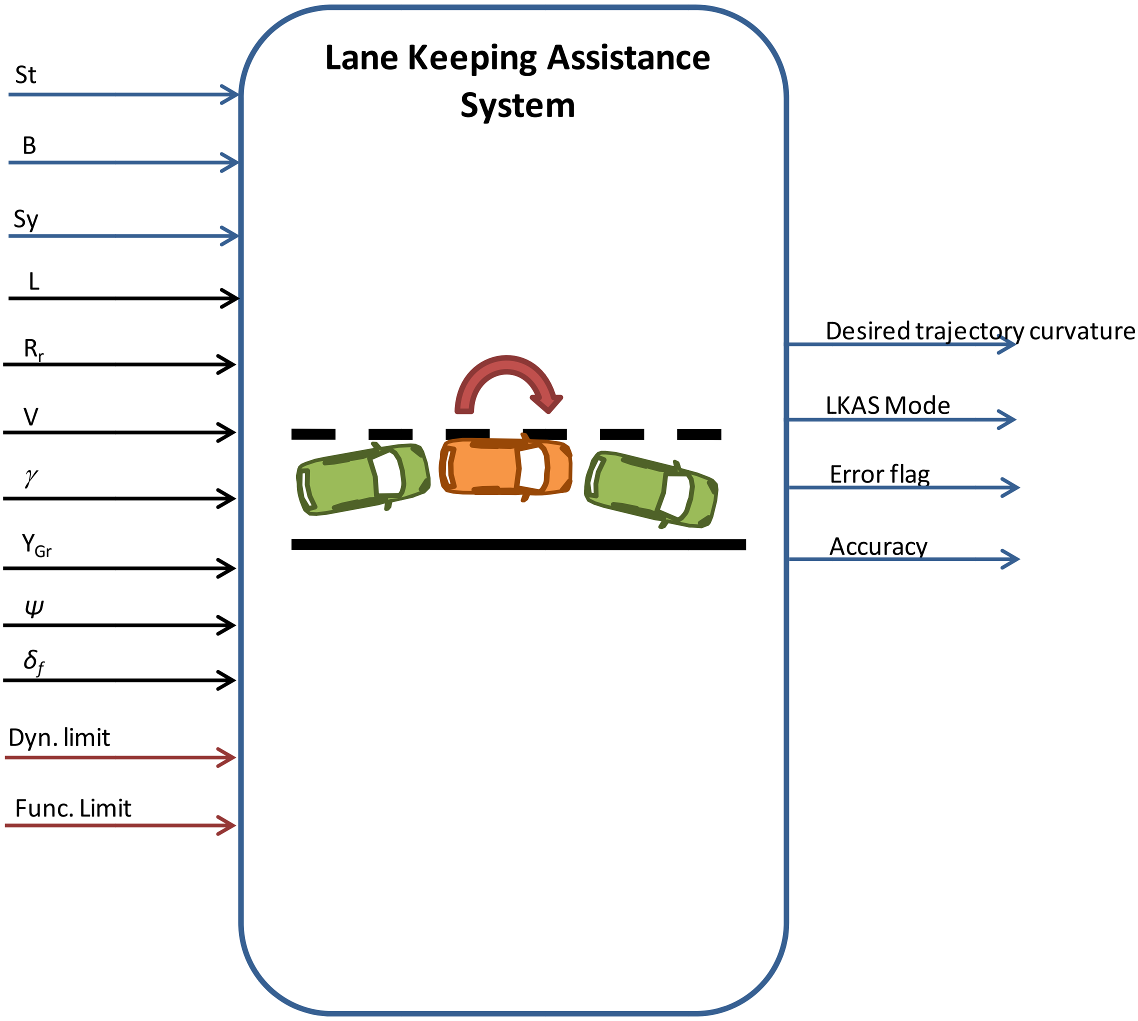

Figure 31: LKAS function I/O representation

3.2.2 Low level description

The only functionality of the LKAS is to keep the vehicle in its lane, upon its activation. We enhance this simple function with a block that handles the error flags, the functional limit and the dynamic limit according to the vehicle state.

3.2.3 Algorithm description

The basic idea of the eFuture LKAS is to predict the steering angle needed to keep the vehicle on the desired trajectory. This strategy can be decomposed in three steps :

- Computation of the required yaw rate φ_R from the predicted (Y_P) and desired (Y_D) lateral displacement. Using a Taylor's second order expansion and supposing that the side slip angle is negligible, it can be found that:
  - $\varphi_R = -\frac{2V}{L_S^2}(Y_D - Y_P)$;

- Computation of the steering angle α required to reach this yaw rate. This is performed with the inverse transfer function of a vehicle dynamic model. Here a linear two-degree-of-freedom is used, two-wheels model;
- Computation of the predicted lateral deviation from the target line. The predicted lateral displacement is deduced from the real lateral displacement Y_{Gr} and from the relative yaw angle Ψ: $Y_P = Y_{Gr} + L_s\Psi$, where L_s is the prediction distance. This prediction has to take into account the road curvature.

This algorithm implementation with Simulink is illustrated in figure Figure 32.

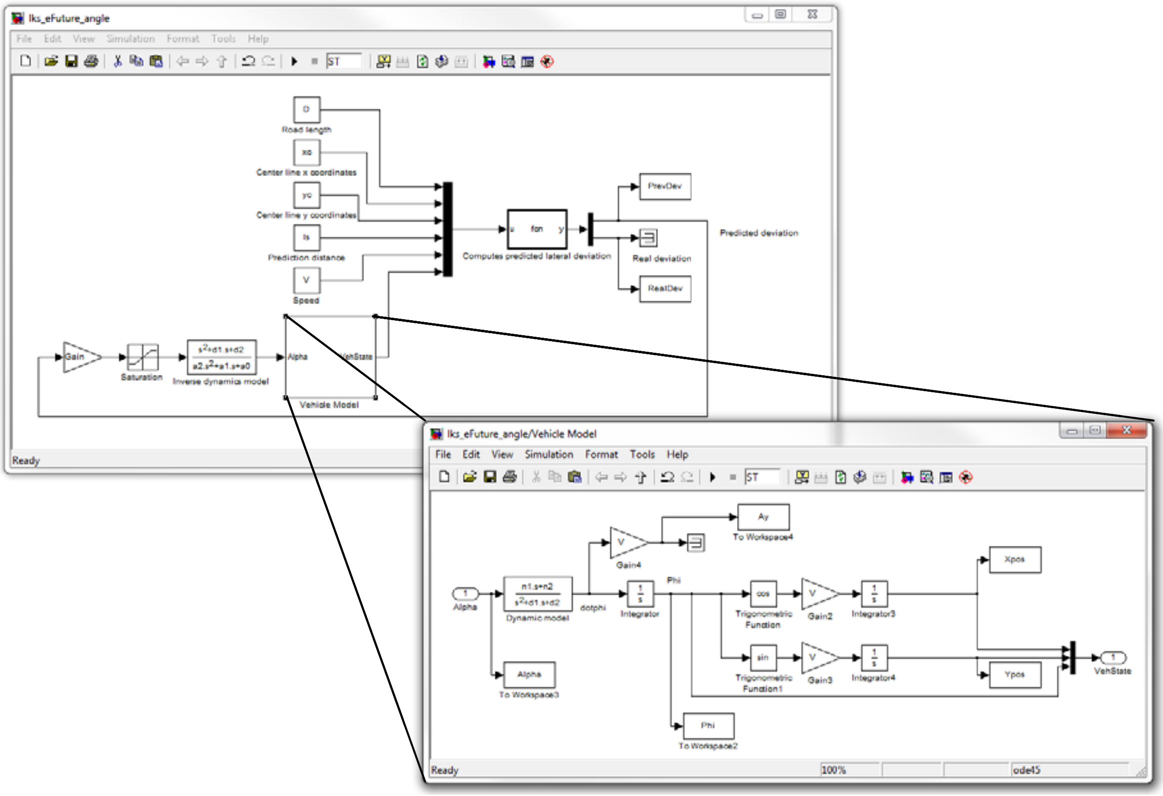

Figure 32: Simulink algorithm illustration

3.2.4 Validation

3.2.4.1 Scenario 1

In this scenario, the vehicle approaches a "S" curve at a constant speed of 100 km/h. The curvature radius measures 600 m over 400 m and -600 m on the remaining distance. The vehicle trajectory on this simulated road is presented in Figure 33.

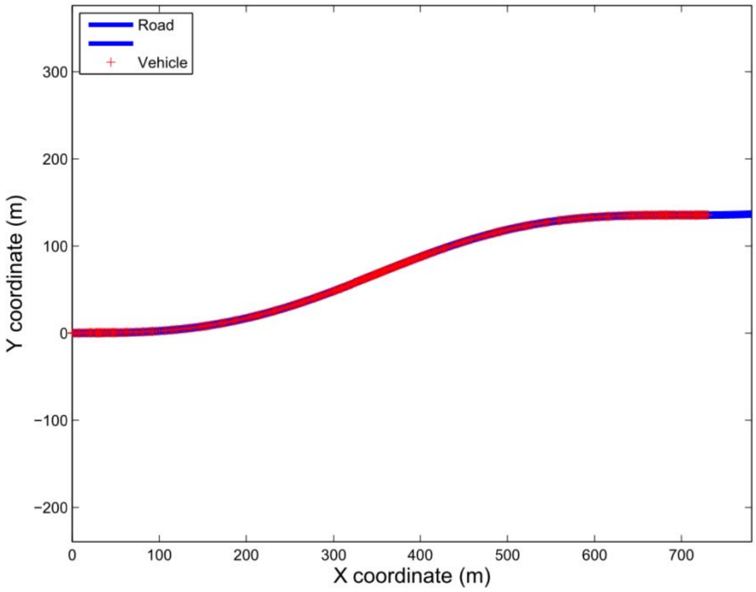

Figure 33: LKAS vehicle trajectory in "S" curve

In Figure 34, on the left, the lateral deviation predicted and effective are showed. It can be seen that the predicted deviation in green can be rather high but the real deviation in blue is contained in the [-0.5;0.5] range. The vehicle succeeds in staying in its lane. The steering angle required to follow this trajectory is plotted on the right.

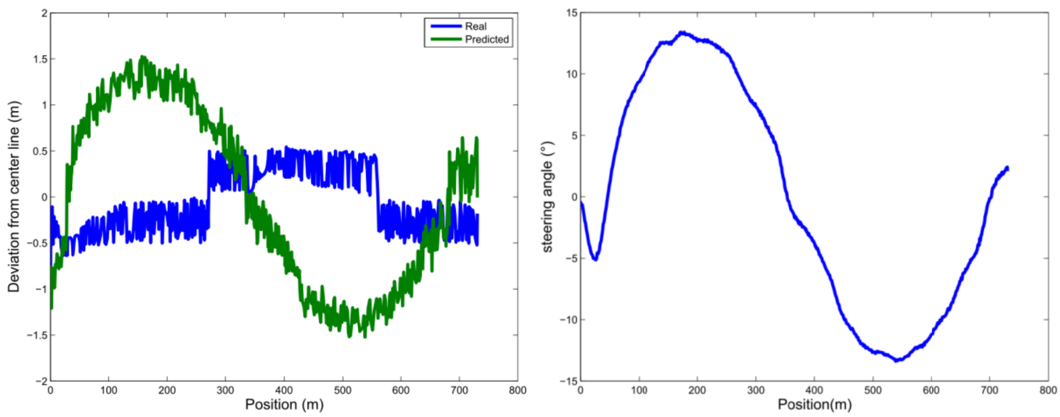

Figure 34: left, LKAS lateral deviation from center line in "S" curve; right, LKAS steering angle in "S" curve

3.2.4.2 Scenario 2

In this scenario, the vehicle is travelling at 100 km/h on a straight road with an initial relative yaw angle of 5°. The road segment is 800 meters long and the first part of the trajectory is presented on Figure 35. The vehicle slightly deviates from the center line at the beginning of the road then get back to the desired line. In this case, the vehicle is close to the lane boundary but a yaw of 5° is very high and usually taken as maximum for the LKAS function.

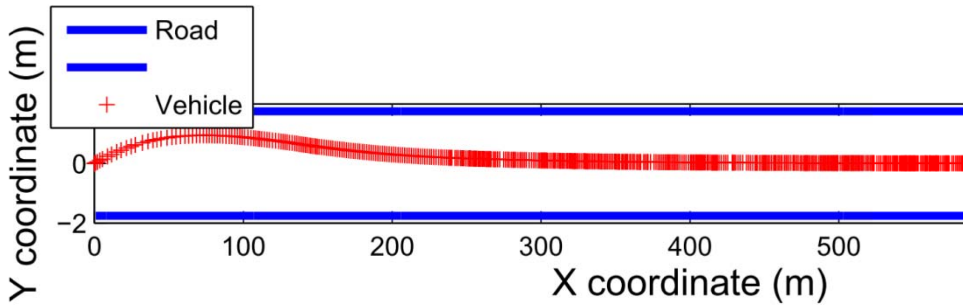

Figure 35: LKAS vehicle trajectory on a straigt road

This effect is more clearly detected on the lateral deviation in Figure 36 where the maximum is deviation is about 1 meter.

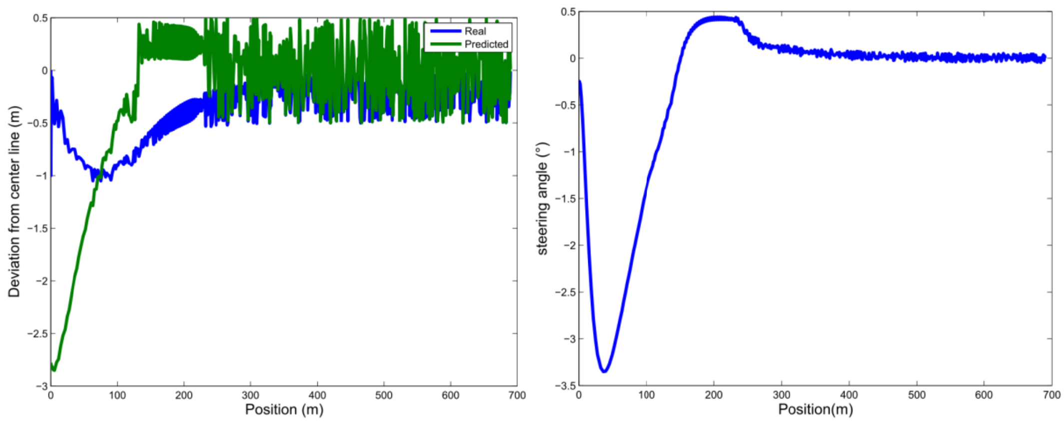

Figure 36: LKAS lateral deviation (left) and steering angle (right) on a straight road

3.3 Lane Change System

This function is not yet developed.

3.3.1 High level description

The Lane Change System, when blinkers are activated, acts on the vehicle commands to change its lane by detecting several lanes, measuring the current vehicle dynamic state and by detecting the presence of other vehicles in the road lanes.

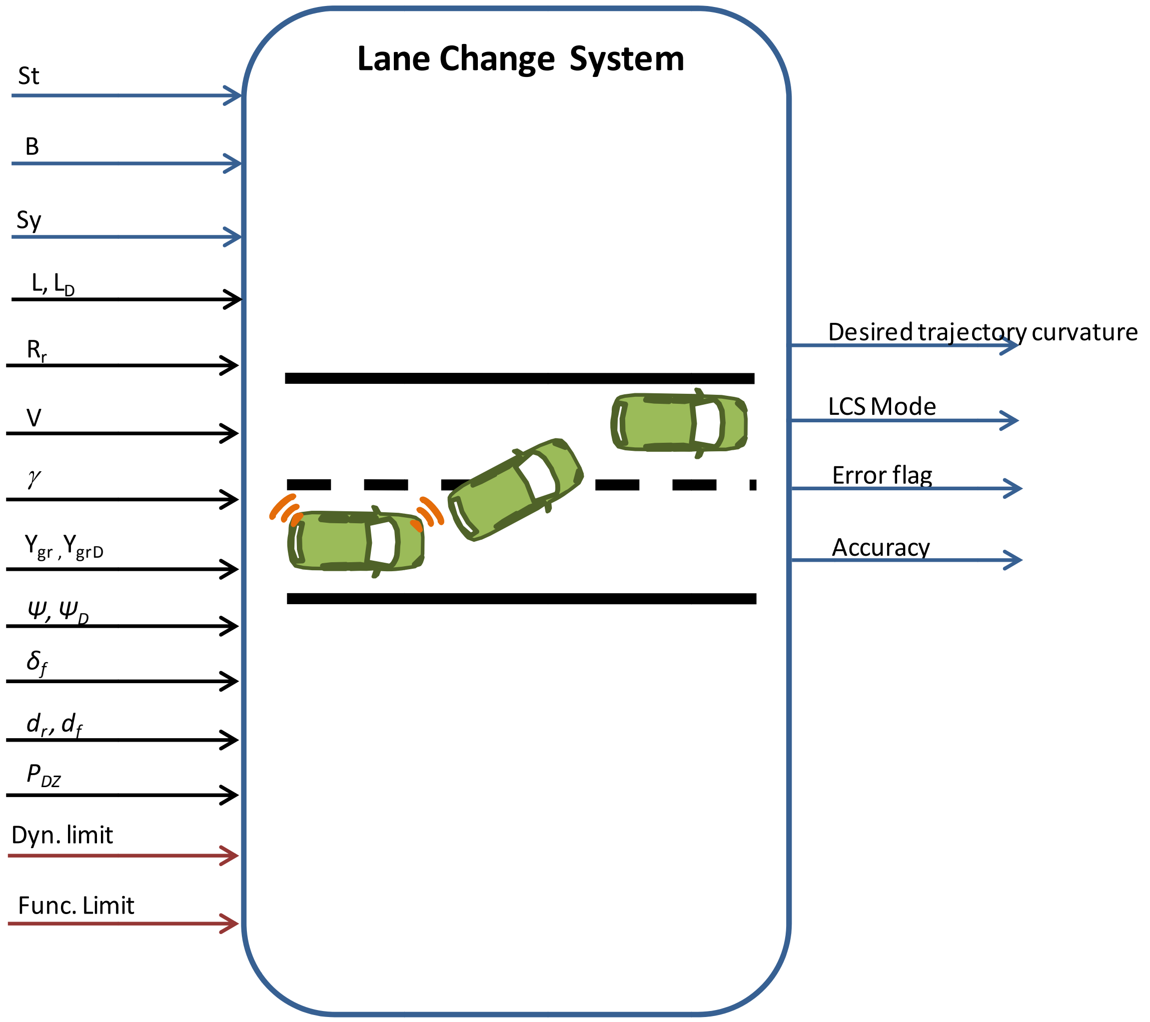

Figure 37: LCS function I/O representation

3.3.2 Low level description

This function can be decomposed in four steps:

- evaluation of the collision risk implied by a lane change maneuver;
- lane change decision: From results of the risk estimation, the lane change is engaged or not;
- lane change maneuver control. This step is performed using a modified LKAS function to take into account a larger lateral deviation;
- validation of the function.

These steps are illustrated in Figure 38

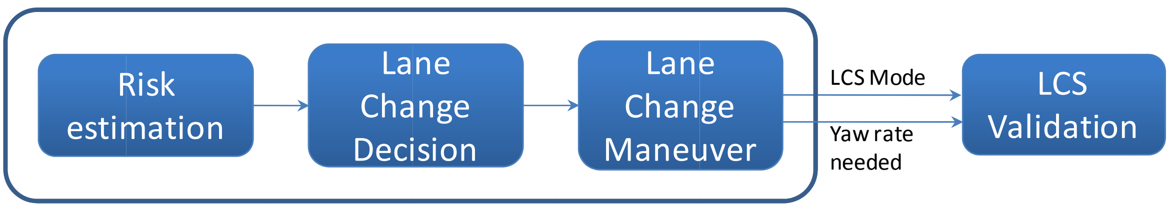

Figure 38: Lane Change description

3.3.3 Validation

Only the demonstration of the lane change manoeuvre module is presented in Figure 39 where it can be seen that the manoeuvre module succeeds in changing the vehicle lane. The vehicle speed is 100 km/h on a straight road.

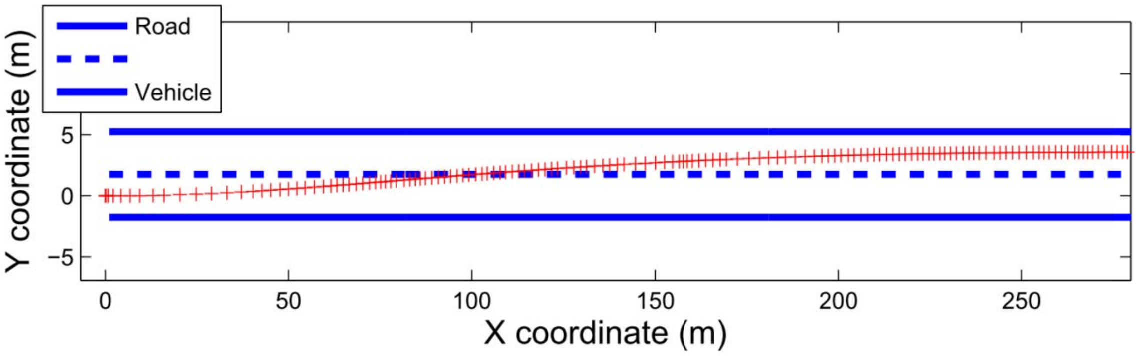

Figure 39: LCS manoeuvre trajectory

Acronyms

ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance System
AEB	Autonomous Emergency Braking
FSRA	Full Speed Range ACC
LCS	Lane Change System
LDW	Lane Departure Warning
LSF	Low Speed Following
LKAS	Lane Keeping Assistance System
SAGA	Smart And Green ACC
TLC	Time to Lane Crossing