sensors

Article

A Sensor Fusion Method Based on an Integrated Neural Network and Kalman Filter for Vehicle Roll Angle Estimation

Leandro Vargas-Meléndez [1], Beatriz L. Boada [1,*,†], María Jesús L. Boada [1,†], Antonio Gauchía [2,†] and Vicente Díaz [1,†]

[1] Mechanical Engineering Department, Universidad Carlos III de Madrid, Avda. de la Universidad 30, Madrid 28911, Spain; lvargas@ing.uc3m.es (L.V.-M.); mjboada@ing.uc3m.es (M.J.L.); vdiaz@ing.uc3m.es (V.D.)

[2] Mechanical Engineering-Engineering Mechanics Department, Michigan Tech University, 1400 Townsend Drive, Houghton 49931, Michigan, MI, USA; antonio@mtu.edu

[*] Correspondence: bboada@ing.uc3m.es; Tel.: +34-91-624-9168

[†] These authors contributed equally to this work.

Academic Editors: Xue-Bo Jin, Feng-Bao Yang, Shuli Sun and Hong Wei

Received: 11 July 2016; Accepted: 22 August 2016; Published: 31 August 2016

Abstract: This article presents a novel estimator based on sensor fusion, which combines the Neural Network (NN) with a Kalman filter in order to estimate the vehicle roll angle. The NN estimates a "pseudo-roll angle" through variables that are easily measured from Inertial Measurement Unit (IMU) sensors. An IMU is a device that is commonly used for vehicle motion detection, and its cost has decreased during recent years. The pseudo-roll angle is introduced in the Kalman filter in order to filter noise and minimize the variance of the norm and maximum errors' estimation. The NN has been trained for J-turn maneuvers, double lane change maneuvers and lane change maneuvers at different speeds and road friction coefficients. The proposed method takes into account the vehicle non-linearities, thus yielding good roll angle estimation. Finally, the proposed estimator has been compared with one that uses the suspension deflections to obtain the pseudo-roll angle. Experimental results show the effectiveness of the proposed NN and Kalman filter-based estimator.

Keywords: sensor fusion; roll angle estimation; neural network; linear Kalman filter

1. Introduction

Recent developments in vehicle technology have steered the industry towards an increase in vehicle safety, and it is now considered to be one of the key features of a vehicle, even at the initial design stages. One of the main causes of traffic accidents in which heavy vehicles are involved is the loss of lateral stability. Heavy vehicles are prone to roll over since the ratio of the height of the center of gravity to the wheel track is high. The loss of lateral stability is caused when the tire-road contact force on one of the wheels becomes zero under lateral acceleration.

Nowadays, vehicles are equipped with control systems in order to improve their handling and stability. Systems, such as Roll Stability Control (RSC) and Electronic Stability Control (ESC), need to know in advance the expected vehicle behavior during different maneuvers in order to adequately actuate on the vehicle systems [1,2].

RSC systems are based on lateral load transfer, which is directly related to the vehicle roll angle [3]. Therefore, to improve the performance of these systems, an accurate measurement of the vehicle roll angle is needed [4]. The roll angle can be obtained from a dual-antenna GPS, but this is a very expensive technique. For this reason, roll angle has to be estimated [5–7]. There are mainly two techniques used for its estimation: integration of information from sensor measurements (sensor fusion) and the usage of a physical vehicle model [8]. Sensor fusion technologies are widely used; they operate by integrating low-cost sensors in many vehicle applications. In principle, the fusion of multisensor data provides significant advantages over single source data. The use of multiple types of sensors may increase the accuracy with which a quantity can be observed and characterized [9]. Doumiati et al. [10] proposed a method to estimate the roll angle using measurements from potentially integrable sensors, such as accelerometers and suspension deflection sensors. If the pitch dynamic effects on roll motion are neglected, the roll angle can be calculated as:

$$\theta = \frac{(\Delta_{11} - \Delta_{12} + \Delta_{21} - \Delta_{22})}{2e_f} - \frac{m_v a_{ym} h}{k_t} \tag{1}$$

where Δ_{ij} is the suspension deflection, a_{ym} is the lateral acceleration, k_t is the roll stiffness resulting from tire stiffness and m_v is the vehicle weight. Nevertheless, suspension deflection sensors are expensive, so real-time measurement of the roll angle is typically not available for vehicles [11].

For this reason, different algorithms based on the fusion of other types of sensors are proposed. Low-cost GPS and onboard vehicle sensors are also employed [12,13]; however, satellite visibility might be poor in urban and forested driving environments, yielding inaccurate estimations [14].

Rajamani et al. [11] propose a dynamic observer that utilizes the information provided by only a lateral accelerometer and a gyroscope. Nevertheless, the estimation of roll angle has a significant error in transient response. In this algorithm, neither measurements noise nor model noise are considered. Other authors propose several sensor fusion techniques in combination with Kalman filters. A Kalman filter is frequently used for sensor fusion applications because it is an optimal estimator, is straightforward to implement and can be adapted to multiple sensor scenarios [15] and take into account the measurements and model noises. Common applications include vehicle localization [16], estimation of sideslip [17] and roll vehicle angle estimation [14,18].

In this paper, we propose a novel estimator based on a Neural Network (NN) combined with a Kalman filter in order to estimate the vehicle roll angle (see Figure 1). Previous work has combined AI-based techniques with a Kalman filter for estimation; however, in our case the IA-based algorithm is based on the improvement of filter performance through the adaptive estimation of the filter statistical information (covariance matrices) [19–21]. A difficulty is that uncertainty learning is a difficult and complex process. In this research, we not only estimate the filter statistical information, but also a "pseudo-parameter", a pseudo-roll angle, which is introduced in the Kalman filter. The NN system estimates a pseudo-roll angle through variables that are easily measured by an IMU, a device that has recently become less expensive. The pseudo-roll angle is introduced in the Kalman filter in order to filter out noise and minimize the variance of the estimated norm and maximum errors.

The rest of the paper is organized as follows. In Section 2, a description of the estimator architecture is given. The estimator architecture is formed by two modules: the NN module and the Kalman module. The former estimates a pseudo-roll angle, and the latter filters the noise and minimizes the norm and maximum errors. Section 3 presents the estimator results, both with simulated and experimental scenarios. A discussion of these results is presented. Finally, Section 4 concludes this paper.

2. Vehicle Roll Angle Estimator Architecture

The architecture of the proposed estimator is given in Figure 1. The architecture is formed by two modules: the NN module and the Kalman module. The NN module estimates a pseudo-roll angle through data, such as the lateral accelerometer signal, a_{ym}, the longitudinal accelerometer signal, a_{xm}, the yaw rate sensor signal, $\dot{\psi}_m$, and the roll rate sensor signal, $\dot{\phi}_m$. These signals are easily measured by an IMU, the cost of which has decreased in recent years. The pseudo-roll angle is fed into the Kalman module in order to filter noise and minimize the maximum and norm errors.

The proposed method has the advantage of taking into account the vehicle non-linearities, thus yielding a good roll angle estimation.

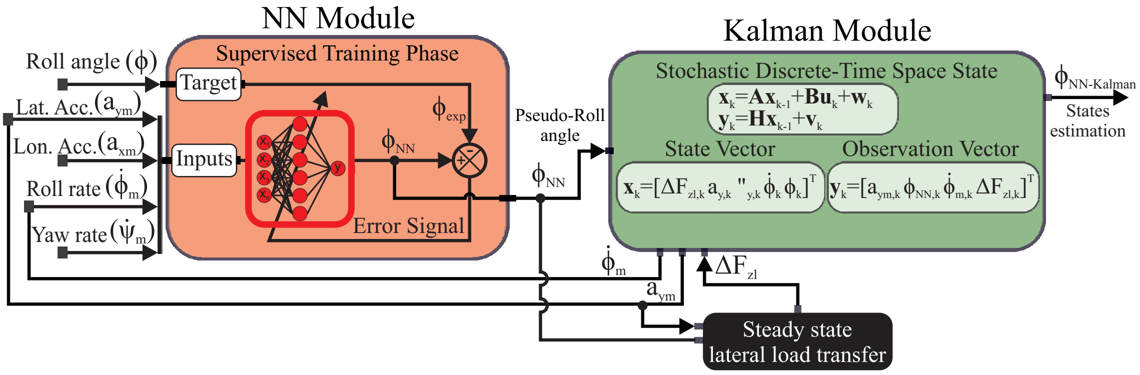


Figure 1. Estimator architecture.

2.1. NN Module

The NN module uses artificial neural networks to estimate roll angle. The preliminary reconstruction of the roll angle from an NN-observer is used as a "pseudo-measurement" in the Kalman filter. The proposed model employs a Back-Propagation (BP) algorithm, which is one of the most widely-used methods for training a neural network. The architecture of the BP neural network is shown in Figure 2. The NN has a single hidden layer with 15 neurons, four inputs (the lateral accelerometer signal, a_{ym}, the longitudinal accelerometer signal, a_{xm}, the yaw rate sensor signal, $\dot{\psi}_m$, and the roll rate sensor signal, $\dot{\phi}_m$) and one output (the vehicle roll angle, ϕ).

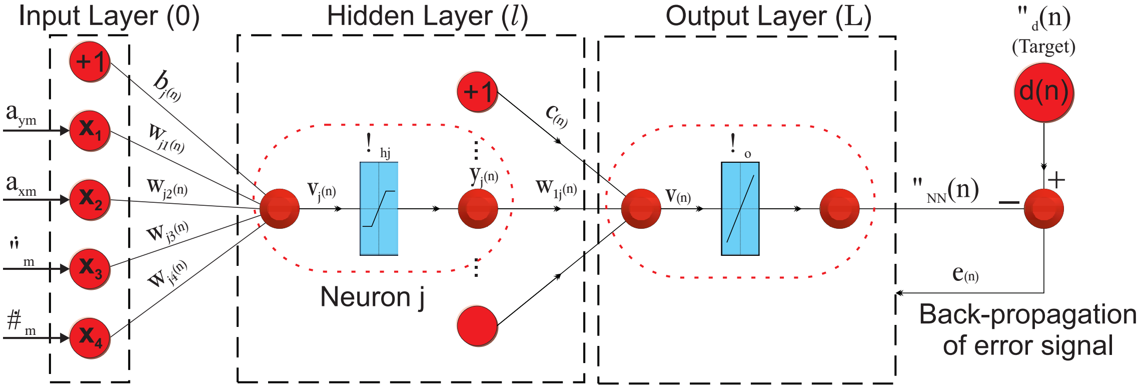


Figure 2. Neural network architecture.

The network parameters, synaptic weights and bias, are adjusted with the error signal. The error signal, e, is defined as the difference between the desired response, ϕ_d, and the estimated response of the network, ϕ_{NN}. The learning process is maintained on an epoch-by-epoch (an epoch is one complete presentation of the entire training set during the training process) basis, until the synaptic weights and bias levels of the network stabilize and the average squared error over the entire training set converges to some minimum value [22].

The sequential mode of training is divided in into five stages. The first stage, called initialization, employs random and small values close to zero for the weights and biases in both the hidden and the output layer. In the second stage, the training patterns $((\mathbf{x}(1), d(1)), \ldots, (\mathbf{x}(n), d(n))$ are presented to the NN. The vector $\mathbf{x}(n) = \{x_1(n), x_2(n), x_3(n), x_4(n)\}$ represents the input vector, and $d(n)$ represents the desired response, respectively, at iteration n. In the third stage, each hidden layer neuron calculates the sum of the weight inputs; it applies an activation function φ_{hj} and sends their results to the output layer. The output of the j-th hidden layer neuron at iteration n is calculated as follows:

$$y_j(n) = \varphi_{hj}(v_j(n)) \tag{2}$$

where $v_j(n)$ is the weighted sum of the inputs of the network in the j-th hidden layer neuron:

$$v_j(n) = \sum_{i=1}^{4} w_{ji}(n)x_i(n) + b_j(n) \tag{3}$$

where b_j is the bias.

These signals are transmitted to the output layer. The output is calculated as follows:

$$\phi_{NN}(n) = \varphi_o(v(n)) \tag{4}$$

where $v(n)$ is the weighted sum of the inputs of the hidden layer in the output layer neuron:

$$v(n) = \sum_{i=1}^{p} w_{1i}(n)y_i(n) + c(n) \tag{5}$$

where p is the total number of hidden layer neurons, c is the bias and φ_o is the activation function in the output layer. Next, the error signal is calculated. When the average squared error achieves a stopping criterion, the training is completed. This third stage is referred to as the forward pass computation. It is worth highlighting that in the forward pass computation, the weights and bias remain unaltered throughout the network. If the stopping criterion is not reached, the network training will continue to the next stage called the backward pass computation. This stage begins at the output layer by passing the error signals leftward through the network layer-by-layer and recursively computing the local gradient for each neuron as:

$$\delta_j^{(l)}(n) = \left[\begin{array}{ll} e_j^{(L)}(n)\delta_j'(v_j^{(L)}(n)) & \text{for neuron j in output layer L} \\ \delta_j'(v_j^{(L)}(n)) \sum\limits_k \delta_k^{(l+1)}(n) w_{kj}^{(l+1)}(n) & \text{for neuron j in hidden layer l} \end{array} \right. \tag{6}$$

where $\delta_j'(\cdot)$ denotes differentiation with respect to the argument. The synaptic weights and biases of the network in layer l according to the generalized delta rule are adjusted as:

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \alpha \left[w_{ji}^{(l)}(n-1) \right] + \eta \delta_j^{(l)} y_i^{l-1}(n) \tag{7}$$

where η is the learning-rate parameter and α is the momentum constant.

The fifth stage is the iteration. This stage presents new epochs of training patterns to iterate the forward and backward computations until the stopping criterion is met.

Training Patterns

The selection of the training patterns is a crucial process. The training patterns must contain data of the vehicle representative maneuvers, so as to characterize non-linear vehicle behavior.

As is mentioned in the previous section, the training pattern is formed by the input vector ($\mathbf{x} = \left\{ a_{ym}, a_{xm}, \dot{\phi}_m, \dot{\psi}_m \right\}$) and the output ($d = \{\phi_d\}$). Each of the training patterns has been obtained from an experimentally-validated TruckSim vehicle model (see Section 3.2).

The maneuvers that have been conducted are a Double Lane Change (DLC) and Lane Change (LC) maneuvers (from 30 to 140 km/h), as well as J-turn maneuvers (from 20 to 45 km/h). All maneuvers have been simulated considering road friction coefficients of 0.3, 0.5 and 1. For DLC and LC maneuvers with a road friction coefficient of 0.3, the maximum speed had to be limited to 80 km/h. Higher speeds than 80 km/h caused the rollover of the vehicle. A summary of the training datasets is shown in Table 1.

Table 1. Training datasets for NN, DLC, Double Lane Change, LC, Lane Change.

Road Friction Coefficients	Maneuvers	Speeds (km/h)
0.5–1	DLC LC	30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140
0.3	DLC LC	30, 40, 50, 60, 70, 80
0.3–0.5–1	J-turn	20, 25, 30, 35, 40, 45

2.2. Kalman Module

The Kalman module employs a discrete stochastic state-space form. The purpose of this module is to estimate the internal state of a linear dynamic system by means of a Kalman filter. The Kalman filter is a mathematical tool that is used for stochastic estimation from noisy sensor measurements. The real vehicle measurements include a substantial quantity of noise, as well as unobserved states in the system, which must be estimated. In this research, the unobserved state is the roll angle. The preliminary reconstruction of the roll angle obtained from the NN-based observer is used as a "pseudo-measurement" input to the Kalman filter. This previous calculation presents the advantage of considering the system non-linearities, thus providing good estimations, even though a linear vehicle model, represented as a state-space model, is used.

2.2.1. State-Space Vehicle Model

The dynamic vehicle model used in the Kalman filter algorithm is a linear model. When the vehicle model and measurement model equations are linear, the Kalman filter estimates the state vector recursively. An advantage of using linear systems is that they are easy to implement allowing the usage of the Kalman filter estimators in real time. For this reason, the dynamic model detailed in the estimation process is a 1-DOF vehicle model, which only represents vehicle roll motion. In Figure 3, the vehicle roll model is shown. The motion is described using a coordinate system (x, y, z) fixed in the vehicle. The vehicle roll angle, ϕ, is referenced from the vehicle's vertical z-axis. It is assumed that the vehicle sprung mass rotates around the roll center of the vehicle. A detailed description of this model can be found in [10]. The differential equation obtained from the vehicle's lateral dynamics can be written as:

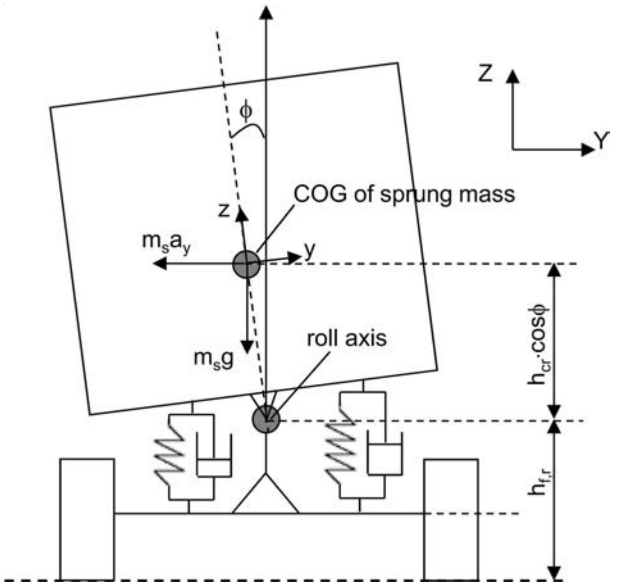


Figure 3. Schematic of the vehicle roll motion.

$$I_{xx}\ddot{\phi} + C_R\dot{\phi} + K_R\phi = m_s a_y h_{cr} + m_s h_{cr} g \sin(\phi) \tag{8}$$

where I_{xx} is the moment of inertia of the sprung mass m_s with respect to the roll axis, C_R and K_R denote respectively the total torsional damping and stiffness coefficients of the roll motion of the vehicle, h_{cr} is the height of the sprung mass about the roll axis, g is the gravitational constant and a_y is the lateral acceleration.

The lateral load transfer function can be obtained assuming that roll acceleration $\ddot{\phi}$ and velocity $\dot{\phi}$ are equal to zero. The steady-state equation for lateral load transfer applied to the left-hand side of the vehicle is given by:

$$\Delta F_{zl} = -2\left(\frac{k_f}{e_f} + \frac{k_r}{e_r}\right)\phi - 2m_s\frac{a_y}{l}\left(\frac{l_r h_f}{e_f} + \frac{l_f h_r}{e_r}\right) \tag{9}$$

where h_f and h_r are the heights of the front and rear roll centers, respectively; k_f and k_r are the front and rear roll stiffnesses, respectively; e_f and e_r are the front and rear vehicle tracks, respectively; and l_f and l_r are the distance from the COG (Center Of Gravity) to the front and rear axles, respectively. It must be noted that the lateral acceleration, a_y, used in Equations (8) and (9), is an inertial acceleration generated at the COG. Since the IMU provides a measurement of the acceleration due to the vehicle's motion (a_{ym}) and due to gravitational acceleration (g), the lateral acceleration (a_y) can be computed as:

$$a_{ym} = a_y \cos(\phi) + g \sin(\phi) \tag{10}$$

Assuming that the small roll angle approximation (i.e., $\sin(\phi) \approx \phi$ and $\cos(\phi) \approx 1$) is valid, the measured lateral acceleration, a_{ym}, can be expressed as:

$$a_{ym} = a_y + g\phi \tag{11}$$

In addition, assuming that the pitching and the bounding motion of sprung mass are neglected and the road bank angle is small, the vehicle roll rate can be expressed as:

$$\phi \approx \phi_m \tag{12}$$

The vehicle model is represented as a continuous time state-space system as follows:

$$\begin{cases} \dot{\mathbf{x}}_s = \mathbf{A}\mathbf{x}_s + \mathbf{w} \\ \mathbf{y} = \mathbf{H}\mathbf{x}_s + \mathbf{v} \end{cases} \tag{13}$$

where $\mathbf{x}_s$ represents the state vector $[\Delta F_{zl}, a_y, \dot{a}_y, \phi, \dot{\phi}]^T$; $\mathbf{A}$ is the state evolution matrix; $\mathbf{y}$ is the measurement vector; $[a_{ym}, \phi, \dot{\phi}_m, \Delta F_{zl}]^T$; $\mathbf{H}$ is the observation matrix; and $\mathbf{w}$ and $\mathbf{v}$ are the state disturbance and the observation noise vectors, respectively, that are assumed to be Gaussian, uncorrelated and zero mean:

$$\begin{aligned} \mathbf{w} &\sim N(0, \mathbf{Q}) \\ \mathbf{v} &\sim N(0, \mathbf{R}) \end{aligned} \tag{14}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are the covariance matrices describing the second-order properties of state and measurement noise:

$$\mathbf{Q} = \begin{bmatrix} 1000(N) & 0 & 0 & 0 & 0 \\ 0 & 0.1(m/s^{-2}) & 0 & 0 & 0 \\ 0 & 0 & 0.1(m/s^{-3}) & 0 & 0 \\ 0 & 0 & 0 & 0.1(rad) & 0 \\ 0 & 0 & 0 & 0 & 0.1(rad/s^{-2}) \end{bmatrix} \tag{15}$$

$$\mathbf{R} = \begin{bmatrix} 0.01(m/s^{-2}) & 0 & 0 & 0 \\ 0 & 0.01(rad) & 0 & 0 \\ 0 & 0 & 0.01(rad/s^{-2}) & 0 \\ 0 & 0 & 0 & 100(N) \end{bmatrix} \tag{16}$$

R depends on sensor quality (the yaw and roll rates) and the lateral load transfer and pseudo-roll angle estimator quality. **Q** is often unknown and is tuned depending the developed model. **Q** and **R** are assumed time invariant and diagonal for simplicity reason.

According to the chosen state-space vector and measurements, the matrices **A** and **H** are defined as:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & -2\frac{m_s}{l}\left(\frac{l_r h_f}{e_f} + \frac{l_f h_r}{e_r}\right) & 0 & -2\left(\frac{k_f}{e_f} + \frac{k_r}{e_r}\right) \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & m_s\frac{h_{cr}}{I_{xx}} & 0 & \frac{m_s g h_{cr} - K_R}{I_{xx}} & -\frac{C_R}{I_{xx}} \end{bmatrix} \quad \mathbf{H} = \begin{bmatrix} 0 & 1 & 0 & g & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{17}$$

In order to operate with the sensor data, the discrete state-space system is obtained using the first order approximation of Euler $\dot{\mathbf{x}}_{k-1} = \frac{\mathbf{x}_k - \mathbf{x}_{k-1}}{T_s}$, where T_s is the sampling time. Therefore, the discrete system can be expressed as:

$$\begin{aligned} \mathbf{x}_{s,k} &= \mathbf{A}_d \mathbf{x}_{s,k-1} + \mathbf{w}_k \\ \mathbf{y}_k &= \mathbf{H}\mathbf{x}_{s,k} + \mathbf{v}_k \end{aligned} \tag{18}$$

where $\mathbf{x}_{s,k} = \left[\Delta F_{zl,k}, a_{y,k}, \dot{a}_{y,k}, \phi\right]$, and the matrix $\mathbf{A}_d$ can be expressed as:

$$\mathbf{A}_d = \begin{bmatrix} 1 & 0 & -2T_s\frac{m_s}{l}\left(\frac{l_r h_f}{e_f} + \frac{l_f h_r}{e_r}\right) & 0 & -2T_s\left(\frac{k_f}{e_f} + \frac{k_r}{e_r}\right) \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & T_s \\ 0 & T_s m_s\frac{h_{cr}}{I_{xx}} & 0 & T_s\frac{m_s g h_{cr} - K_R}{I_{xx}} & 1 + T_s\frac{C_R}{I_{xx}} \end{bmatrix} \tag{19}$$

2.2.2. Kalman Filter Algorithm

In this work, a Linear Kalman Filter (LKF) algorithm was used to estimate the vehicle state. The LKF is summarized in the following recursive equations:

1. Time update

   Prediction of state and covariance:

$$\bar{\mathbf{x}}_{s,k|k-1} = \mathbf{A}_d \bar{\mathbf{x}}_{s,k-1|k-1} \tag{20}$$

$$\mathbf{P}_{s,k|k-1} = \mathbf{A}_d \mathbf{P}_{s,k-1|k-1}\mathbf{A}_d^T + \mathbf{Q}_s \tag{21}$$

2. Measurement update:

Kalman gain:

$$\mathbf{K}_{s,k} = \mathbf{P}_{s,k|k-1}\mathbf{H}^T[\mathbf{H}\mathbf{P}_{s,k|k-1}\mathbf{H}^T + \mathbf{R}_s]^{-1} \tag{22}$$

State and covariance estimation:

$$\bar{\mathbf{x}}_{s,k|k} = \bar{\mathbf{x}}_{s,k|k-1} + \mathbf{K}_{s,k}[\mathbf{y}_{measured} - \mathbf{H}\bar{\mathbf{x}}_{s,k|k-1}] \tag{23}$$

$$\mathbf{P}_{s,k|k} = [\mathbf{I} - \mathbf{K}_{s,k}\mathbf{H}]\mathbf{P}_{s,k|k-1} \tag{24}$$

The vector $y_{measured} = [a_{ym}, \phi, \phi, \Delta F_{zl}]^T$ contains sensor data, such as the lateral acceleration, a_{ym}, and the roll rate, $\dot{\phi}$, and pseudo-measurements, such as the lateral load transfer, ΔF_{zl}, calculated by Equation (9), and roll angle, ϕ. In order to prove the effectiveness of the proposed method, the pseudo-roll angle is computed in two different ways: (1) considering the suspension deflection, Equation (1); (2) considering the proposed NN estimator.

3. Results and Discussion

In this section, firstly, simulation results are presented to prove the effectiveness of the estimator proposed for different severe maneuvers and road conditions. Secondly, the proposed estimator is validated by means of experimental results using a real vehicle.

3.1. Experimental Vehicle Setup

The vehicle used for this research was a Mercedes Sprinter, as shown in Figure 4. The vehicle was equipped with a steering angle sensor MSW 250 Nm from Kistler, a Vbox 3i dual antenna data logger, an IMU and two GPS antennas from Racelogic. The IMU was mounted on the vehicle base, close to its COG. The two antennas were mounted at 90° to the true heading of the vehicle and on the roof, in order to accurately measure the roll angle. The roll angle value was used to validate the proposed estimator. Suspension deflection was experimentally measured by means of two linear potentiometers, Type SA-LP075 from 2D-Data, recording data from for the front suspension, and two sensors, Type LVDT MTN from Monitran, for the rear suspension.

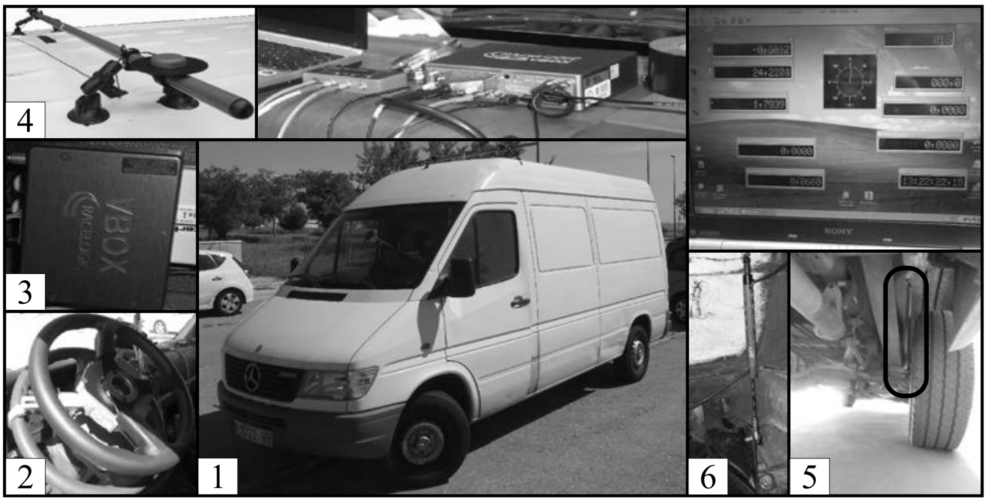


Figure 4. Experimental setup on a test vehicle. (1) Mercedes Sprinter. (2) Steering angle sensor MSW 250 Nm. (3) Vbox 3i dual antenna data logger. (4) GPS antennas. (5) LVDT MNT potentiometer. (6) SA-LP075 potentiometer.

The installed sensors provided measurements of the steering wheel, δ, the lateral acceleration, a_{ym}, the longitudinal acceleration, a_x, the vehicle longitudinal speed, V_x, the yaw rate, $\dot{\psi}$, the roll rate,

$\dot{\phi}$, the roll angle, ϕ, the front left suspension deflection, Δ_{11}, the front right suspension deflection, Δ_{12}, the rear left suspension deflection, Δ_{21}, and the rear right suspension deflection, Δ_{22}.

3.2. Experimental Adjustment of Vehicle Model Parameters

The TruckSim simulation vehicle model was validated using experimental results from the real vehicle. TruckSim software [23] is a widely-used simulation software in the automotive industry that combines traditional and modern multi-body vehicle dynamics based on parametric modeling. One of the main advantages of using a simulated vehicle model is the ability to perform different types of vehicle maneuvers that attempt to avoid possible accidents under different road conditions. In addition, simulation models guarantee test reproducibility.

The schema for the vehicle model validation is shown in Figure 5. The lateral acceleration ($a_{ym,t}$), the roll rate ($\dot{\phi}_t$), the roll angle (ϕ_t) and the yaw rate ($\dot{\psi}_t$) obtained from the simulated model were compared with experimental data. The model parameters were adjusted by trial and error according to the differences between the experimental and simulation data.

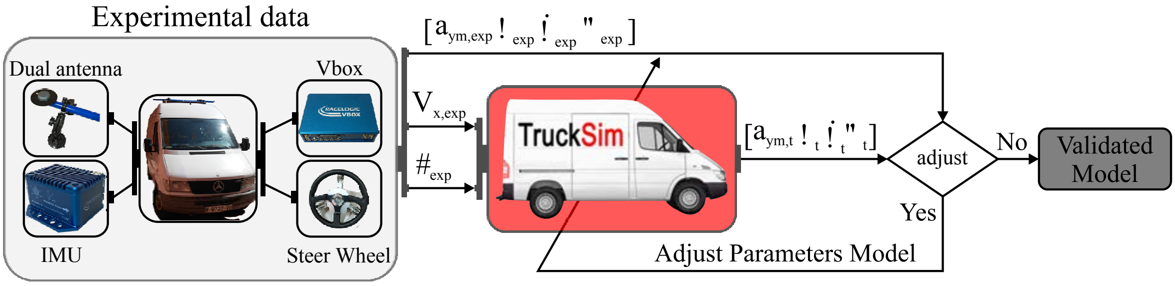


Figure 5. Scheme of the vehicle model design.

The performance of the vehicle model was proven in both the DLC and LC maneuvers with vehicle speeds of 50 and 70 km/h on dry pavement. Figure 6 shows the comparative results for the TruckSim simulation vehicle model using the experimental data obtained during a DLC maneuver for the real vehicle traveling at 70 km/h. The figure indicates excellent agreement between the simulation and the experimental results.

In addition to the graphical evidence of the effectiveness of the proposed simulation model, a quantitative analysis that took into consideration the error of the different maneuvers was computed. The following equation was used to represent the norm error as a function of time [24]:

$$E_t = \frac{\varepsilon_t}{\sigma_t} \tag{25}$$

where:

$$\begin{aligned} \varepsilon_t &= \int_0^T \left(\lambda_{Exp} - \lambda_{Trucksim}\right)^2 \\ \sigma_t &= \int_0^T \left(\lambda_{Exp} - \mu_{Exp}\right)^2 \end{aligned} \tag{26}$$

where λ_{Exp} and $\lambda_{Trucksim}$ represent the measured and simulated lateral acceleration, yaw and roll rates and roll angle, respectively, and μ_{Exp} is the mean value of the lateral acceleration, yaw rate and roll angle obtained from the real vehicle during time T.

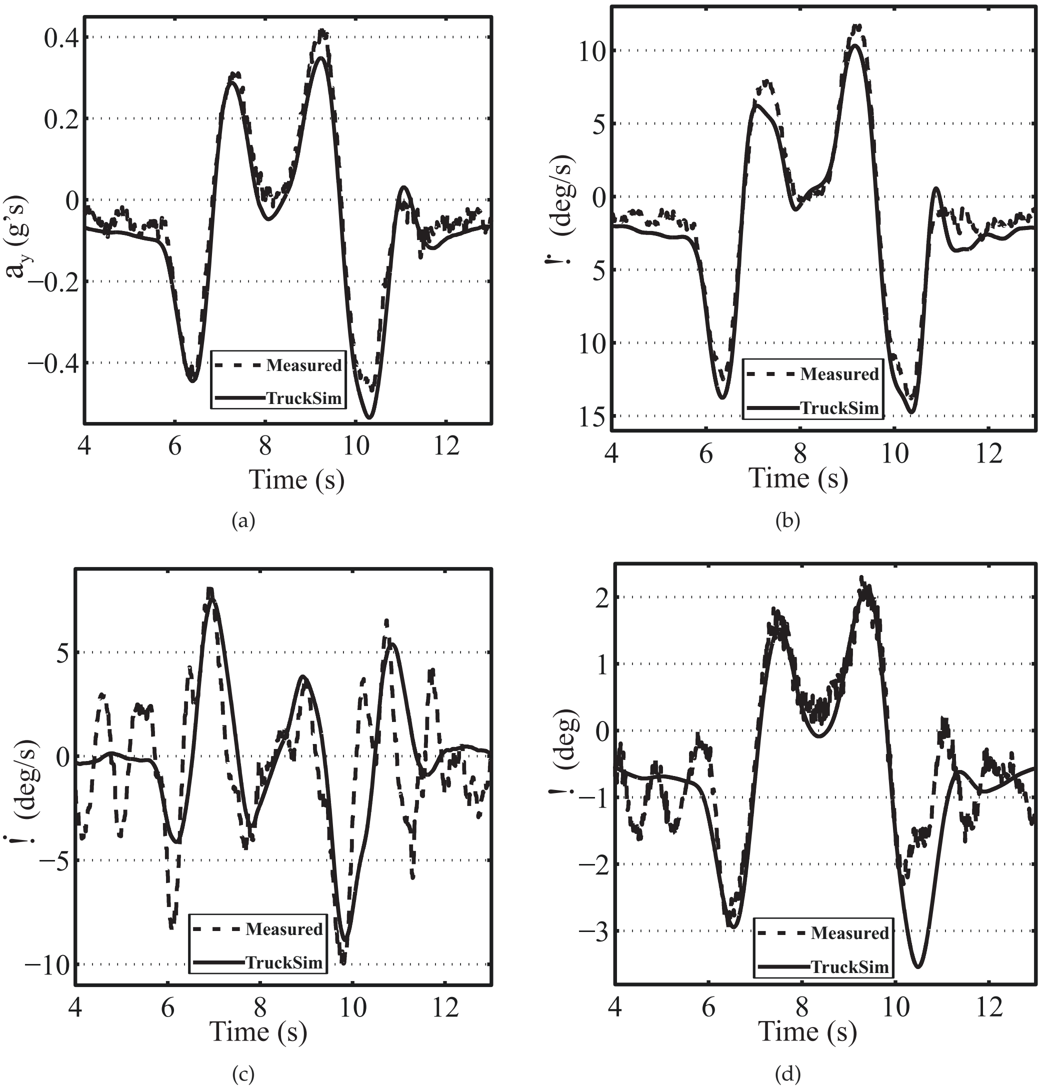


Figure 6. DLC maneuvers at 70 km/h for the experimental adjustment of vehicle model parameters (solid line: TruckSim; dashed line: measured): (**a**) lateral acceleration, (**b**) yaw rate, (**c**) roll rate and (**d**) roll angle.

For the DLC at 70 km/h (see Figure 6), the norm error of the lateral acceleration, the yaw and roll rates and the roll angle were 0.221, 0.210, 0.792 and 0.522, respectively. The norm and maximum errors are provided in Table 2 for DLC at 50 km/h and LC at 50 and 70 km/h. From the results, we can conclude that the created vehicle model accurately represents the real vehicle.

Table 2. Norm and maximum errors for the DLC and LC maneuvers.

	DLC at 50 km/h		**DLC at 70 km/h**		**LC at 50 km/h**		**LC at 70 km/h**	
	E_t	$E_{\max}$	E_t	$E_{\max}$	E_t	$E_{\max}$	E_t	$E_{\max}$
a_{ym} $(g's)$	0.130	0.080	0.221	0.138	0.138	0.035	0.418	0.050
$\dot{\psi}$ $(^\circ/s)$	0.114	1.748	0.210	2.791	0.109	1.365	0.176	0.827
$\dot{\phi}$ $(^\circ/s)$	0.696	4.023	0.792	8.240	0.587	3.322	1.134	5.115
ϕ $(^\circ)$	0.678	2.206	0.522	2.068	0.621	0.675	1.279	0.273

Table 3 shows the adjusted parameters for the state-space vehicle model.

Table 3. State-space vehicle model parameters, COG, Center Of Gravity.

Symbol	Description	Value	Unit
C_R	Total torsional damping of the suspension	54,447.87	Nms/rad
e_f	Front vehicle track	1.638	m
e_r	Rear vehicle track	1.630	m
h_{cr}	Height of the sprung mass about the roll axis	0.980	m
h_f	Height of the front roll center	0.348	m
h_r	Height of the rear roll center	0.348	m
I_{xx}	Roll inertia	500	kg·m^2
k_f	Front roll stiffness	371,530.334	Nm/rad
k_r	Rear roll stiffness	371,530.334	Nm/rad
K_R	Total torsional stiffness of the suspension	743,060.669	Nm/rad
l_f	Distance from front tire to COG	1.51	m
l_r	Distance from rear tire to COG	2.04	m
m_s	sprung mass	1700	kg

3.3. Simulated Validation

To prove the effectiveness of the proposed roll angle estimator based on NN and the Kalman filter, two severe maneuvers, such as the sine sweep and slalom, were conducted. The former was performed at 50 and 70 km/h on a road surface with friction coefficients of 0.7 and 0.3, respectively. The latter was conducted at 35 km/h for the same friction coefficients. In order to analyze the effect of the sensor measurement's noise on the estimation of the roll angle, Gaussian noises with zero mean and variances of 0.01 m/s^2, 0.01 m/s^2, 0.01°/s and 0.01°/s were added to a_{ym} (lateral acceleration), a_x (longitudinal acceleration), $\dot{\phi}$ (roll rate) and $\dot{\psi}$ (yaw rate), respectively.

Figure 7 shows the comparative results of the Deflection + LKF-based observer (DEF + LKF) and NN + LKF-based observer for a slalom maneuver at 35 km/h with a friction coefficient of 0.3. Figure 8 shows the comparative results of the DEF + LKF-based observer and NN + LKF-based observer for a sine sweep maneuver at 70 km/h with a friction coefficient of 0.3. These figures demonstrate that the proposed observer based on NN yields a better behavior than the observer based on the suspension deflection.

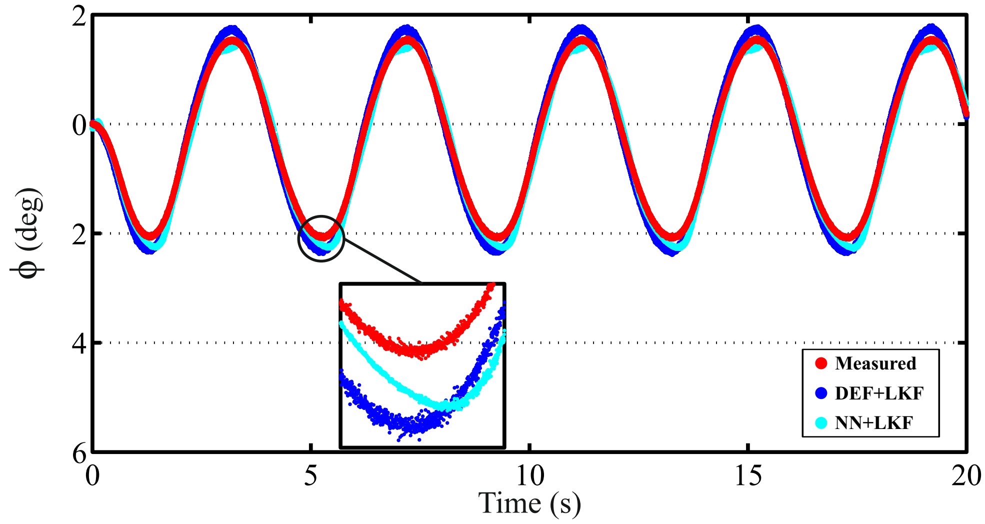


Figure 7. Simulation results for a slalom maneuver at 35 km/h with a friction coefficient of 0.3 (red points: measured; blue points: Deflection (DEF) + Linear Kalman Filter (LKF); cyan points: NN + LKF).

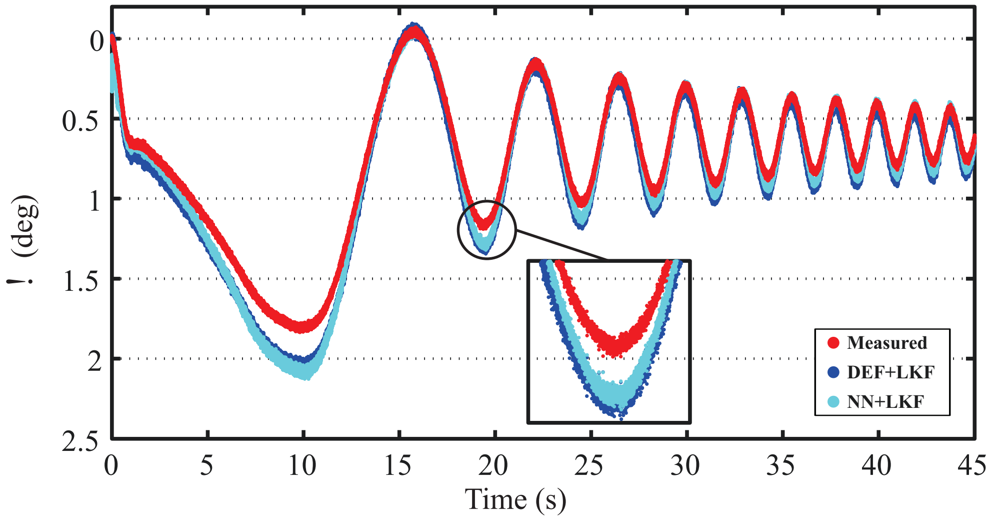


Figure 8. Simulation results for a sine sweep maneuver at 70 km/h with a friction coefficient of 0.3 (red points: measured; blue points: DEF + LKF; cyan points: NN + LKF).

The norm and maximum errors are provided in Table 4. The norm errors for the NN + LKF-based estimator are smaller than those from DEF + LKF-based estimator. The differences in maximum errors are higher in the cases for which the friction coefficient is 0.3. However, the maximum difference is 0.086°, which is negligible.

The proposed method provides better results in 100% of the analyzed simulated cases (four cases) for the norm error; whereas, a 50% success rate is obtained for the maximum error compared with the DEF + LKF method.

Table 4. Norm and maximum errors for roll angle estimators for sine sweep and slalom maneuvers.

	SLA-35-0.7		**SLA-35-0.3**		**SS-50-0.7**		**SS-70-0.3**	
	E_t (−)	$E_{\max}$ (°)	E_t (−)	$E_{\max}$ (°)	E_t (−)	$E_{\max}$ (°)	E_t (−)	$E_{\max}$ (°)
DEF + LKF	0.221	0.158	0.250	0.279	0.122	0.302	0.129	0.295
NN + LKF (*)	0.093	0.152	0.230	0.365	0.078	0.282	0.099	0.349

(*) Proposed; SLA-35-0.7: Slalom at 35 km/h and friction coefficient 0.7; SLA-35-0.3: Slalom at 35 km/h and friction coefficient 0.3; SS-50-0.7: Sine sweep at 50 km/h and friction coefficient 0.7; SS-70-0.3: Sine sweep at 70 km/h and friction coefficient 0.3.

3.4. Experimental Validation

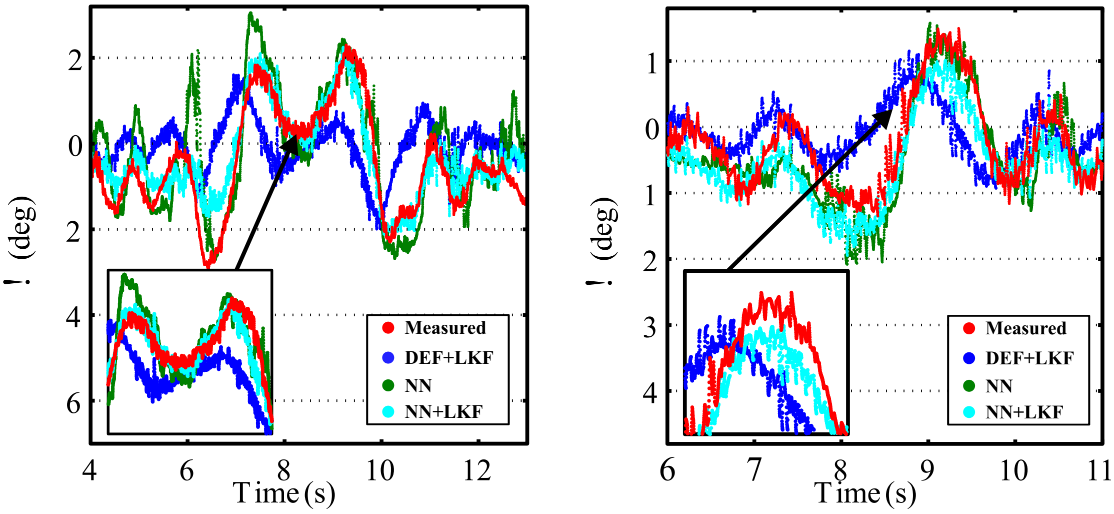
The performance of the proposed estimator was verified for a real vehicle traveling at different speeds on dry pavement under different maneuvers.

Figure 9 shows the results for DLC and LC maneuvers for a real vehicle traveling at 70 km/h. Table 5 shows the norm and maximum errors for DLC and LC at speeds of 50 and 70 km/h. In all cases, the norm error of the proposed estimator is smaller than the estimator based on suspension deflection. However, when the vehicle is traveling at 50 km/h, the maximum error of the NN + LKF-based observer is greater for both maneuvers. Nevertheless, the maximum difference is 0.2°, which is acceptable taking into account that the maximum measured roll angle at 50 km/h is about 2°. In order to prove the necessity of incorporating a LKF in the NN estimator, the results for a single NN estimator are also shown. In the majority of cases (three out of four cases), the NN + LKF-based estimator improves the roll angle estimation.

Table 5. Norm and maximum errors for roll angle estimators for the DLC and LC maneuvers.

	DLC at 50 km/h		**DLC at 70 km/h**		**LC at 50 km/h**		**LC at 70 km/h**	
	E_t $(-)$	$E_{\max}$ $(^\circ)$	E_t $(-)$	$E_{\max}$ $(^\circ)$	E_t $(-)$	$E_{\max}$ $(^\circ)$	E_t $(-)$	$E_{\max}$ $(^\circ)$
DEF + LKF	0.912	1.526	1.038	2.786	1.206	0.903	2.087	0.477
NN	0.806	2.602	0.692	1.620	0.726	0.963	0.849	0.962
NN + LKF (*)	0.547	1.723	0.446	1.841	0.478	0.961	0.964	0.276

(*) Proposed.



(a)

(b)

Figure 9. Experimental results for a vehicle traveling at 70 km/h on dry pavement. (red points: measured; blue points: DEF + LKF; green points: NN; cyan points: NN + LKF): (**a**) DLC maneuver and (**b**) LC maneuver.

Additionally, the proposed estimator was evaluated when the vehicle is traveling at about 45 km/h under a slalom maneuver. The results are depicted in Figure 10. The norm and maximum errors of the NN + LKF estimator are 0.921 and 2.070°, respectively; whereas, the norm and maximum errors for the NN-based and DEF + LKF-based estimators are [1.040; 2.232°] and [1.181; 2.081°], respectively. In this case, the proposed estimator yields lower errors.

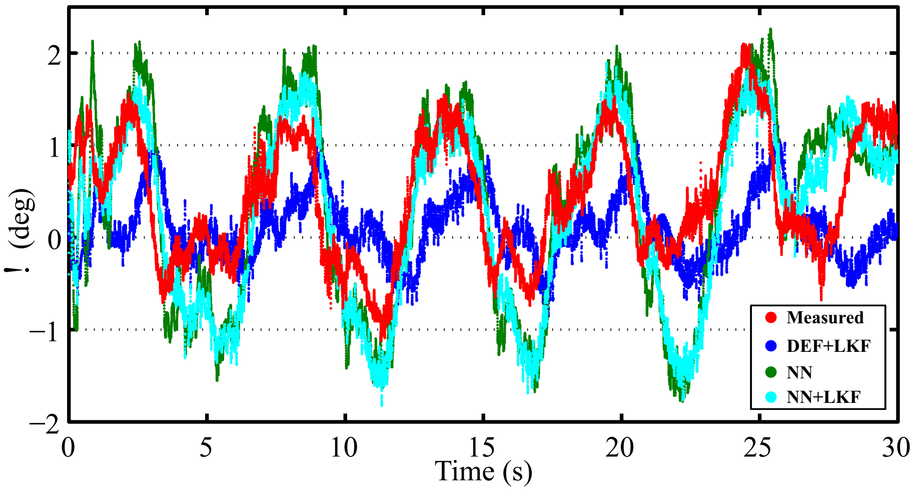


Figure 10. Experimental results for slalom maneuver (red points: measured; blue points: DEF + LKF; green points: NN; cyan points: NN + LKF).

Finally, two tests were performed with a combination of different maneuvers, as depicted in Figures 11 and 12. These figures show the real environment and the vehicle trajectory, the steering wheel angle profile, the longitudinal vehicle speed profile and the real and estimated roll angle.

The first test corresponds to a real vehicle traveling on dry pavement under a J-turn and slalom maneuvers and the second one to the same vehicle under a J-turn and DLC. Figure 11a shows a J-turn and slalom maneuvers, while Figure 12a depicts a J-turn and DLC maneuvers. Tables 6 and 7 show the norm and maximum errors. The proposed NN + LKF-based estimator proves to be better than the suspension deflection and single NN estimators, for both the norm and maximum errors.

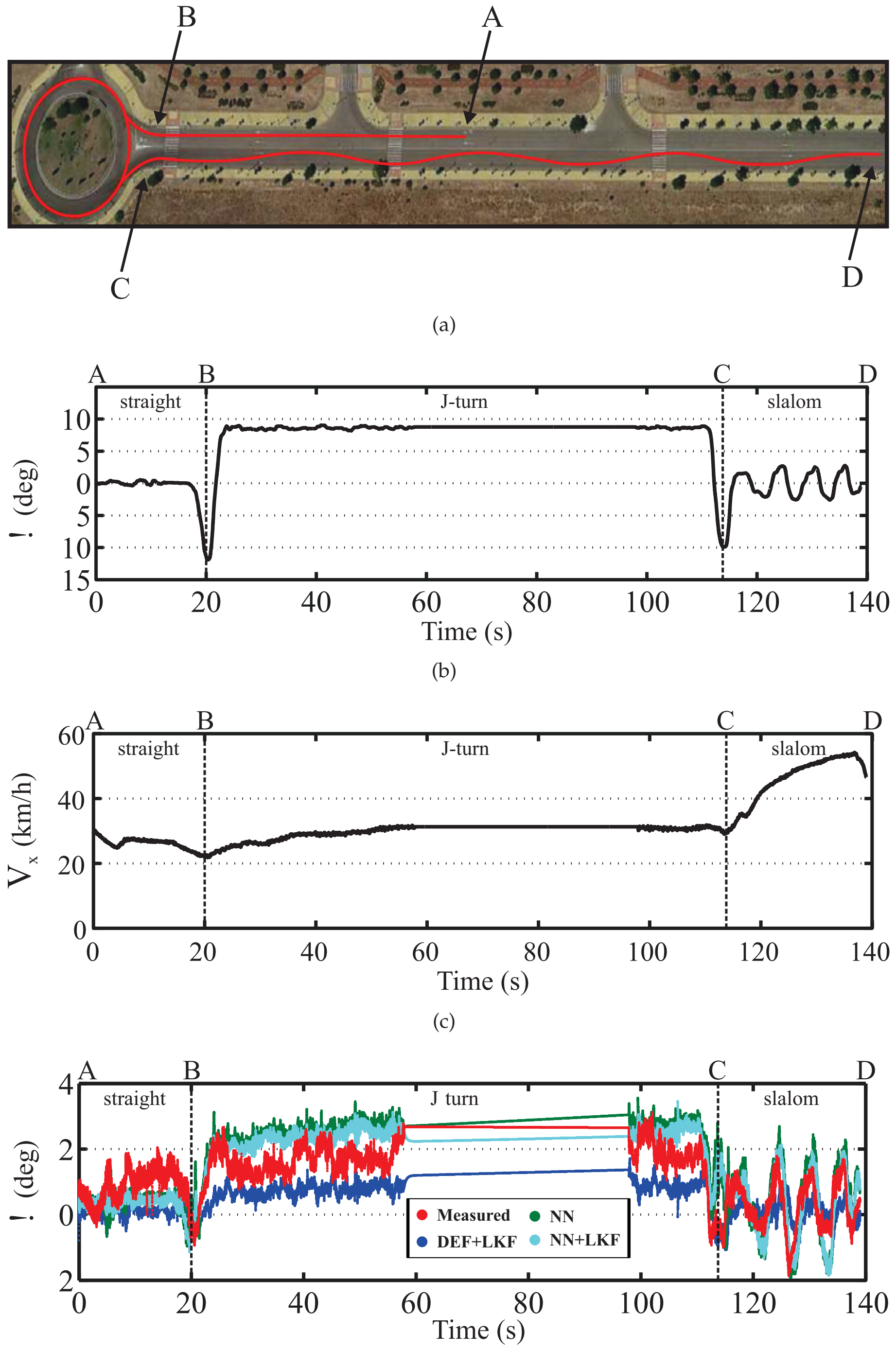
The proposed method provides better results in 100% of the analyzed experimental cases (10 cases) for the norm error; whereas, an 80% success rate is obtained for the maximum error compared with the DEF + LKF method. In the worst case, the maximum difference between the DEF + LKF and NN + LKF is 0.197°, which is negligible.

Table 6. Norm and maximum errors for roll angle estimators for J-turn and slalom maneuvers.

	Total Sector		J-Turn Sector		Slalom Sector	
	E_t $(-)$	$E_{\max}$ $(^\circ)$	E_t $(-)$	$E_{\max}$ $(^\circ)$	E_t $(-)$	$E_{\max}$ $(^\circ)$
NN	0.757	2.071	1.174	1.799	1.026	1.664
DEF + LKF	1.081	2.495	1.571	2.220	1.073	1.980
NN + LKF	0.655	1.689	0.969	1.689	0.879	1.591

Table 7. Norm and maximum errors for roll angle estimators for J-turn and DLC maneuvers.

	Total Sector		J-Turn Sector		DLC Sector	
	E_t $(-)$	$E_{\max}$ $(^\circ)$	E_t $(-)$	$E_{\max}$ $(^\circ)$	E_t $(-)$	$E_{\max}$ $(^\circ)$
NN	1.347	3.680	1.547	1.572	0.836	2.811
DEF + LKF	1.687	4.196	1.646	2.663	1.461	3.752
NN + LKF	1.323	2.307	1.440	1.501	0.753	2.093



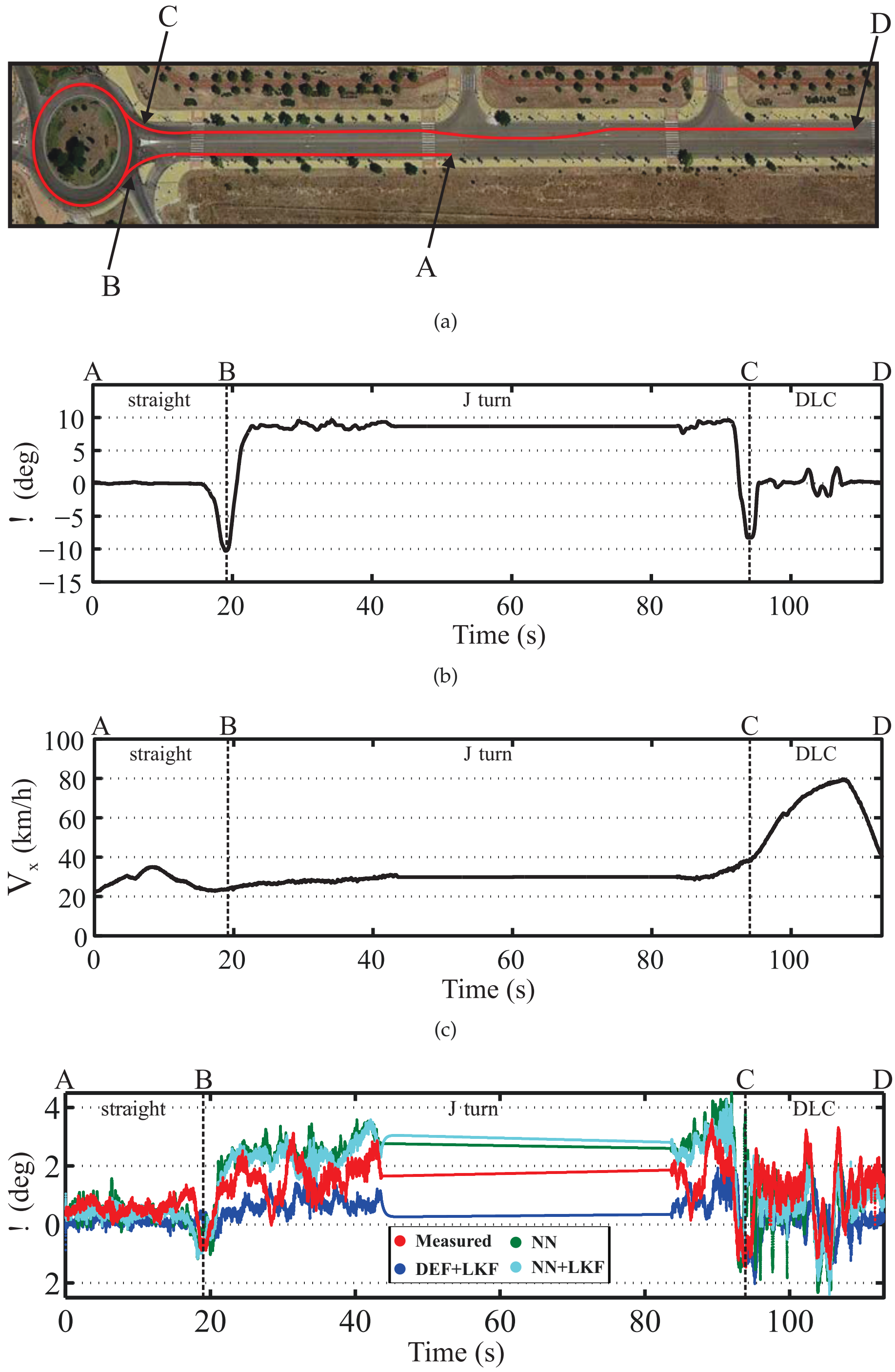
(a)

(b)

(c)

(d)

Figure 11. Experimental results for J-turn and slalom maneuvers (red points: measured; blue points: DEF + LKF; green points: NN; cyan points: NN + LKF): (**a**) the real environment and the vehicle trajectory, (**b**) the steering wheel angle profile, (**c**) the longitudinal vehicle speed profile and (**d**) the real and estimated roll angle.



(a)

(b)

(c)

(d)

Figure 12. Experimental results for J-turn and DLC maneuvers (red points: measured; blue points: DEF + LKF; green points: NN; cyan points: NN + LKF): (**a**) the real environment and the vehicle trajectory, (**b**) the steering wheel angle profile, (**c**) the longitudinal vehicle speed profile and (**d**) the real and estimated roll angle.

4. Conclusions

This article presents a novel estimator based on sensor fusion, which combines NN and LKF in order to estimate the vehicle roll angle. The proposed vehicle roll angle estimator architecture is formed by two modules: the NN module and the Kalman module. The NN module estimates a pseudo-roll angle through data, such as the lateral and longitudinal accelerations, the yaw and roll rates. The pseudo-roll angle is passed on to the Kalman module in order to filter noise and minimize the norm and maximum errors. The proposed NN + LKF-based estimator takes into account the vehicle motion nonlinearities. The main advantages of the proposed method are that only a single IMU is needed, and no additional suspension deflection sensors are required.

The NN + LKF-based estimator was validated with simulated and experimental results. The simulation analysis has proven the effectiveness of the proposed estimator in severe maneuvers with low and medium friction coefficients (0.3 and 0.7). In addition, experimental tests, such as, slalom, J-turn and DLC maneuvers, were conducted on dry pavement. For the overall analyzed tests, simulated and experimental ones, the proposed method yielded better results in 100% for the norm error compared with the DEF + LKF method; whereas, a 71% success rate was obtained for the maximum error, reaching an 80% success rate for experimental tests. In the worst case, the maximum difference between the DEF + LKF and NN + LKF is 0.197°, which is negligible.

Experimental results also show the necessity of including the LKF in the NN estimator in order to filter noise and, therefore, to improve the roll angle estimation.

Acknowledgments: This work was possible thanks to the funds provided by the Spanish Government through the CICYTProject TRA2013-48030-C2-1-R.

Author Contributions: Leandro Vargas-Meléndez, Beatriz L. Boada, María Jesús L. Boada, Antonio Gauchía and Vicente Díaz conceived and designed the simulations and the experiments; Leandro Vargas-Meléndez and Beatriz L. Boada performed the experiments; Leandro Vargas-Meléndez, Beatriz L. Boada, María Jesús L. Boada, Antonio Gauchía and Vicente Díaz analyzed the data; Leandro Vargas-Meléndez, Beatriz L. Boada and María Jesús L. Boada contributed with mathematical algorithms; and finally Leandro Vargas-Meléndez, Beatriz L. Boada, María Jesús L. Boada, Antonio Gauchía and Vicente Díaz wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Boada, M.J.L.; Boada, B.L.; Muñoz, A.; Díaz, V. Integrated control of front-wheel steering and front braking forces on the basis of fuzzy logic. *J. Automob. Eng.* **2006**, *220*, 253–267.
2. Hickman, J.S.; Guo, F.; Camden, M.C.; Hanowski, R.J.; Medina, A.; Mabry, J.E. Efficacy of roll stability control and lane departure warning systems using carrier-collected data. *J. Saf. Res.* **2015**, *52*, 59–63.
3. Boada, M.J.L.; Gauchía, A.; Calvo, J.A.; Díaz, V. Active roll control using reinforcement learning for a single unit heavy vehicle. *Int. J. Heavy Veh. Syst.* **2009**, *16*, 412–430.
4. Sampson, D.J.; Cebon, D. Active Roll Control of Single Unit Heavy Road Vehicles. *Veh. Syst. Dyn.* **2003**, *40*, 229–270.
5. Rajamani, R.; Piyabongkarn, D.; Tsourapas, V.; Lew, J. Real-time estimation of roll angle and CG height for active rollover prevention applications. In Proceedings of the 2009 American Control Conference, St. Louis, MO, USA, 10–12 June 2009.
6. Tafner, R.; Reichhartinger, M.; Horn, M. Robust online roll dynamics identification of a vehicle using sliding mode concepts. *Control Eng. Pract.* **2014**, *29*, 235–246.
7. Tseng, H.E.; Xu, L.; Hrovat, D. Estimation of land vehicle roll and pitch angles. *Veh. Syst. Dyn.* **2007**, *45*, 433–443.
8. Ryu, J.; Rossetter, E.; Gerdes, J. Vehicle sideslip and roll parameter estimation using GPS. In Proceedings of the International Symposium on Advanced Vehicle Control (AVEC), Hiroshima, Japan, 9–13 September 2002.
9. Hall, D.L.D.L.; Member, S.; Llinas, J. An introduction to multisensor data fusion. *IEEE Proc.* **1997**, *85*, 6–23.
10. Doumiati, M.; Baffet, G.; Lechner, D.; Victorino, A.; Charara, A. Embedded Estimation of the Tire/Road Forces and Validation in a Laboratory Vehicle. In Proceedings of 9th International Symposium on Advanced Vehicle Control, Kobe, Japan, 6–9 October 2008.
11. Rajamani, R.; Piyabongkarn, D.; Tsourapas, V.; Lew, J.Y. Parameter and State Estimation in Vehicle Roll Dynamics. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 1558–1567.
12. Jo, K.; Chu, K.; Sunwoo, M. Interacting multiple model filter-based sensor fusion of GPS with in-vehicle sensors for real-time vehicle positioning. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 329–343.

13. Bevly, D.M.; Ryu, J.; Gerdes, J.C. Integrating INS sensors with GPS measurements for continuous estimation of vehicle sideslip, roll, and tire cornering stiffness. *IEEE Trans. Intell. Transp. Syst.* **2006**, *7*, 483–493.
14. Nam, K.; Oh, S.; Fujimoto, H.; Hori, Y. Estimation of Sideslip and Roll Angles of Electric Vehicles Using Lateral Tire Force Sensors Through RLS and Kalman Filter Approaches. *IEEE Trans. Ind. Electron.* **2013**, *60*, 988–1000.
15. Chun, D.; Stol, K. Vehicle Motion Estimation using Low-Cost Optical Flow and Sensor Fusion 2012. In Proceedings of the 2012 19th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), Auckland, New Zealand, 28–30 November 2012; pp. 28–30.
16. Yoon, S.W.; Park, S.B.; Kim, J.S. Kalman Filter Sensor Fusion for Mecanum Wheeled Automated Guided Vehicle Localization. *J. Sens.* **2015**, *2015*, 1–7.
17. Boada, B.L.; Boada, M.J.L.; Diaz, V. Vehicle sideslip angle measurement based on sensor data fusion using an integrated ANFIS and an Unscented Kalman Filter algorithm. *Mech. Syst. Signal Process.* **2016**, *72–73*, 832–845.
18. Zhang, S.; Yu, S.; Liu, C.; Yuan, X.; Liu, S. A Dual-Linear Kalman Filter for Real-Time Orientation Determination System Using Low-Cost MEMS Sensors. *Sensors* **2016**, *16*, doi:10.3390/s16020264.
19. Loebis, D.; Sutton, R.; Chudley, J.; Naeem, W. Adaptive tuning of a Kalman filter via fuzzy logic for an intelligent AUV navigation system. *Control Eng. Pract.* **2004**, *12*, 1531–1539.
20. Chatterjee, A. Differential evolution tuned fuzzy supervisor adapted extended Kalman filtering for SLAM problems in mobile robots. *Robotica* **2009**, *27*, doi:10.1017/S0263574708004827.
21. Korniyenko, O.V.; Sharawi, M.S.; Aloi, D.N. Neural network based approach for tuning Kalman filter. In Proceedings of the IEEE International Conference on Electro Information Technology, Lincoln, NE, USA, 22–25 May 2005.
22. Simon, H. *Neural Networks: A Comprehensive Foundation*, 2nd ed.; Publisher: Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999.
23. TruckSim. Available online: https://www.carsim.com/products/trucksim/ (accessed on 27 July 2016).
24. Boada, M.J.L.; Calvo, J.A.; Boada, B.L.; Díaz, V. Modeling of a magnetorheological damper by recursive lazy learning. *Int. J. Non Linear Mech.* **2011**, *46*, 479–485.