Data Fusion Algorithms for Lane Departure Warning Systems

Gianni Cario, Alessandro Casavola, Giuseppe Franzè, Marco Lupia

Abstract—This paper discusses data-fusion algorithms for computing the “Time-to-Lane-Crossing” (TLC) of a vehicle traveling along a lane on the basis of road images, collected by an onboard video-camera, and kinematic data coming from car sensors. Such algorithms are instrumental to built Lane Departure Warning Systems (LDWS) with enhanced predictive capabilities and tolerant to a certain amount of errors and outliers in video-camera and sensors data flows. Two Kalman filters are outlined and used for data fusion purposes. Simulations within the Carsim® simulation tool have been undertaken to verify the effectiveness of these filters during critical driving scenarios.

I. 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 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. Amongst many, we focus here on the development of Lane Departure Warning Systems (LDWS) which, according to a recent report of the EU Intelligent Car Initiative [2], 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%.

LDWS refers to systems that try to help the driver to stay into the lane. A microcontroller, such as Digital Signal Processing (DSP) units, equipped with an on-board camera is typically used to identify the lane stripes. Some interesting contributions to LDWS development can be found in [3]. An interesting approach is the so-called TLC-based method, first proposed by Godthelp et al. [4], where an alarm is triggered when the TLC is below a specified threshold. In general, TLC-based methods provide earlier warnings than roadside rumble stripes (RRS), because alarms are triggered with sufficient advance before the driver being really in danger.

In the design of this kind of vehicular active safety systems, Data Fusion algorithms [5], [6] are going to play a key role. The main reason is that higher reliability and fault tolerance properties to sensor faults can be achieved by combining information coming from different sensors.

In this paper, we propose a TLC-based LDWS system, where a single calibrated camera has been used for capturing road images and data fusion algorithms has been implemented for determining the lane markings and estimating the TLC time. Specifically, for data fusion purposes, we have considered two kinds of Kalman filters well suited for nonlinear dynamics: the Extended Kalman Filter (EKF) [7] and the Unscented Kalman Filter (UKF) [8].

In order to verify the efficiency of the proposed LDWS, some critical driving scenarios have been conceived with the aid of a well-known vehicle simulator, the Carsim® [9]. The simulations are particularly useful to compare the effectiveness of the EKF and UKF filters when anomalous driving behaviors or camera occlusions events occur.

II. SYSTEM OVERVIEW

In this section we will give an overview of the LDWS under development. Fig. 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 and angular speed sensors mounted on rear wheels.

In Fig. 2, the overall functional and computational scheme for the proposed LDWS 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, not 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 vehicle model by using a suitable Data Fusion algorithm. In this paper, this phase will be accomplished by using two algorithms: Extended Kalman Filter [7] and Unscented Kalman Filter [8].

The Warning Generation module is in charge of generating an alarm whenever necessary on the basis of information coming from the Data Acquisition and Elaboration module and 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.

![System Overview](image)

Fig. 2. System Overview.

### III. TIME TO LANE 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 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 [10],[11] 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 practice, the computation of the TLC is performed by using an approximation procedure based 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

Then, the TLC can be easily computed as the ratio between the lateral position and the rate of change of the lateral position [10]

$$TLC = \frac{Y}{\dot{Y}}$$

(1)

Here the lateral velocity $\dot{Y}$ is obtained by means of a Data Fusion algorithm that, by using the mathematical vehicle model, allows to estimate the vehicle position ($Y$) and the yaw angle ($\phi$)). Specifically, we have

$$\dot{Y} = V \sin \phi$$

with $V$ the vehicle velocity.

It is worth to note that, even if the assumption b) is not realistic, in [10] the expression (1) has been proved to be a tight overestimation of the minimum TLC, whose accuracy increases as the time to cross the lane decreases.

### IV. DATA FUSION

This section is devoted to describe the proposed Kalman filters for lateral speed estimation purposes. In the sequel, we will first discuss the mathematical vehicle model, then the Extended Kalman Filter and the Unscented Kalman Filter will be outlined.

#### A. Vehicle Model

A vast variety of mathematical models able to describe the vehicle dynamics during driving have been proposed in the literature (see [12] 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 [13]. 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 [14].

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 $\phi$. Hereafter, $V_{RL}$ and $V_{RR}$ denote the longitudinal velocity of the rear left and rear right wheel respectively, $V$ the longitudinal velocity of the vehicle and $B$ the distance between rear wheels. Then, a continuous-time description can be derived as follows:

$$\begin{align}
\dot{x}(t) &= V(t) \cos(\phi(t)) \\
\dot{y}(t) &= V(t) \sin(\phi(t)) \\
\dot{\phi} &= \frac{V_{RR}(t) - V_{RL}(t)}{B}
\end{align}$$

(2)

with $V(t) = \frac{V_{RL}(t) + V_{RR}(t)}{2}$.
The continuous-time system (2) can be discretized using forward Euler differences with a sampling time $\Delta T$. As a result, the following discrete-time description is achieved

\[
\begin{bmatrix}
x(k)
y(k)
\phi(k)
R(k)
\end{bmatrix} = \begin{bmatrix}
x(k-1) + V(k)\Delta T \cos(\phi(k-1)) \\
y(k-1) + V(k)\Delta T \sin(\phi(k-1)) \\
\phi(k-1) + \frac{V(k)}{R(k)} \Delta T \\
R(k-1)
\end{bmatrix}
\]

(3)

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. Therefore in order to compensate the possible occurrence of such effects, a wheel radius $R(k)$ component can be added to the state description and the following four state model can be proposed:

\[
\begin{bmatrix}
x(k)
y(k)
\phi(k)
R(k)
\end{bmatrix} = \begin{bmatrix}
x(k-1) + \omega_R(k) R(k-1) \Delta T \cos(\phi(k-1)) \\
y(k-1) + \omega_R(k) R(k-1) \Delta T \sin(\phi(k-1)) \\
\phi(k-1) + \frac{\omega_R(k) R(k-1) \Delta T}{B} \\
R(k-1)
\end{bmatrix} + \begin{bmatrix}
\varepsilon_x(k) \\
\varepsilon_y(k) \\
\varepsilon_\phi(k) \\
\varepsilon_R(k)
\end{bmatrix}
\]

(4)

where $R(k) \in \mathbb{R}$, $\omega_R(k)$ and $\omega_L(k)$ are the forward wheel angular velocities measured by the wheel sensors. Moreover, the additive vector $[\varepsilon_x(k), \varepsilon_y(k), \varepsilon_\phi(k), \varepsilon_R(k)]^T$, 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.

### B. Extended Kalman filter

The Kalman Filter (KF) 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) [7] 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

\[
x_k = f_k(x_{k-1}) + w_{k-1},
\]

(5)

\[
z_k = h_k(x_k) + R_k
\]

(6)

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, i.e. $w_k \sim \mathcal{N}(0, Q_k)$, $v_k \sim \mathcal{N}(0, R_k)$. 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 estimated value of the random vector is the one that minimizes the cost function

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

(7)

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 (5)

\[
\Phi_k = \frac{\partial f_k}{\partial x} |_{x=\hat{x}_{k-1}}
\]

(8)

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

\[

\begin{align*}
P_{k|k-1} &= \Phi_k P_{k-1} \Phi_k^T + Q_k, \\
\hat{x}_{k|k-1} &= f_k(\hat{x}_{k-1})
\end{align*}
\]

(9)

Then, given the current measurement $z_k$ and by linearizing the output equation (6) according to

\[
H_k = \frac{\partial h_k}{\partial x} |_{x=\hat{x}_{k|k-1}}
\]

(10)

the following Kalman observer gain is derived

\[
K_k = P_{k|k-1} H_k^T (R_k + H_k P_{k|k-1} H_k^T)^{-1}
\]

(11)

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

\[
\begin{align*}
\hat{x}_k &= \hat{x}_{k|k-1} + K_k (z_k - h_k(\hat{x}_{k|k-1})), \\
P_k &= (I - K_k H_k) P_{k|k-1}
\end{align*}
\]

(12)

(13)

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 (4), 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.

### C. Unscented Kalman filter

It is well known that the Extended Kalman filter could give rise to poor estimation performance when the plant model (5)-(6) has an highly nonlinear structure, because it does work on a linearized model of the nonlinear state space description. Such a drawback can be avoided by using a further extension of the Kalman filter: the Unscented Kalman Filter (UKF) [8].

The UKF is based on a different idea: the mean and the covariance matrix are updated by resorting to a deterministic sampling technique, the unscented transform, whose aim is to select an appropriate minimal set of sample points (sigma points). Hence, all the sigma points are propagated through the state transition function $f_k(\cdot)$ and the observation function $h_k(\cdot)$, from which both the mean and the covariance matrix of the state estimate are finally recovered.
In particular, the sigma points are selected as follows
\[
(X_{k-1}^{(0)}) = \hat{x}_{k-1}
\]
\[
(X_{k-1}^{(m)}) = \hat{x}_{k-1} + \text{col}_m \left( \sqrt{(n + \lambda)P_{k-1}} \right), \quad m = 1, \ldots, n
\]  
(14)
\[
(X_{k-1}^{(m)}) = \hat{x}_{k-1} + \text{col}_{m-n} \left( \sqrt{(n + \lambda)P_{k-1}} \right), \quad m = n+1, \ldots, 2n
\]
where \( n = \text{dim}(x_k) \), and \( \lambda = \alpha^2(n + \chi) - n \), with \( \alpha \) and \( \chi \) scaling factors.
Essentially the EKF algorithm consists of two phases. In the first step the sigma point predictions of the state and of the covariance error matrix are computed by
\[
\hat{x}_{k|k-1} = \sum_{m=0}^{2M} \omega(m) X_k^{(m)}
\]
\[
P_{k|k-1} = \sum_{m=0}^{2M} \Omega(m) (X_k^{(m)} - \hat{x}_{k|k-1})(X_k^{(m)} - \hat{x}_{k|k-1})^T + Q_{k-1}
\]
where \( X_k^{(m)} = f_k(X_{k-1}^{(m)}) \). Then, a correction phase is performed by using the following equations:
\[
\hat{x}_k = \hat{x}_{k|k-1} + K_k (Z_k - \hat{z}_k^{(m)})
\]
\[
P_k = P_{k|k-1} - K_k S_k K_k^T
\]
where
\[
Z_k^{(m)} = h_{k-1}(X_k^{(m)}), \quad \hat{z}_k = \sum_{m=0}^{2M} \omega(m) Z_k^{(m)}
\]
\[
S_k = \sum_{m=0}^{2M} \Omega(m) (Z_k^{(m)} - \hat{z}_k)(Z_k^{(m)} - \hat{z}_k)^T + R_k
\]
\[
K_k = \sum_{m=0}^{2M} \Omega(m) (X_k^{(m)} - \hat{x}_{k|k-1})(Z_k^{(m)} - \hat{z}_k)^T S_k^{-1}
\]
with \( \omega^{(0)} \) and \( \Omega^{(0)} \), fixed weights selected as follows
\[
\omega^{(0)} = \frac{1}{(n+\lambda)^2}, \quad \omega^{(m)} = \frac{1}{2(n+\lambda)}, \quad m = 1, \ldots, 2n
\]
\[
\Omega^{(0)} = \frac{1}{(n+\lambda)^2}, \quad \Omega^{(m)} = \frac{1}{2(n+\lambda)}, \quad m = 1, \ldots, 2n
\]
where \( \beta \) is a parameter used to incorporate any prior knowledge about the distribution of the state \( x_k \).

V. Vision System

This section is devoted to describe the proposed vision algorithm. Two main phases can be characterized: Lane Detection and Lane Tracking.

A. Lane Detection

The lane detection system consists of all steps related to each frame elaboration in extracting relevant features and it includes five steps that will be discussed in details below.

1) 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.

2) 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 of a 3D object in a new planar image 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) [15].

3) 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 stripe may be distinguished from 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. An edge point is defined as the zero crossing of the Laplacian of the function \( IPM(x, y) \) [16]
\[
L(x, y) = \nabla^2 IPM(x, y) = \frac{\partial^2 IPM(x, y)}{\partial x^2} + \frac{\partial^2 IPM(x, y)}{\partial y^2}
\]
(18)
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 the following smoothing two-dimensional Gaussian filter of the following form
\[
G(x, y) = \frac{1}{2\pi\sigma^2}e^{-\frac{(x^2+y^2)}{2\sigma^2}}, \quad \sigma \text{ the standard deviation, } (19)
\]
and then applying the Laplacian operator to the obtained result.

The next step is to build the following binary matrix
\[
IPM_b(x, y) = \begin{cases} 
1, & L(x, y) < \lambda_{th} \min(L(x, y)) \\
0, & \text{otherwise} 
\end{cases}
\]
(20)
where \((x, y)\) are the coordinates of a generic pixel and \( \lambda_{th} \) 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 \( IPM(x, y) \).

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
\[
\nabla^2 IPM^{\theta}(x, y) := D_{xx} \cos^2(\theta) + D_{yy} \sin^2(\theta) - 2D_{xy} \cos(\theta) \sin(\theta)
\]
(21)
where \( D_{xx}, D_{yy}, D_{xy} \) represent the derivatives of \( IPM(x, y) \) computed by means of steerable filters [17]. Then, 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
\[
\theta_{max} = \tan^{-1}\left(\frac{D_{xx} - D_{yy} + \xi}{2D_{xy}}\right)
\]
\[
\xi = \sqrt{D_{xx}^2 - 2D_{xx}D_{yy} + D_{yy}^2 + 4D_{xy}^2}
\]  
(22)

Finally, by moving the search along the maximum directions, the stripe pixels selection is performed by searching for zero crossing of \(L(x, y)\) [17].

**Line Fitting:** In this phase, we resort to a simple parabolic road model [18] which is a sufficiently accurate approximation of the clothoid model usually used in civil engineering [19]. Therefore, each stripe can be simply described by the following quadratic function
\[
y(x) = c + bx + ax^2
\]  
(23)

where \(y\) and \(x\) represent the physical coordinates, while the sign of the constant \(c\) depends on which line is taken into consideration w.r.t. the optical axis \(x\). Here, for curve fitting purposes, we apply a well-established algorithm known as RANdom SAmple Consensus procedure (RANSAC) [20].

**B. 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) in order to estimate and update the coefficients \((a, b, c)\) of the line model (23) during the vehicle motion.

**VI. SIMULATION RESULTS**

All the above software modules (Inverse perspective, Steerable filters, RANSAC, KF, EKF and UKF) have been implemented within the Matlab/Simulink® package. Simulations have been carried out by using video and sensors data provided by the Carsim® simulator. The aim of the simulations is twofold: first the effectiveness of the proposed LDWS is proved, then we show that the Unscented Filter Kalman outperforms the Extended Filter Kalman when critical road scenarios are taken into consideration. To this end, the two Data Fusion algorithms (UKF and EKF) are contrasted, w.r.t. the exact data provided by Carsim, both in terms of the dynamical behavior (yaw angle and lateral position estimates) and the TLC time computation. Let us consider the following critical-driving scenario:

**Single lane scenario** - While the vehicle is proceeding along a straight road with the longitudinal velocity profile shown in Fig. 3, it unintentionally displaces from the center lane towards the right and/or left boundaries and viceversa with a varying lateral velocity \(\dot{y} \in [-6, 6] \text{ m/s}\).

Figs. 4-6 collect the numerical results achieved by using both the EKF and UKF strategies.

In particular, Figs. 4-5 contrast the estimates of the lateral positions and yaw angles w.r.t. the “true values” given by the Carsim simulator. As it clearly results, the UKF shows a good capability to adequately adapt to the critical conditions due to the abrupt changes occurring in the longitudinal velocity, while the EKF is not able to manage such events and gives rise to remarkable discrepancies. Such estimation errors affect directly the TLC computation (see Fig. 6). In fact, the TLC time computed by EKF is sometimes uncorrect.
The consequence is that false alarms could be generated at higher rates. On the contrary, UKF is much more robust to these anomalies and should be recommended in situations where high speeds and accelerations have to be considered.

However, note that the UKF shows an on-line computational burden remarkably worse than EKF. By using the profiler Matlab/Simulink routine, we can measure the total CPU time of each execution step for both methods: we got $T_{ukf} = 7.61 \text{ sec.}$ for UKF and $T_{ekf} = 2.55 \text{ sec.}$ for EKF.

Finally, we provide a further simulation to verify the effectiveness of the proposed LDWS strategy under camera occlusions due to other vehicles hiding the front road view or unfavorable light reflection or glare which can easily obscure part of the camera view. To this end, we have supposed that the data coming from the vision system are not available during some time interval. In Fig. 7, the lateral position under three occlusions is depicted. There it is shown that the LDWS system is still capable to manage such unpredictable events when a UKF data fusion algorithm is embedded (similar results are expected for EKF), while, if no data fusion methods are used as standard in actual commercial devices, the lateral position ($Y_{\text{true}}$) cannot be recovered from other sensors’ data, Fig. 7 (Top).

VII. CONCLUSIONS

In this paper, a TLC-based lane departure warning system has been presented. We have proposed a model-based data fusion strategy for the computation of the TLC and the generation of warnings about possible imminent lane departures. The vehicle has been described by means of a differential wheels kinematic model. Two Kalman filters (EKF and UKF) have been discussed and used for data fusion purposes. Simulations have been carried out by using the Carsim simulator with the aim to verify the effectiveness of both Kalman filters. The numerical results have shown that the UKF is less sensitive than EKF to critical conditions, such as abrupt lateral and longitudinal variations.

REFERENCES