PREDICTIVE TIME-TO-LANE-CROSSING ESTIMATION FOR LANE DEPARTURE WARNING SYSTEMS

Gianni Cario
Alessandro Casavola
Giuseppe Franzè
Marco Lupia
{gcario,casavola,franze,mlupia@deis.unical.it},
DEIS-Università degli Studi della Calabria,
Via Pietro Bucci, Cubo 42-C, 87036, Rende (CS),
Italy
Gianluigi Brasili
g.brasili@infomobility.it,
Infomobility.it S.p.A.,
Via Isidoro e Lepido Facii,
64100, Sant'Atto di Teramo (TE),
Italy
Paper Number 09-0312

ABSTRACT

This paper presents a data fusion algorithm which is able to robustly estimate the Time-to-Lane-Crossing (TLC) of a vehicle traveling along a lane on the basis of road images, collected by an on-board videocamera, and kinematic data coming from car sensors. This algorithm is instrumental to built Lane Departure Warning Systems (LDWS) with enhanced predictive capabilities which allow the generation of earlier warnings able to better prevent dangerous driving situations coming from unintentional vehicle lane crossing occurrences. Comparisons with no predictive strategies are carried out and discussed in order to verify the effectiveness of the proposed approach in some critical driving scenarios simulated within the Carsim simulation framework.

INTRODUCTION

Traffic safety is a key problem in nowadays automotive industry, having relevant social and economic impacts. The European Road Safety Observatory (ERSO) document [1] reports over than $1,000,000$ accidents, with around $40,000$ fatalities, only in 2006. In Italy, more than 15% [2] of this amount is due to driver fatigue and negligence: in many cases the driver falls asleep, making the vehicle to leave its designated lane and possibly causing an accident.
During the last two decades much effort has been devoted to the development of Advanced Driving Assistance Systems (ADAS). AntiLock Braking (ABS) or Electronic Stability Program (ESP) apparatuses are well known examples of such systems, the latter being now standard equipments in all commercial cars where they contribute to the overall vehicle stability and safety. Many new ideas and concepts for enriching existing ADAS systems with new functionalities are currently under development or have been recently introduced into the market. Amongst many, we focus here on the development of Lane Departure Warning Systems (LWDS) which, according to a recent report of the EU Intelligent Car Initiative [3], are supposed to have the potentiality to save $1,500$ accidents in 2010, given a 0.6% of penetration rate, and $14,000$ in 2020 for a penetration rate of 7%.

LWDS refer to systems that try to help the driver to stay into the lane. A DSP equipped with an on-board camera is typically used to identify the lane strips, computing the position and the heading with respect to the lane and the TLC by exploiting data, such as wheel speeds and yaw and steering angles, taken from the car ECUs, via the CAN bus, or provided by additional sensors. Some interesting contributions to LDWS development can be found in [4]-[10]. An interesting approach is the so-called TLC-based method, first proposed by Godthelp et *al.* [11], where an alarm is triggered when the TLC is below a specified threshold. Such systems typically use acoustic or vibration warnings, the latter applied to the driving seat or the steering wheel. In general, TLC-based methods provide earlier warnings than roadside rumble strips (RRS), because alarms are triggered with sufficient advance before the driver being really in danger.

Here, the development of a TLC-based LDWS system is described. A single calibrated camera has been used for capturing road images and a data fusion algorithm has been implemented and used for determining the lane markings and estimating the TLC time. Details on the data fusion algorithm are also reported.

The outline of the paper is as follows. First, the TLC estimation problem is defined and two strategies for its computation described. Hence, the mathematical model of the vehicle is described and used, along with an Extended Kalman Filter, for data fusion. Computer simulations and comparisons with no-predictive approaches are presented. Finally, some conclusions end the paper.

SYSTEM OVERVIEW

In this section we will give an overview of the **LDWS** system under development. Figure 1 depicts the set of devices used for estimating the vehicle dynamics and detecting the road stripes. In particular, it is assumed that the vehicle is equipped with a camera mounted behind the windshield, an absolute angle sensor is used for measuring the steering angle and an angular speed sensor is mounted on a rear wheel.

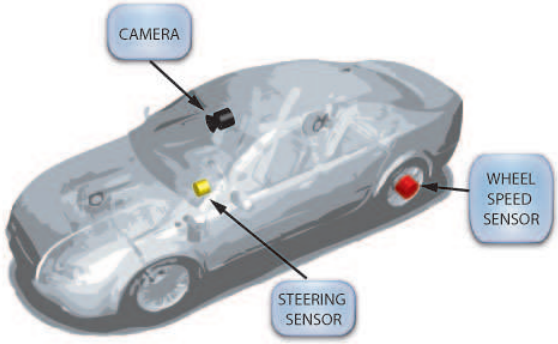

Figure 1. System Overview.

In figure 2, the overall functional and computational scheme for the proposed **LDWS** system is reported. The ingredients can be summarized as follows. Based on a mathematical description of the vehicle, that will be discussed later, the **LDWS** consists of two functional blocks: the *Data Acquisition and Elaboration* and the *Warning Generation* modules.

In the *Data Acquisition and Elaboration* module, the lane geometry and the vehicle position relative to the lane are estimated from the camera frames: such a task is of course crucial to detect a lane departure because it provides unique information for that purpose, no derivable by other on-board sensors.

Because all driver assistance systems share the need of knowing the driving surroundings, the information coming from the *Video Frame Elaboration* and from the *Sensor Data Elaboration*, i.e. elaboration of kinematics data coming from the on-board sensors, are combined into a model of the vehicle surroundings by using a suitable *Data Fusion* algorithm. Typically, a data fusion algorithm operates in discrete time cycles. At each step, a measurement update includes new sensor measurements into the model, while a time update predicts the model behavior from the current state towards the next fusion cycle.

In this paper, such a phase will be performed by means of an Extended Kalman Filter [12]. The *Warning Generation* module is in charge to generate an alarm whenever necessary on the basis of information coming from the *Data Acquisition and Elaboration* module. The latter consists of a *Lane Departure Detection* scheme which is mainly based on the computation of an estimate of the TLC time.

Finally, the LDWS could be connected to some Human Machine Interface (HMI), e.g. acoustic alarms or LCDs, in order to advise the driver of the forthcoming lane departure.

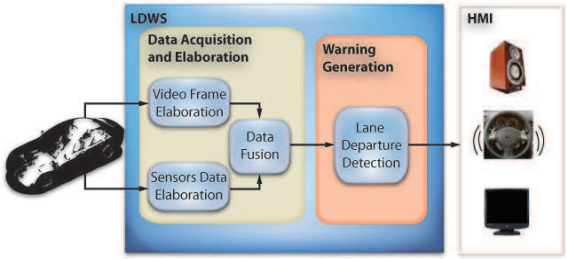

Figure 2. System Overview.

TIME TO LINE CROSSING

Roughly speaking, TLC can be defined as the available time interval before a vehicle crosses any lane boundary following a pre-specified path direction. An important application of TLC in driver warning systems is to detect instances when the vehicle actually moves out of the lane and to warn the driver in order to avoid an immediate accident. As a consequence, it could be considered a further indicator to support the driver assistance in case of severe impairments caused by drowsiness.

In the last decade, many researchers have studied the problem of an exact TLC computation (see [13],[14] and references therein). Unfortunately, exact real-time TLC computation is not an easy task due to several limitations concerning an *a priori* knowledge of both the vehicle trajectory and the lane geometry. Beside this, another major restriction factor is the complexity of its computation in real-time. In the sequel, we will discuss two methods that allow a quite effective TLC evaluation.

The trigonometric computation

This first method has been proposed in [13] and the key idea is that the vehicle is rarely driving on a straight path, therefore it has been assumed that the vehicle trajectory alternates between curves to left and to the right. A mathematical description of the TLC is as follows

$$TLC = \frac{DLC}{u}, \quad \forall\, u > 0 \qquad (1).$$

where DLC $[m]$ is the distance to lane crossing along the vehicle path and u $[m/s]$ the vehicle speed. Note that the parameter DLC is directly computable via the cosine rule (see Figures 3, 4):

$$DLC = \alpha\, R_v \qquad (2).$$

where the radius of the vehicle path R_v is computed as $R_v = u/r$, with r $[rad/s]$ the yaw rate. As far as the parameter α is concerned, it represents the angle between the line from the centre point (X_v, Y_v) of the vehicle trajectory to the lane departure point d and the line

from the front wheel to the centre point (see Figures 3, 4). Differently from R_v, its computation depends on the road geometry. First, let us consider the straight road scenario (Figure 3). In this case, α is computed

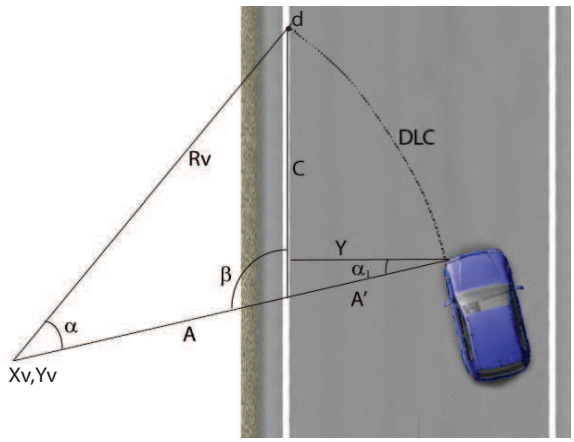

Figure 3. DLC calculation for a straight road.

by using the cosine rule

$$\alpha = \arccos\left(\frac{A^2 + R_v^2 - C^2}{2\,A\,C}\right) \qquad (3).$$

where

- $A = Rv - A'$ and $A' = Y/cos(\alpha_1)$, with Y the distance between the front wheel and the lane boundary (along a perpendicular line to the road) and α_1 the angle between such a perpendicular line and the line from the front wheel to the centre point (X_v, Y_v)
- $C = \frac{2\cdot A\cdot\cos(\beta)+\sqrt{(2\cdot A\cdot\cos(\beta))^2-4(A^2-R_v^2)}}{2}$

Conversely, Figure 4 depicts the road curve scenario. In

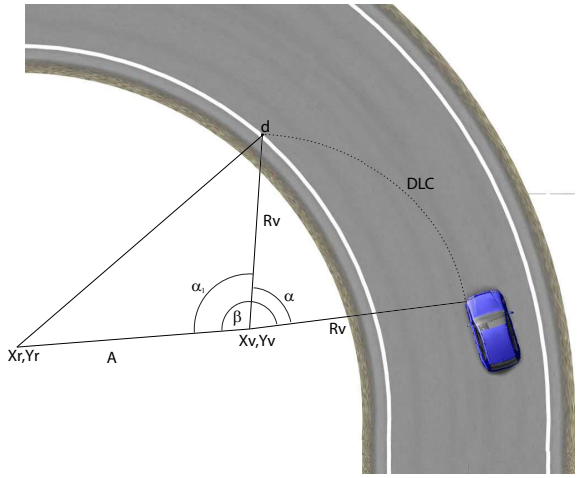

Figure 4. DLC calculation for curved road.

this other case, α is differently computed as

$$\alpha = \beta - \alpha_1 \qquad (4).$$

where

- β is the angle between the line passing from the centre point (X_v, Y_v) of the vehicle curve to the centre point of the road curve (X_r, Y_r) and the line passing from (X_v, Y_v) and the left front wheel (if the vehicle turns towards the inner lane boundary);
- $\alpha_1 = \arccos\left(\frac{(A^2+R_v^2-R_r^2)}{(2\cdot A\cdot R_v)}\right)$, with A the distance between (X_r, Y_r) and (X_v, Y_v), and R_r the radius of the curved road segment.

The approximate computation

In practice, the computation of the TLC is performed by using an approximation procedure because the trigonometric method is based on the knowledge of relevant parameters , e.g. the distance to line crossing, the radii of the vehicle path and of the curved road segment, that in real scenarios are not available. Even if an approximate computation of the TLC is of interest, its calculation depends on the following assumptions

a) the lateral vehicle position Y is *a priori* known or can easily be measured

b) the lateral vehicle velocity $\dot{Y}$ is constant, which imposes that the vehicle preserves a constant velocity while approaching to lane boundaries

It is well recognized that the computation of the lateral vehicle position Y is a more simpler task than the computation of the vehicle radii and curved road segment paths [14].
Then, the TLC can be easily computed as the ratio between the lateral position and the rate of change of the lateral position [13]

$$TLC = \frac{Y}{\dot{Y}} \qquad (5).$$

It is worth to note that, even if the assumption b) is not realistic, in [13] the expression (5) has been proved to be a tight overestimation of the minimum TLC (1), whose accuracy increases as the time to cross the lane decreases. Finally, it is interesting to recall that the use of this approximation has provided good simulation results as testified in [13].

LATERAL SPEED ESTIMATION

This section is devoted to describe a Kalman-based filter for lateral speed estimation purposes. In the sequel, we will first discuss the mathematical vehicle model, then the extended Kalman filter will be outlined and applied to the vehicle model under consideration.

Vehicle Model

A vast variety of mathematical models able to describe the vehicle dynamics during driving have been proposed in the literature (see [15], [16] for a detailed survey). In

most cases, even if many models are very accurate, they usually require a good knowledge of many vehicle parameters (stiffness, yaw moment of inertia, etc.) and this precludes their practical use [17], [18]. Therefore, it is necessary to look for mathematical models that are sufficiently accurate and simple to be used in practical contexts. A well-known vehicle description that satisfies these requirements is the kinematic model proposed in [19].

Such a model is based on a three state description, that comprises the Cartesian coordinates (x, y) of the vehicle CoG, mid-way centered between the rear wheels, and the vehicle orientation angle ϕ. Following the notation of Figure 5, V_{RW} hereafter denotes the longitudinal velocity of the rear wheels, V_{FW} the longitudinal velocity of the front wheels (taking care of the steered angle δ) and B the wheelbase. Then, a continuous-time description can be derived as follows:

$$\begin{cases} \dot{x}(t) &= V_{RW}(t)\cos(\phi(t)) \\ \dot{y}(t) &= V_{RW}(t)\sin(\phi(t)) \\ \dot{\phi} &= \frac{V_{FW}(t)\sin(\delta(t))}{B} \end{cases} \quad (6).$$

with

$$\begin{aligned} V_{FW}(t) &= \frac{V_{RW}(t)}{\cos(\delta(t))} \\ \dot{\phi}(t) &= \frac{V_{RW}(t)\tan(\delta(t))}{B} \end{aligned}$$

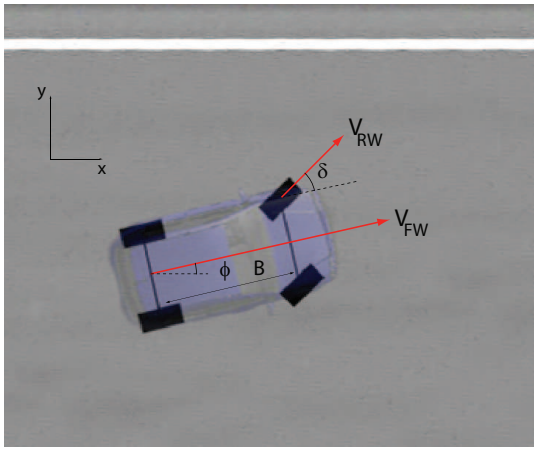

Figure 5. Vehicle's kinematic model

The continuous-time system (6) can be discretized using forward Euler differences with a sampling time ΔT. As a result, the following discrete-time description is achieved

$$\begin{bmatrix} x(k) \\ y(k) \\ \phi(k) \end{bmatrix} = \begin{bmatrix} x(k-1) + V_{RW}(k)\Delta T\cos(\phi(k-1)) \\ y(k-1) + V_{RW}(k)\Delta T\sin(\phi(k-1)) \\ \phi(k-1) + \frac{V_{RW}(k)\Delta T}{B}\tan(\delta(k)) \end{bmatrix} \quad (7).$$

Note that such a model does not take into consideration the discrepancy between the *vehicle speed* and the *wheel speed* when spinning or skidding phenomena occur. The same holds true for the difference between the measured steer angle and the actual angle steered in presence of wheel side slip situations. Therefore, to compensate for some of these effects a *wheel radius* state can be added, hereafter named $R(k)$. In particular, such a quantity increases when the wheel slips whereas it conversely decreases when the wheel skids. Beside this, it is important to underline that the wheel radius varies w.r.t. different vehicle payloads, temperatures and tyre pressures. Hence, the following four state model can be proposed:

$$\begin{bmatrix} x(k) \\ y(k) \\ \phi(k) \\ R(k) \end{bmatrix} = \begin{bmatrix} x(k-1) + \omega(k)R(k-1)\Delta T\cos(\phi(k-1)) \\ y(k-1) + \omega(k)R(k-1)\Delta T\sin(\phi(k-1)) \\ \phi(k-1) + \frac{\omega(k)R(k-1)\Delta T}{B}\tan(\delta(k)) \\ R(k-1) \end{bmatrix} + \begin{bmatrix} \epsilon_x(k) \\ \epsilon_y(k) \\ \epsilon_\phi(k) \\ \epsilon_R(k) \end{bmatrix} \quad (8).$$

where $R(k) \in \mathbb{R}$ and $\omega(k)$ is the forward wheel angular velocity measured by the wheel sensor. Moreover, the additive vector $[\epsilon_x(k),\ \epsilon_y(k),\ \epsilon_\phi(k),\ \epsilon_R(k)]^T$, whose components are stochastic processes with zero mean and fixed variances, reflect inaccuracies in the state model and the static error occurring when the vehicle is in a steady-state condition.

Extended Kalman filter

The Kalman Filter (KF) [12],[20] is one of the most widely used methods for tracking and estimation due to its simplicity, optimality, tractability and robustness. However, the application of the KF to nonlinear systems can be difficult. The most common approach is to use the Extended Kalman Filter (EKF) [21], [22] which simply linearizes the nonlinear model along the trajectory so that the traditional linear Kalman filter can locally be applied at each computational step.

Let us consider the following nonlinear discrete-time system

$$\begin{aligned} x_k &= f_{k-1}(x_{k-1}) + w_{k-1}, & (9). \\ z_k &= h_k(x_k) + v_k & (10). \end{aligned}$$

where x_k represents the state vector of the system, z_k the measurement vector, w_k the noise process due to disturbances and modelling errors and v_k the measurement noise. It is assumed that the noise vectors w_k and v_k are zero-mean, uncorrelated and with covariance matrices $Q_k = Q_k^T > 0$ and $R_k = R_k^T > 0$ respectively, ie.

$$w_k \sim \aleph(0, Q_k), \quad v_k \sim \aleph(0, R_k)$$

The signal and measurement noises are assumed uncorrelated also with the initial state x_0. Then, the estimation problem can be stated, in general terms, as follows: given the observations set $Z_k := \{z_0, z_1, \ldots, z_k\}$ evaluate an estimate $\hat{x}_k$ of x_k such that a suitable criterion is minimized. In the sequel, we will consider the mean-square error estimator, and therefore, the esti-

mated value of the random vector is the one that minimizes the cost function

$$J[\hat{x}_k] = E[(x_k - \hat{x}_k)^2 | Z_k] \quad (11).$$

At each time instant k, the EKF design can be split in two parts: time update (prediction) and measurement update (correction). In the first part, given the current estimates of the process state $\hat{x}_{k-1}$ and covariance matrix P_{k-1} and based on the linearization of the state equation (9)

$$\Phi_k = \left. \frac{\partial f_k}{\partial x} \right|_{x=\hat{x}_{k-1}} \quad (12).$$

the updating of the covariance matrix and state prediction $\hat{x}_{k|k-1}$ are performed as follows

$$P_{k|k-1} = \Phi_k P_{k-1} \Phi_k^T + Q_k, \quad (13).$$
$$\hat{x}_{k|k-1} = f_k(\hat{x}_{k-1}) \quad (14).$$

Then, given the current measurement z_k and by linearizing the output equation (10) according to

$$H_k = \left. \frac{\partial h_k}{\partial x} \right|_{x=\hat{x}_{k|k-1}} \quad (15).$$

the following Kalman observer gain is derived

$$K_k = P_{k|k-1} H_k^T (R_k + H_k P_{k|k-1} H_k^T)^{-1} \quad (16).$$

Finally, the state and the matrix covariance estimates are updated as

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k(z_k - h_k(\hat{x}_{k|k-1})), \quad (17).$$
$$P_k = (I - K_k H_k) P_{k|k-1} \quad (18).$$

and the procedure is iterated.
Because the aim is to use the EKF for estimating the lateral position $y(k)$ and the yaw angle $\phi(k)$ of the vehicle model (8), real measurements $(y(k), \phi(k))$ are needed. Such a task will be accomplished by resorting to data made available by the vision system, because we assume that the vehicle is not equipped with gyroscopes and/or radar/GPS devices.

VISION SYSTEM

This section is devoted to describe the proposed vision algorithm. Two main phases can be characterized: Lane Detection and Lane Tracking. It is assumed that a camera is mounted behind the vehicle windshield and used for capturing road image frames.

Lane Detection

The lane detection system is represented in Figure 6. It consists of all steps related to each frame elaboration in extracting relevant features. It includes five steps that will be discussed in details below.

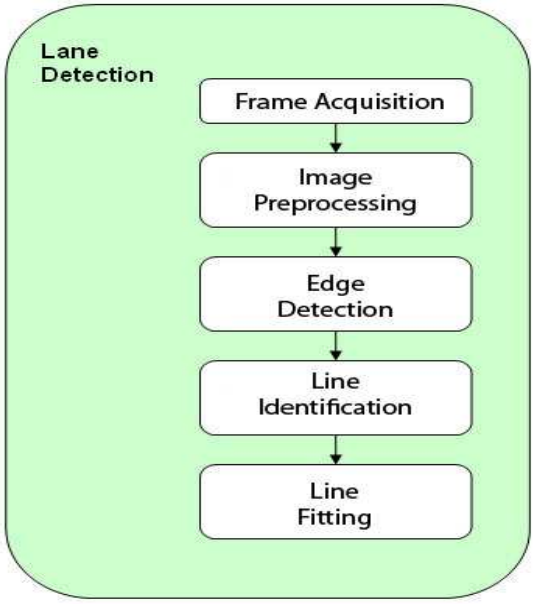

Figure 6. Lane Detection.

Frame acquisition – In this first phase, the aim is to recover image frames from the vehicle camera. To this end, it is important to adequately set the camera position on the vehicle and its orientation w.r.t. the horizontal road line. An example of an acquisition frame is shown in Figure 7.

Figure 7. Frame acquisition.

Image Preprocessing - Inverse Perspective – Once an image frame is obtained, an image processing phase is required. Here, we apply the Inverse perspective mapping, hereafter denoted as **IPM**.
The **IPM** is a geometrical transformation technique that re-maps each pixel of the 2D perspective view (see Figure 7) of a 3D object in a new planar image (see Figure 10) with a bird's eye view. In other words, the **IPM** is the projection from the image plane $I = (u, v) \in \mathbb{R}^2$ onto the Euclidean space $W = (x, y, z) \in \mathbb{R}^3$ (world space) [23], [24], [25].
Therefore, the side view geometrical model of the **IPM** is as depicted in Figure 8, while Figure 9 represents the top view geometrical model.

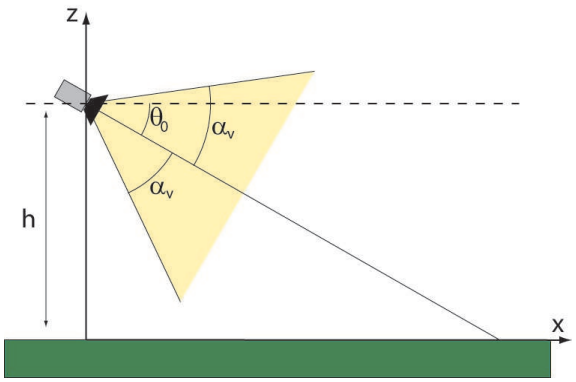

Figure 8. Inverse Perspective: lateral view.

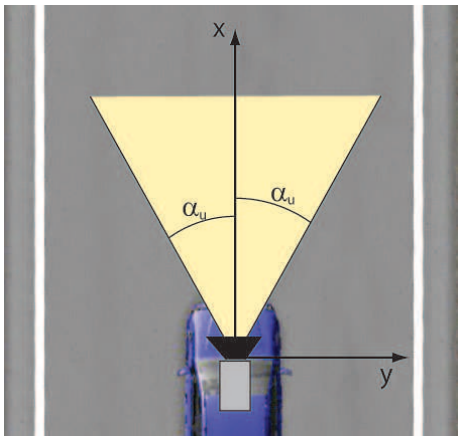

Figure 9. Inverse Perspective: bird view.

In particular, the equations describing the projection from the image plane I onto the world space W and viceversa are given by

$$\begin{aligned} x(r) &= h\left(\frac{1+\left[1-2\left(\frac{r-1}{m-1}\right)\right]\tan(\alpha_v)\tan(\theta_0)}{\tan(\theta_0)-\left[1-2\left(\frac{r-1}{m-1}\right)\right]\tan(\alpha_v)}\right) \\ y(r,c) &= h\left(\frac{1+\left[1-2\left(\frac{c-1}{r-1}\right)\right]\tan(\alpha_u)}{\sin(\theta_0)-\left[1-2\left(\frac{r-1}{m-1}\right)\right]\tan(\alpha_v)\cos(\theta_0)}\right) \end{aligned} \quad (19).$$

$$\begin{aligned} r(x) &= \frac{m-1}{2}\left(1+\frac{h-x\tan(\theta_0)}{h\tan(\theta_0)+x}\coth(\alpha_v)\right)+1 \\ c(x,y) &= \frac{n-1}{2}\left(1-\frac{y}{h\sin(\theta_0)+x\cos(\theta_0)}\coth(\alpha_u)\right)+1 \end{aligned} \quad (20).$$

where

- h the height of camera w.r.t. the ground level;
- $m \times n$ the image resolution;
- (r, c) image pixel coordinates;
- α_v and α_u vertical and horizontal camera half-angle of view, respectively;
- θ_0 pitch camera angle.

By using the equation (19), each **IPM** image pixel is rephrased by adopting the real world metric coordinates (x, y), see Figure 10.

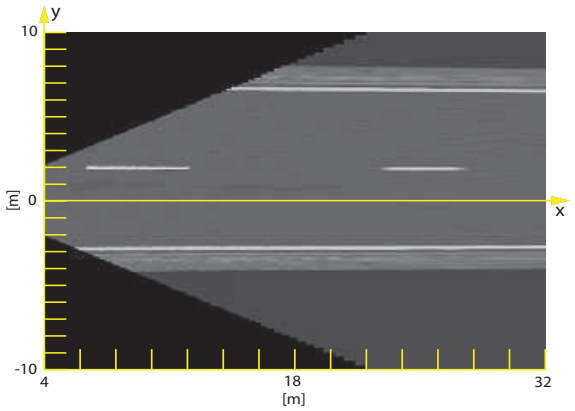

Figure 10. Inverse Perspective.

Edge Detection and Line Identification – The task of this phase is that of identifying points in a digital image at which the image brightness changes sharply or more formally has discontinuities. For instance, a strip may be distinguished from the asphalt by means of the associated intensity changes. The ultimate goal of the edge detection is the characterization of significant intensity changes in the digital image in terms of edge points.

To this end let us denote with $IPM(x, y)$ the gray-scale image (see Figure 10). An edge point is defined as the zero crossing of the Laplacian of the function $IPM(x, y)$ [32] (see Figure 12)

$$\begin{aligned} L(x,y) = &\ \nabla^2 IPM(x,y) = \\ = &\ \frac{\partial^2 IPM(x,y)}{\partial x^2} + \frac{\partial^2 IPM(x,y)}{\partial y^2} \end{aligned} \quad (21).$$

The intensity changes can be identified by using the above Laplacian operator. However, because the computation of $L(x, y)$ is highly sensitive to image noises unacceptable errors could arise. For this reason, a well-known procedure consists of first convolving the function $IPM(x, y)$ with a smoothing two-dimensional Gaussian filter of the following form

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}}, \ \sigma \text{ the standard deviation,} \quad (22).$$

and then applying the Laplacian operator to the obtained result.

Here the idea is to reverse such steps thanks to the linearity properties of ∇^2 : first we compute $\nabla^2 G(x, y)$, then the result is convolved with $IPM(x, y)$. The main reason of such a choice is that it allows one to off-line compute $\nabla^2 G(x, y)$ and to on-line use low-demanding filters.

To reduce further the computational burden, we select the class of steerable filters introduced by Freeman and Adelson [26]. Such filters can be rotated very efficiently by taking a suitable linear combination of a small number of filters. Steerable filters have a number of desirable properties that make them excellent for lane detection applications.

The steerable filters used here are based on second derivatives of the Gaussian filter (22), where

$$\begin{aligned} G_{xx}(x,y) &= \frac{(x^2-\sigma^2)}{2\pi\sigma^6} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \\ G_{xy}(x,y) &= \frac{xy}{2\pi\sigma^6} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \\ G_{yy}(x,y) &= \frac{(y^2-\sigma^2)}{2\pi\sigma^6} e^{\frac{-(x^2+y^2)}{2\sigma^2}} \end{aligned} \quad (23).$$

are the second order derivative filter kernels which can computed off-line and separated into their x and y components. At each time instant, the filters are convolved with the gray-scale image $IPM(x,y)$ to get its derivatives, i.e. D_{xx}, D_{xy}, D_{yy}. The next step is to build the following binary matrix

$$IPM_b(x,y) = \begin{cases} 1, & L(\bar{x},\bar{y}) < \lambda_{th}\, min(L(x,y)) \\ 0, & otherwise \end{cases} \quad (24).$$

where $(\bar{x},\bar{y})$ are the coordinates of a generic pixel and λ_{th} represents a threshold used to discriminate the edge pixels. The matrix $IPM_b(x,y)$ is used to appropriately select the edge pixels (1-entries) on the image $IPM(x,y)$.

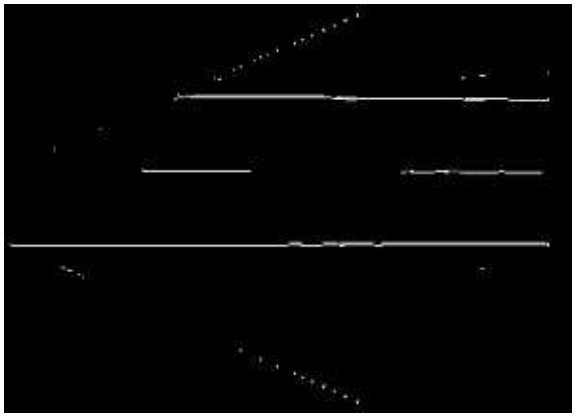

Figure 11. Edge Detection.

Amongst all the edge pixels, only the stripes need to be detected. Therefore, an additional filtering phase is necessary. In particular, the $\nabla^2 IPM(x,y)$ w.r.t. any angle orientation is defined as follows

$$\begin{aligned} \nabla^2 IPM^{\theta}(x,y) := \quad & D_{xx}\cos^2(\theta) + D_{yy}\sin^2(\theta) \\ & -2\, D_{xy}\cos(\theta)\sin(\theta) \end{aligned} \quad (25).$$

and we want to determine all pixels (x,y) at which the gradient of the Gaussian $\nabla^2 IPM^{\theta}(x,y)$ along the direction perpendicular to the stripe assumes a maximum value. This can be achieved by computing

$$\begin{aligned} \theta_{max} &= \tan^{-1}\left(\frac{D_{xx}-D_{yy}+\xi}{2D_{xy}}\right) \\ \xi &= \sqrt{D_{xx}^2 - 2D_{xx}D_{yy} + D_{yy}^2 + 4D_{xy}^2} \end{aligned} \quad (26).$$

Finally, by moving the search along the maximum directions, the stripe pixels selection is performed by searching for zero crossing of $L(x,y)$ [27] (see Figure 11).

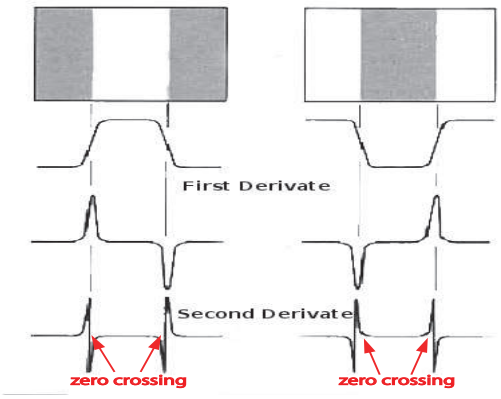

Figure 12. Zero crossing.

Line Fitting – In this phase, we resort to a simple parabolic road model [28] which is a sufficiently accurate approximation of the clothoid model usually used in civil engineering [29]. Therefore, each stripe can be simply described by the following quadratic function

$$y(x) = c + bx + ax^2 \quad (27).$$

where y and x represent the physical coordinates as depicted in Figure 10 while the sign of the constant c depends on which line is taken into consideration w.r.t. the optical axis x (see Figure 13: positive values of c correspond to the red line while negative ones to the green line).

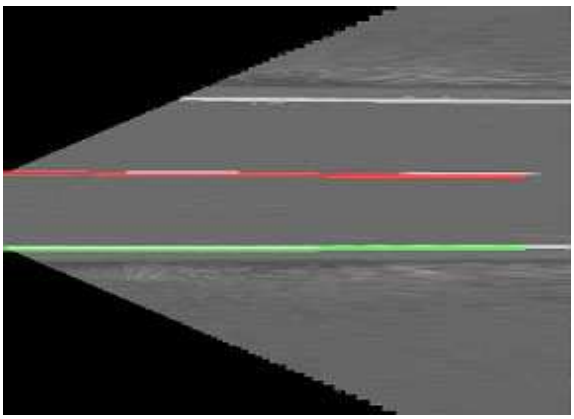

Figure 13. Line Fitting.

Here, for curve fitting purposes, we apply a well-established algorithm known as RANdom SAmple Consensus procedure (**RANSAC**) [30]. The **RANSAC** is a robust fitting algorithm that has been successfully applied in several computer vision problems [31]. The algorithm consists in an iterative procedure to estimate the unknown parameters of a given mathematical model using a set of measured data. It can be considered non-deterministic in the sense that it produces reasonable results within a pre-specified probability. **RANSAC**, as opposite to the conventional smoothing techniques, uses as small initial data sets as feasible and enlarges these sets with consistent data as much as possible.
The paradigm can be more formally stated as follows.

1. Given a model $\mathcal{M}$, which requires a minimum of n data points to instantiate its free parameters, a set $\mathcal{D}$ of data points such that $card(\mathcal{D}) \geq n$ and a number N_{max} of trials;
2. Randomly select a subset $\mathcal{S}_i$ of n data points from $\mathcal{D}$ and instantiate the model $\mathcal{M}$ as $\mathcal{M}_i$;
3. Determine a subset $\mathcal{S}_i^* \subset \mathcal{D}$ of data points such that it satisfies a fixed tolerance error w.r.t. $\mathcal{M}_i$. $\mathcal{S}_i^*$ is defined as the consensus set of $\mathcal{M}_i$;
4. If $card(\mathcal{S}_i^*) \geq N_{th}$ (a given threshold which is a function of the number of data points of $\mathcal{D}$ not considered in $\mathcal{S}_i^*$) and $i < N_{max}$, use $\mathcal{S}_i^*$ to generate a new instantiate model $\mathcal{M}_{i+1}$, $i := i + 1$ and goto the step 3.;
5. Else if $card(\mathcal{S}_i^*) < N_{th}$ and $i < N_{max}$, $i := i + 1$ and goto the step 2.;
6. Else if $i = N_{max}$,
   - if $card(\mathcal{S}_{N_{max}}^*) \geq N_{th}$ then use $\mathcal{M}_{N_{max}}$
   - otherwise consider that model $\mathcal{M}_i$ such that $card(\mathcal{S}_i^*)$ is maximal.

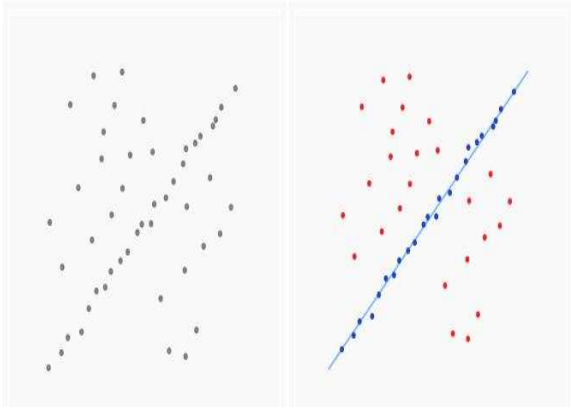

Figure 14. Ransac Fitting.

Lane Tracking

The second phase of the proposed vision system consists of the development of a Lane Tracking algorithm. The elaborations here take care of data coming from different video frames and try to make consistent quantitative conclusions on how the lane changes during the vehicle motion. To this end, we will use a Kalman Filter (KF) [22] in order to estimate and update the coefficients $(a,\ b,\ c)$ of the line model (27) during the vehicle motion. Therefore, we consider the following linear time-invariant system

$$\begin{aligned} x_k &= Ax_{k-1} + w_{k-1} \\ z_k &= Hx_k + v_k \end{aligned} \quad (28).$$

where the state is

$$x = [a\ \ b\ \ c\ \ \Delta a\ \ \Delta b\ \ \Delta c]^T \quad (29).$$

with $\Delta x_k := x_k - x_{k-1}$, w and v are zero mean white noises with covariance matrices Q and R respectively, and

$$A = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Following the standard notation, at each time step k the state estimation is given by

$$\hat{x}_k = \bar{x}_k + K_k(z_k - H\bar{x}_k) \quad (30).$$

with

$$\begin{aligned} \bar{x} &= A\hat{x}_{k-1} && (31). \\ \bar{P}_k &= A\hat{P}_{k-1}A^T + Q && (32). \\ \hat{P}_k &= (I - K_k H)\,\bar{P}_k && (33). \\ K_k &= \bar{P}_k H^T \left(H\bar{P}_k H^T + R\right)^{-1} && (34). \end{aligned}$$

where the above iteration is initialized with $\hat{x}_0 = 0$ and a covariance matrix $\hat{P}_0$ is appropriately chosen as indicated in [22].

SIMULATION RESULTS

All the above software modules (EKF, Inverse perspective, Steerable filters, **RANSAC** and KF) have been implemented within the Matlab/Simulink® package. Simulations have been carried out by using video and sensors data provided by the Carsim® simulator.
Simulations were conducted to estimate key vehicle parameters and to validate the models used in vehicle dynamics simulation. Accurate knowledge of the parameters is useful for system design, for evaluation of results in simulation, and for on-board use, in estimating the vehicle and roadway states and to compute the TLC time.
We have considered the following simulation scenario.
Double lane crossing - *While the vehicle is proceeding along a straight road with a longitudinal velocity of* $90\,Km/h$*, it moves from the right lane to the left one with a constant lateral velocity of* $0.31\,m/s$ *during the time interval* $[11,\ 26]\,sec$*. Then, it remains on the left lane until time instant* $35\,sec.$ *and finally the vehicle changes again the lane in the time interval* $[35,\ 50]\,sec.$
Numerical results are reported in next Figures 15-18. First, we have compared the estimates of the lateral position, lateral velocity and yaw angle against the exact data provided by Carsim. As it clearly results, the proposed EKF is able to accurately estimate such vehicle kinematical parameters. In particular, Figure 15 shows that the proposed procedure allows one to accurately identify the lane boundaries (with a relative mean error around 2%) while a larger relative error (around 10%)

is detected when the car crosses the line. However, this will not influence the successive TLC computation. The latter is mainly due to the fact that in such a case the EKF has to be updated for recovering the new lane position: this operation leads to a certain degree of loss of tracking because the EKF has to acquire at least ten frames of the new lane for a more accurate identification. Similar remarks apply for the lateral velocity and yaw angle estimates (see Figures 16, 17), even if a larger discrepancy (relative mean error around 5%) w.r.t. the exact values arises. However, such estimates are still consistent even for relatively large steer motions.

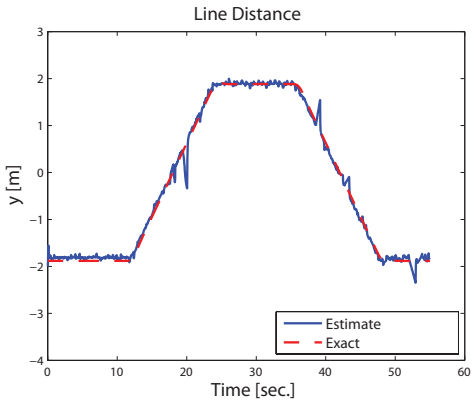

Figure 15. Lateral Position estimation

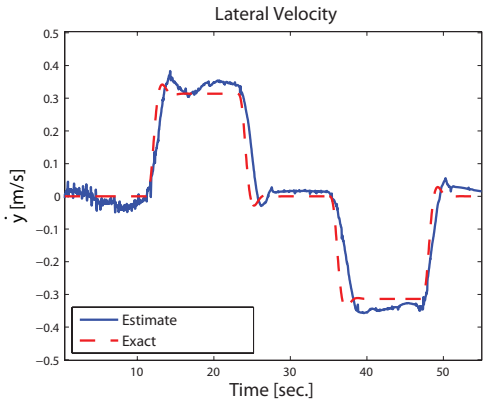

Figure 16. Lateral Velocity estimation

Finally, Figure 18 depicts the TLC computation by means of the predictive approach described in the previous sections. First, the TLC computation has been saturated (in software) at five seconds if larger values result. There, it is assumed that a lane departure warning is issued if TLC is lower than 1.5 seconds. As it is evident from the figure, the TLC estimate is sufficiently accurate. In fact, the relative mean error w.r.t. the exact curve (dashed line) is approximately around 5%.
Next simulations are instrumental to show the capability of the proposed approach to avoid false alarms for drivers who hug one side of the lane.
For comparisons purposes, we have contrasted the proposed approach with the recently proposed no-predictive method described in [9], [10]. By referring to the scheme depicted in Figure 6, the main differences of such a strategy w.r.t. the proposed LDWS can be summarized as follows:

- **Lane detection -**
  - *Image preprocessing*: each single frame is first converted to a gray-scale picture, then the bottom region under the frame horizon, named Region of Interest (ROI), of the image is selected [28] for the next steps;
  - *Edge detection*: a Sobel filter [32] is applied to each single ROI;
  - *Line identification*: this task is achieved by resorting to the Hough Transform [32] that allows one to map each road line into an accumulator point of the Hough parameter space in the (ρ, θ) coordinates;
  - *Line fitting*: a linear model of the road is used [9].
- **Lane Tracking -** A Kalman Filter is used in order to numerically identify the accumulator points $(\rho_i, \theta_i),\ i = 1, 2$ of the Hough parameter space which describe the lines of two adjacent and successive frames;

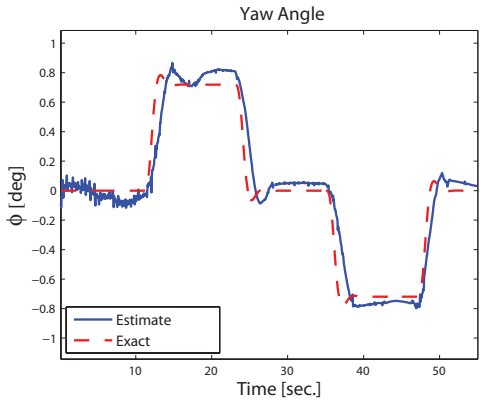

Figure 17. Yaw Angle estimation

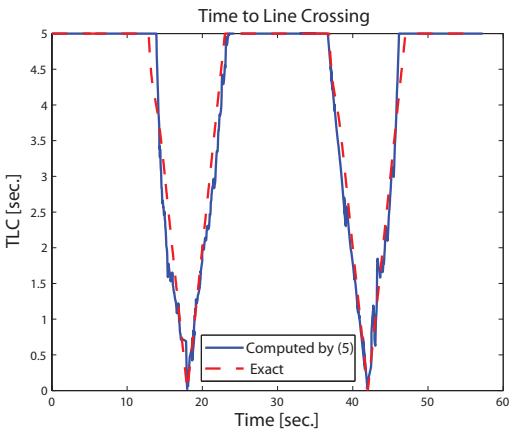

Figure 18. TLC computation

- **Lane Departure -** The numerical method is as follows. Let θ_l and θ_r the left and right orientation angles of the lane boundaries of a specified frame, see Figure 19. If the vehicle is traveling in a straight portion of the road and stays at the center of the lane we have $\theta_l + \theta_r \approx 0$, with $\theta_l < 0$ and $\theta_r > 0$. If the vehicle drifts to its left, both θ_l and θ_r increase, while if the vehicle drifts to its right, both θ_l and θ_r decrease. Thus, a simple and efficient measure for trajectory deviations is given by

$$\beta = |\theta_l + \theta_r| \qquad (35).$$

If β gets sufficiently large, the vehicle is leaving the center of the lane. In practice, β is compared to a threshold T, and a lane departure warning is issued if $\beta > T$.

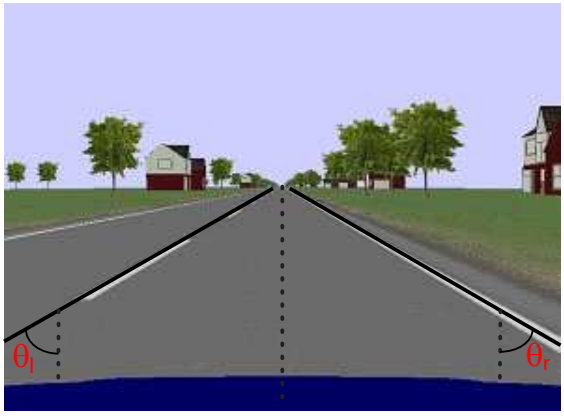

Figure 19. Orientation of lane boundaries

Finally, Figure 20 reports the simple lane departure warning activation scheme of [9], [10] with the threshold T set to $T = 30^o$. Then, at each time instant the absolute sum of angles β is computed. Hence, if the numerical value $\beta(t)$ (continuous line) overcomes the threshold T (dashed-line) a warning (red circle) is instantaneously activated.

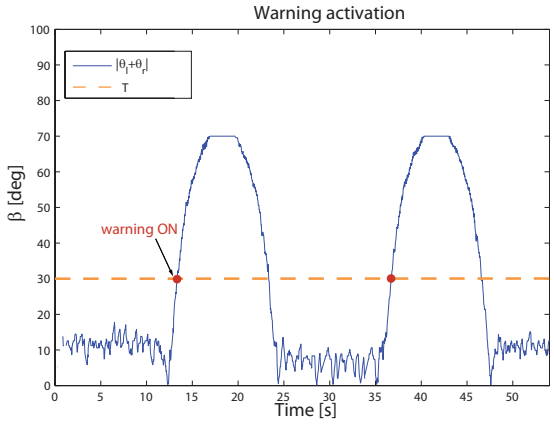

Figure 20. Departure warning method [9]

In the sequel we shall consider the following critical situation.

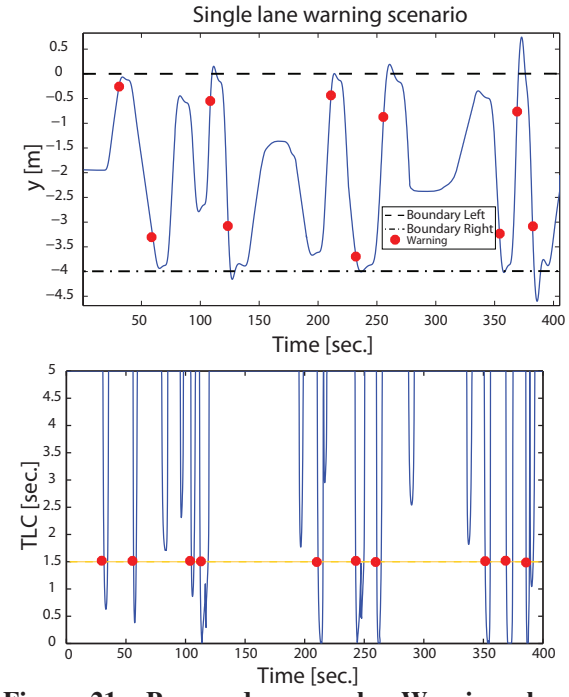

Figure 21: Proposed approach: Warning alarms (Up) TLC computation (Down)

Single lane warning scenario - *While the vehicle is proceeding along a straight road with a longitudinal velocity of* $70\,Km/h$*, it shows unintentional displacements from the center lane towards the right and/or left boundaries and viceversa with a varying lateral velocity* $\dot{y} \in [-0.5,\ 0.5]\,m/s$.
This scenario simulates situations when drivers are sleepy or drowsy driving and they are not capable to adequately conduct the vehicle.
Figure 21 depicts the sequence of warnings computed by the proposed LDWS strategy. The dashed and dot-dashed lines describe the left and right lane boundaries respectively while the red circles represent the vehicle positions (distance from the boundary) when the warning is issued. Moreover, the TLC computation is also provided in the figure.
Figure 22 shows the warning events signaled by the no-predictive strategy [9]. In this case, besides the red circles which represent correct warnings, this strategy gives rise to some false alarms (green circles). The main reason for the latter relies upon the fact that the alarm is automatically activated when the vehicle stays at specified orientations with respect to the lane boundaries without taking care of the vehicle dynamics. In principle, one could increase the threshold to reduce the generation of false alarms but this would imply the activation of true alarms too late for any safe maneuver.

CONCLUSIONS

In this paper, the development of a TLC-based lane departure warning system has been presented. An on-board vision system has been used for collecting road

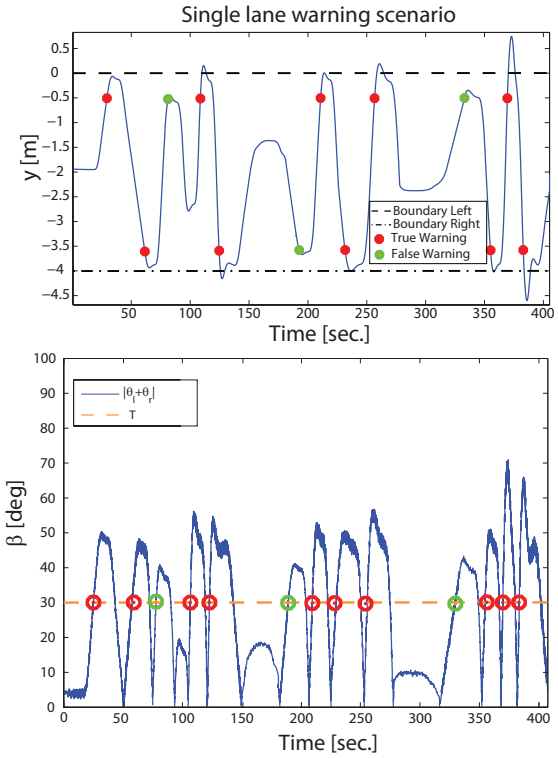

Figure 22: Approach of [9]: Warning alarms (Up) Warning activation (Down)

images and extracting useful features, relevant to identify the lane strips and compute the position and the heading of the vehicle with respect to the lane. Beside a single calibrated camera mounted behind the windshield, also steering angle and angular speed sensors are used to collect relevant kinematical data to be used in a model-based data-fusion strategy for the computation of the TLC and the generation of warnings about possible imminent lane departures.
Experimental results have shown good accuracy and robustness, w.r.t. road and weather conditions, in the estimation of the TLC. It has been also shown that the proposed LDWS system is able to reduce false alarms and increase, in comparison with traditional no model-based strategies, the time margins for warnings generation.

Future work will include the full development of a hw/sw demonstrator to be mounted in a commercial car to verify the effectiveness of the concepts and the quality of the implemented strategies. Field tests will be also conducted in order to verify the drivers acceptance of this sort of equipments in terms of readiness and accuracy in generating lane departure warnings or possible rejections due to a too high interference into their normal driving habits.

ACKNOWLEDGEMENT

This work has been partially supported by Orangee s.r.l., under the contract n. 1861, 22 November 2007 entitled: *Algoritmi e metodi per il riconoscimento tempo-reale della corsia percorsa tramite visione artificiale e della stima del tempo di invasione di corsia attigua*.

REFERENCES

[1] *Annual Statistical Report 2008*, European Road Safety Observatory (ERSO), 2008.

[2] *Incidenti Stradali - Anno 2007*, Istituto Nazionale di Statistica (ISTAT), 2008.

[3] *i2010 Intelligent Car brochure*,http://ec.europa.eu/inteligentcar, 2009

[4] P. Batavia, *Driver-adaptive lane departure warning systems*, Ph.D. Dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 1999.

[5] D. J. LeBlanc, G. E. Johnson, and P. J. Th. Venhovens, et al., *CAPC: an implementation of a road-departure warning system*, Proceedings of IEEE International Conference on Control Application, pp. 590-595, 1996.

[6] Y. Wang, E.K. Teoh, D. Shen, *Lane detection and tracking using B-Snake*, Image and Vision Computing 22, pp.269-280, 2004.

[7] J. C. McCall, M. M. Trivedi, *Video-Based Lane Estimation and Tracking for Driver Assistance: Survay, System and Evalutation*, IEEE Trans. on Intelligent Transportation Systems, Vol. 7(1), 2006.

[8] M. Bertozzi, A. Broggi, *GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection*, IEEE Trans. on Image Processing, Vol. 7(1), pp.62-81, 1998.

[9] J. W. Lee, *A machine vision system for lane-departure detection*, Computer Vision and Image Understanding 86, pp. 52-78, 2002.

[10] J.W. Lee, U.K. Yi, *A lane-departure identification based on LBPE, Hough transform, and linear regression*, Computer Vision and Understanding 99, pp.359-383, 2005.

[11] H. Godthelp, P. Milgram, and G. J. Blaauw, *The development of a time-related measure to describe driver strategy*, Human Factors, 26, pp. 257-268, 1984.

[12] M. S. Grewal, A. P. Andrews, *Kalman Filtering: Theory and Pratice Using Matlab*, 2nd Ed., Wiley, 2001.

[13] W. van Winsum, K.A. Brookhuis, D. de Waard, *A comparison of different ways to approximate time-to-line crossing (TLC) during car driving*, Accident Analysis and Prevention 32, pp. 47-56, 2000.

[14] S. Mammar, S. Glaser, M. Netto, *Time to lane crossing for lane departure avoindace: A theoretical study and experimantal setting*, IEEE Trans. on Intellingent Transportation Systsems 7(2), pp. 226-241, 2006.

[15] Farrelly J., Wellstead P., *Estimation of Lateral velocity*, Proceeding of the 1996 IEEE International conference on Control Applications, Dearborn, pp. 552-557, 1996.

[16] A. Y. Ungoren, H. Peng, H.E. Tseng, *A Study on Lateral Speed Estimation Methods*, Int. J. Vehicle Autonomous Systems,Vol. 2, Nos 1/2, pp. 126-144, 2004.

[17] U. Kiencke, A. Daib, *Observation of Lateral Vehicle Dynamics*, Control Eng. Practice, Vol. 5, No. 8, pp. 1145-1150, 1997

[18] C. Lin, A. G. Ulsoy, D. J. LeBlanc, *Vehicle Dynamics and External Disturbance Estimation for Vehicle Path Prediction*, IEEE TRANS. on Control System Technology, Vol. 8(3), 2000.

[19] S. Clark, *Autonomous Land Vehicle Navigation using Millimetre Wave Radar*, PhD Thesis Department of Mechanical and Mechatronic Engineering, University of Sydney, 1999.

[20] H. W. Sorenson, *Kalman Filtering: theory and application*. IEEE Press, 1985.

[21] G. T. Schmidt, *Pratical Aspects of Kalman Filtering Implementation*, AGARD-LS-82, NATO Advisory Group for Aerospace Research and Development, 1976.

[22] D. E. Catlin, *Estimation, Control and the Discrete Kalman Filter*, In Applied Mathematical Sciences 71, Springer-Verlag, 1989.

[23] A. M. Muad, A. Hussain, S. A. Samad, M. M. Mustaffa, B. Y. Majlis, *Implementation of inverse perspective mapping algorithm for the development of an automatic lane tracking system*, TENCON 2004. 2004 IEEE Region 10 Conference (A),pp.207-210, 2004.

[24] A. Broggi, *An Image Reorganization Procedure for Automative Road Following Systems*, Proc. International Conference on Image Processing, Vol. 3. 1995. pp. 532-535, 1995.

[25] D. H. Ballard, C.M Brown, *Computer Vision*, Prentice Hall, New Jersey, 1982.

[26] Freeman, Adelson, *The design and use of steerable filters*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 13(9), pp.891-906, 1991.

[27] M. Jacob, M. Unser, *Design of Steerable Filters for Feature Detection Using Canny-Like Criteria*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 26(8), pp.1007-1019, 2004

[28] C.R. Jung,C.R. Kelber, *A lane departure warning system based on a linear-parabolic lane model*, Proceedings of IEEE Intelligent Vehicles Symposium, pp.891-895, Parma, Italy, 2004.

[29] K.G. Baass, *The use of clothoid templates in highway design*, Transportation Forum, Vol. 1, pp. 47-52, 1987.

[30] M. A. Fischler, R. C. Bolles, *Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography*, Comm. of the ACM, Vol. 24, pp. 381395, 1981.

[31] Zu Kim, *Realtime Lane Tracking of Curved Local Road*, Proceedings of the 2006 IEEE Intelligent Transportation Systems Conference Toronto, pp. 1149-1155, 2006.

[32] R. C. Gonzalez, R. E. Woods, *Digital Image Processing*, 2nd Ed. Prentice Hall, 2002.