

Article



Estimation of Vehicle Attitude, Acceleration, and Angular Velocity Using Convolutional Neural Network and Dual Extended Kalman Filter

Minseok Ok ¹, Sungsuk Ok ² and Jahng Hyon Park ^{1,*}

- ¹ Department of Automotive Electronics & Control Engineering, Hanyang University, Seoul 04763, Korea; doree81@hanyang.ac.kr
- ² Autonomous Driving Center, Hyundai Motor Company R&D Division, Seoul 06182, Korea; sungsuk.ok@hyundai.com
- * Correspondence: jpark@hanyang.ac.kr; Tel.: +82-2-2220-0448

Abstract: The acceleration of a vehicle is important information in vehicle states. The vehicle acceleration is measured by an inertial measurement unit (IMU). However, gravity affects the IMU when there is a transition in vehicle attitude; thus, the IMU produces an incorrect signal output. Therefore, vehicle attitude information is essential for obtaining correct acceleration information. This paper proposes a convolutional neural network (CNN) for attitude estimation. Using sequential data of a vehicle's chassis sensor signal, the roll and pitch angles of a vehicle can be estimated without using a high-cost sensor such as a global positioning system or a six-dimensional IMU. This paper also proposes a dual-extended Kalman filter (DEKF), which can accurately estimate acceleration/angular velocity based on the estimated roll/pitch information. The proposed method is validated by real-car experiment data and CarSim, a vehicle simulator. It accurately estimates the attitude estimation with limited sensors, and the exact acceleration/angular velocity is estimated considering the roll and pitch angle with de-noising effect. In addition, the DEKF can improve the modeling accuracy and can estimate the roll and pitch rates.

Keywords: sensor fusion; state estimation; vehicle dynamics; convolutional neural network; dual extended Kalman filter; vehicle roll and pitch angle; vehicle acceleration and angular velocity

1. Introduction

In recent decades, the vehicle controller has been developed significantly for stability and user convenience. Typically, the electronic stability controller (ESC) or active roll stabilization (ARS) are used for ensuring vehicle stability in chassis controller, adaptive cruise control (ACC), or lane keeping system (LKS) as well as for ensuring convenience in advanced driver assistant system (ADAS). To improve the performance of these controllers, vehicle states must be estimated with a high accuracy. The commonly required and important state information of aforementioned controllers are acceleration and angular velocity. In a vehicle, the inertial measure unit (IMU) measures the acceleration and angular velocity using an inertial force. However, if a transition in attitude (roll and pitch) occurs, the gravitational force is reflected in the sensor value. The IMU cannot distinguish between gravitational force and inertial forces; therefore, a change in the attitude of the vehicle causes a fatal error in the IMU.

Several studies have been conducted to overcome the errors in these accelerometers and to estimate the exact state of the vehicle. In [1,2], the adaptive Kalman filter was designed to minimize the effect of the accelerometer offset errors. In addition, the Kalman filter was used to estimate the accelerometer offset and vehicle velocity [3–5]. However, the accuracy of estimating vehicle states was limited because of a lack of information of the vehicle attitude.



Citation: Ok, M.; Ok, S.; Park, J.H. Estimation of Vehicle Attitude, Acceleration, and Angular Velocity Using Convolutional Neural Network and Dual Extended Kalman Filter. *Sensors* 2021, *21*, 1282. https:// doi.org/10.3390/s21041282

Academic Editor: Maria Jesús López Boada Received: 14 January 2021

Accepted: 8 February 2021 Published: 11 February 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). Several studies have been conducted to estimate the accurate vehicle attitude in various ways. Using IMU or GPS or vehicle dynamics, researchers tried to estimate the vehicle roll and pitch angle. In addition, due to the recent developments in artificial intelligence and neural networks, data-driven estimators have been used for state estimation [6]. These estimators have the advantage of having high accuracy by training the data directly in situations where accurate mathematical modeling is difficult. The literature review is discussed separately in Section 2.

This paper proposes a novel vehicle attitude estimator using a convolutional neural network (CNN), based on the advantages of a data-driven estimator. Figure 1 shows the architecture of the proposed algorithm. First, we select the features based on vehicle roll and pitch dynamics, importance judgment and sensor usability. By using the selected features sequentially [7], the three-dimensional IMU and vehicle sensor build a CNN-based regression model that can estimate the vehicle roll and pitch angles without GPS. Moreover, a dual extended Kalman filter (DEKF) is designed to estimate the exact acceleration/angular velocity based on the estimated roll and pitch angles with removing the effect of gravitational force. The proposed model is based on the sprung mass sixdegree-of-freedom (6-DOF) model and increases the modeling accuracy while estimating the tire cornering stiffness simultaneously. The performance of the proposed algorithm was verified using real car experiment data and MATLAB/Simulink with CarSim, a vehicle simulator. A CNN conducted training and verification using data collected from various driving conditions as an experimental vehicle with RT-3002 which is a high-accuracy GPSinertial navigation system (INS) from Oxford Technical Solutions Ltd. The performance of the DEKF was verified with a scenario in which postural changes occur using CarSim and MATLAB/Simulink.



Figure 1. Architecture overview.

The remainder of this paper is organized as follows. Section 2 describes the related literature review. Section 3 explains the methodology including neural network and DEKF. Section 4 describes the results of performance verification. Section 5 summarizes the study and contributions.

2. Literature Review

Several studies have been conducted to estimate the accurate vehicle attitude in various ways. In [8], authors proposed an observer which can estimate the land vehicle's roll and pitch by using an IMU and the kinematics model. Also, the adaptive Kalman filter was proposed based on IMU aided by vehicle dynamics [9]. However, these methods have a low accuracy in dynamic situations. This problem was attempted to be solved in [10–13], based on the Kalman filter by compensating for the external acceleration that interfered with the estimation of the attitude, but the accuracy was limit.

Therefore, many researchers have proposed sensor fusion method using not only IMU but fusion with global positioning system (GPS). Representatively, the method in which IMU and GPS fusion with Kalman filter [14–16] or sliding mode observer [17] can be used.

These methods can increase the accuracy, but they require high-cost sensors such as GPS and are highly affected by GPS performance.

Accordingly, vehicle attitude estimator studies were conducted without the use of GPS, based on the characteristics of the vehicle dynamics. Using vehicle roll dynamics, a dynamic observer design based on a reliable rollover index [18] was proposed. Also, the Kalman filter [19] and the robust observer [20] based on vehicle roll dynamics which estimate roll angle were proposed. Alternatively, using lateral dynamics, the research which estimate road bank angle [21,22] was proposed. However, these methods exhibited limited accuracy in transient situations. A method for estimating both roll and pitch attitudes using a six-dimensional IMU and bicycle model [23,24] was also studied. These methods exhibited a high accuracy, but they have a short validation range and required hard-to-get data such as six-dimensional IMU.

Several recent studies have also been conducted to estimate the vehicle roll and pitch angle using neural network. In [25–27], the vehicle roll angle estimation method using sensor fusion with a neural network and Kalman filter was proposed. Furthermore, a vehicle roll and road bank angle estimation method based on a deep neural network was introduced [28]. However, the pitch attitudes were not estimated, and the six-dimensional IMU data such as roll rate or vertical acceleration were required, which complicated the estimation. Literature reviews are summarized in Table 1.

 Table 1. Literature review summarizing.

References	Methodology	Model
[8]	Linear observer	IMU kinematic model
[9]	Kalman filter	IMU + Vehicle dynamics (bicycle model and wheel model)
[10–13]	Kalman filter	IMU external acceleration model
[14–16]	Kalman filter	IMU + GPS model
[17]	Sliding mode observer	IMU + GPS model
[18]	Dynamic observer	Vehicle roll dynamics
[19]	Kalman filter	Vehicle roll dynamics
[20]	Dual Kalman filter	Vehicle roll dynamics
[21,22]	Linear observer	Vehicle lateral dynamics (bicycle model)
[23,24]	Linear observer, Sliding mode observer	Vehicle lateral dynamics (bicycle model) + IMU
[25–27]	Kalman filter + neural network	Vehicle roll dynamics + fully connected layer
[28]	Neural network	Fully connected layer

3. Methodology

3.1. Data-Driven-Based Estimator (Neural Network) Design

3.1.1. Feature Selection

Feature selection is based on vehicle dynamics, sensor usability, and attention mechanism. The purpose of neural network is to design a regression model for calculating the roll and pitch angles, so select the primary feature candidates for the vehicle's roll and pitch dynamics. Vehicle roll dynamics [29] can be written as:

$$\Sigma M_x = h_R m_s gsin\phi - \frac{1}{2} k_s l_s^2 sin\phi - \frac{1}{2} b_s l_s^2 \dot{\phi} sin\phi + \Sigma F_y (h_{CG} - h_R)$$
(1)

$$\Sigma F_y = 2C_{\alpha f} \left(\delta - \frac{v_y + l_f \dot{\psi}}{v_x} \right) + 2C_{\alpha r} \left(-\frac{v_y - l_r \dot{\psi}}{v_x} \right) + m_s g sin\phi.$$
⁽²⁾

where M_x is the moment respect to the roll axis, h_R is the height of the from center of gravity to roll center, m_s is the vehicle sprung mass, g is the gravity acceleration, ϕ is the roll of vehicle, k_s is the stiffness of the suspension, l_s is the length of the wheel track, b_s is the damping coefficient of the suspension, and h_{CG} is the height of the center of gravity from the ground.

The lateral force applied to the vehicle can be expressed by (2) based on the bicycle model. $C_{\alpha f}$ and $C_{\alpha r}$ are the cornering stiffness values of each front and rear tire, respectively, δ is the steering angle, v_x and v_y are the velocities of each x and y axes, respectively, l_f and l_r are the length of each front and rear wheel axis from the center of gravity, and ψ is the yaw rate.

The vehicle pitch dynamics can be written as [29]:

$$\Sigma M_y = -k_s l_f^2 \sin\theta - k_s l_r^2 \sin\theta - b_s l_f^2 \theta \cos\theta - b_s l_r^2 \theta \cos\theta - (\Sigma F_{drive} - \Sigma F_{brake}) h_{CG}$$
(3)

$$\Sigma F_{drive} = \left[\frac{N_{tf} \eta_{tf}}{r_{eff}} T_e - \left\{ (I_e + I_t) N_{tf}^2 + I_d N_f^2 + I_w \right\} \frac{a_x}{r_{eff}^2} \right]$$
(4)

$$\Sigma F_{brake} = \frac{2P_{brake}A_{caliper}\mu_{pad}R_b}{r_{eff}}$$
(5)

where M_{y} is the moment with respect to the pitch axis and θ is the pitch of the vehicle.

The traction force applied to the vehicle can be expressed by (4) based on the vehicle drive-line dynamics. N_{tf} is the gear ratio of the transmission and differential gears, η_{tf} is the efficiency of the power transfer from the engine to the wheel axis, r_{eff} is the effective radius of the tire, T_e is the engine torque, I_e , I_t , I_d , I_w are the inertia of each engine, transmission, differential gear, and wheel, respectively, and a_x is the longitudinal acceleration of the vehicle.

Brake force applied to the vehicle is given by (5). P_{brake} is the pressure of the master cylinder, $A_{caliper}$ is the area of the brake pad caliper, μ_{pad} is the friction coefficient of the brake pad, and R_b is the distance from the wheel center to the brake pad.

Based on (1)–(5), we can choose only variables, except for static parameters such as the vehicle mass. As a result, a total of 13 feature candidates were selected, including the acceleration and yaw rate. Subsequently, seven features were selected considering the availability of sensors in the experimental vehicle. Table 2 shows the selected features. Based on these, we conducted an analysis of importance based on the attention mechanism [30]. The results of the importance analysis are presented in Appendix A. According to Appendix A, the final features were selected same as Table 2.

Table 2. Selected features considering sensor usability.

Feature Name	Description
Ax	Vehicle longitudinal acceleration from IMU
Ау	Vehicle lateral acceleration from IMU
Yaw Rate	Vehicle yaw rate from IMU
Brake Pres	Brake pressure from the master cylinder
Str Angle	Steering wheel angle
Throttle	Engine throttle valve opening degree (0~1)
Σ WheelSpd (Vx)	Sum of the wheel rotation speed

3.1.2. Network Design

Before configuring a 2-d input to the CNN, we calculate $\phi_{static,pseudo}$ and $\theta_{static,pseudo}$ reflecting the static roll and pitch angles using feature sensor values. The pseudo values are written as:

$$\phi_{static,pseudo} = e^{-a \cdot V_x} \cdot \sin^{-1} \left(\frac{a_y}{g} \right) \tag{6}$$

$$\theta_{static,pseudo} = e^{-b \cdot V_x} e^{-c \cdot \theta_{throttle}} e^{-\frac{d}{P_{brake}}} \cdot \sin^{-1} \left(-\frac{a_x}{g}\right)$$
(7)

where a,b,c,d are the constant design parameters. (5) and (6) are similar to the road bank and the slope angle, respectively when the vehicle's wheel speed is zero. If the vehicle's wheel speed increases, (5) and (6) become zero.

The neural network architecture is shown in Figure 2. The network is composed of CNN part and fully connected layer (FCL) part in parallel configuration. The input array is composed of a mux of each sequential feature sensor data including (5), (6). The sequential information is 2 s for 0.01 s, therefore the input array size is (200×9) . CNN part uses all inputs, namely all-time series data in the past 2 s, but FCL part uses only the last row of the input array, meaning only current step data. This means that the CNN part is designed with the intention of estimating dynamic vehicle body changes while driving and FCL part is designed to estimate the roll and pitch angles in static scenarios. The CNN part comprises four convolution layers and two fully connected layers. The first layer converts the input matrix into a square matrix and shuffles the sensor placement order. Then, it passes through the three convolution layers, and then pass the one convolution layer with large size filter and wide strides for compressing the data. Next, unfold to fully connected layer and make the last layer's size (256×1). In FCL part, there are four layers and final layer's size is also (256×1). The two final layers are concatenated, and then the regression model is constructed, which calculates two outputs using one fully connected layer. Hyperparameters of the neural network such as the number of filters or activation function are described in Appendix B.



Figure 2. Neural network architecture.

3.2. Dual Extended Falman Filter Design

3.2.1. State Space Model

The state space model is based on the vehicle 6-DOF sprung mass model [31,32] for expressing the six-dimensional acceleration and angular velocity. The dynamics are composed based on the Euler rigid body equation and the force or moment of each axis which is given by each dynamic [29]. Figure 3 shows the six-dimensional motion of vehicle sprung mass. The forces or moment of each axis are given by:

$$\Sigma F_x = m_s \left(\dot{v_x} + v_z \dot{\theta} - v_y \dot{\psi} - h_R \ddot{\theta} + h_R \dot{\phi} \dot{\psi} \right) \tag{8}$$

$$\Sigma F_y = m_s \left(\dot{v_y} + v_x \dot{\psi} - v_z \dot{\phi} - h_R \ddot{\phi} + h_R \dot{\theta} \dot{\psi} \right) \tag{9}$$

$$\Sigma F_z = m_s \left(\dot{v_z} + v_y \dot{\phi} - v_x \dot{\theta} - h_R \dot{\phi}^2 - h_R \dot{\theta}^2 \right)$$
(10)

$$\Sigma M_x = I_x \ddot{\phi} + (I_z - I_y) \dot{\theta} \dot{\psi} - m_s h_R \left(\dot{v_y} + v_x \dot{\psi} - v_z \dot{\phi} - h_R \ddot{\phi} + h_R \dot{\theta} \dot{\psi} \right)$$
(11)

$$\Sigma M_{y} = I_{y}\ddot{\theta} + (I_{x} - I_{z})\dot{\phi}\dot{\psi}$$
(12)

$$\Sigma M_z = I_z \ddot{\psi} + (I_y - I_x) \dot{\phi} \dot{\theta}$$
(13)

where ΣF_x , ΣF_y , ΣF_z are the sum of forces of the *x*, *y*, and *z* axes, respectively. ΣM_x , ΣM_y , ΣM_z are the sum of moments of the *x*, *y*, and *z* axes, respectively. I_x , I_y , I_z are the moments of inertia along the *x*, *y*, and *z* axes, respectively. ΣF_x , ΣF_y , ΣF_z , ΣM_x , ΣM_y , and ΣM_z can be derived from a vehicle model such as a bicycle model. Therefore, the nonlinear state space equation can be written as:

$$\dot{x}_{1} = \frac{\left(\frac{N_{tf}\eta_{tf}}{r_{eff}}u_{1} - \Sigma F_{brake} - \frac{1}{2}\rho C_{d}Ax_{1}^{2} - m_{s}gsin(x_{8})\right)}{\left\{m_{s} + \frac{(I_{e}+I_{l})N_{tf}^{2} + I_{d}N_{f}^{2} + I_{w}}{r_{eff}^{2}}\right\}} - x_{3}x_{5} + x_{2}x_{6} + h_{R}\dot{x}_{5} - h_{R}x_{4}x_{6}$$
(14)

$$\dot{x}_2 = \frac{1}{m_s} \{ \Sigma F_y + m_s gsin(x_7) \} - x_1 x_6 + x_3 x_4 + h_R \dot{x}_4 - h_R x_5 x_6$$
(15)

$$\dot{x}_3 = \frac{1}{m_s} k_s \sin(x_8) \left(l_f - l_r \right) + b_s \cos(x_8) x_5 \left(l_f - l_r \right) - x_2 x_4 + x_1 x_5 + h_R x_4^2 + h_R x_5^2$$
(16)

$$\dot{x}_{4} = \frac{1}{I_{x} + m_{s}h_{R}^{2}} \left\{ h_{R}m_{s}gsin(x_{7}) - \frac{1}{2}k_{s}l_{s}^{2}sin(x_{7}) - \frac{1}{2}b_{s}l_{s}^{2}x_{4}\cos(x_{7}) + \Sigma F_{y}(h_{CG} - h_{R}) - (I_{z} - I_{y})x_{5}x_{6} + m_{s}h_{R}(\dot{x}_{2} + x_{1}x_{6} - x_{3}x_{4} + h_{R}x_{5}x_{6}) \right\}$$
(17)

$$\dot{x}_{5} = \frac{1}{I_{y}} \left\{ - \left(l_{f}^{2} + l_{r}^{2} \right) k_{s} \sin(x_{8}) - \left(l_{f}^{2} + l_{r}^{2} \right) b_{s} x_{5} \cos(x_{8}) - (\Sigma F_{drive} - \Sigma F_{brake}) h_{CG} - (I_{x} - I_{z}) x_{5} x_{6} \right\}$$
(18)

$$\dot{x}_{6} = \frac{1}{I_{z}} \left\{ -\frac{2l_{f}C_{\alpha f} - 2l_{r}C_{\alpha r}}{x_{1}}x_{2} - \frac{2l_{f}^{2}C_{\alpha f} + 2l_{r}^{2}C_{\alpha r}}{x_{1}}x_{6} + 2l_{f}C_{\alpha f}u_{3} - (I_{y} - I_{x})x_{4}x_{5} \right\}$$
(19)

$$\dot{x}_7 = x_4 \tag{20}$$

$$\dot{x}_8 = x_5 \tag{21}$$



Figure 3. Vehicle sprung mass 6 degree of freedom motion.

 ΣF_{drive} , ΣF_{brake} , and ΣF_y have been described previously. The state vector $x = \begin{bmatrix} v_x v_y v_z \dot{\phi} \dot{\theta} \dot{\psi} \phi \theta \end{bmatrix}$, state input $u = [T_e P_{brake} \delta]$. ρ is the density of the air, C_d is the drag coefficient and A is the frontal area of the vehicle.

$$a_{x} = \frac{\left(\frac{N_{tf}\eta_{tf}}{r_{eff}}u_{1} - \Sigma F_{brake} - \frac{1}{2}\rho C_{d}Ax_{1}^{2} - m_{s}gsin(x_{8})\right)}{\left\{m_{s} + \frac{(I_{e}+I_{t})N_{tf}^{2} + I_{d}N_{f}^{2} + I_{w}}{r_{eff}^{2}}\right\}}$$
(22)

$$a_y = \frac{1}{m_s} \left\{ \Sigma F_y + m_s gsin(x_7) \right\}$$
(23)

 $\dot{\psi} = x_6 \tag{24}$

 $\phi = x_7 \tag{25}$

$$\theta = x_8 \tag{26}$$

3.2.2. Observability Check

Before designing the estimator, the observability must be checked. The observability of the nonlinear state space model can be checked by the Lie derivative [33]. When the state space equation is expressed as $\dot{x} = f(x, u)$, y = h(x), the Lie derivative and observability matrix can be written as:

$$L_f^0 = h(x)L_f^{k+1} = \frac{\partial L_f^k}{\partial x}f = \nabla L_f^k \cdot f$$
(27)

$$O = \begin{bmatrix} \nabla L_f^0 \\ \nabla L_f^1 \\ \vdots \\ \nabla L_f^{n-1} \end{bmatrix}_{x=x_0}$$
(28)

where *O* is the observability matrix. Using the rank of the observability matrix, the system's observability can be seen locally. As a result of checking the rank of the observability matrix, the observability matrix has full rank in range of $v_x \neq 0$; therefore, this system is locally observable for the range except $v_x = 0$.

3.2.3. Dual Extended Kalman Filter Module

Among the vehicle dynamics parameters, the cornering stiffness varies under conditions such as the vehicle load. To reduce the errors while modeling, the cornering stiffness should be estimated.

This study adopts the DEKF as an estimator for reducing the error and increasing the modeling accuracy. Figure 4 shows the DEKF scheme. The state vector, state input, state output, and state space equation have been discussed in Section 3.2.1. The DEKF module works according to the following recursive algorithm [34]:



Figure 4. Scheme of dual-extended Kalman filter (DEKF).

Parameter prediction:

$$\hat{x}_{p}^{-}(t) = \hat{x}_{p}(t-1) \tag{29}$$

$$P_p^{-}(t) = P_p(t-1) + Q_p \tag{30}$$

State prediction:

$$\hat{x}_{s}^{-}(t) = f\left(\hat{x}_{s}(t-1), u(t), \hat{x}_{p}^{-}(t)\right)$$
(31)

$$P_s^{-}(t) = F_s(t)P_s(t-1)F_s^{T}(t) + Q_s$$
(32)

State update:

$$K_{s}(t) = P_{s}^{-}(t)H_{s}^{T}(t)\left[H_{s}(t)P_{s}^{-}(t)H_{s}^{T}(t) + R_{s}\right]^{-1}$$
(33)

$$\hat{x}_{s}(t) = \hat{x}_{s}^{-}(t) + K_{s}(t) \left[y(t) - h \left(\hat{x}_{s}^{-}(t) \right) \right]$$
(34)

$$P_s(t) = P_s^-(t) - K_s(t)H_s(t)P_s^-(t)$$
(35)

Parameter update:

$$K_{p}(t) = P_{p}^{-}(t)H_{p}^{T}(t)\left[H_{p}(t)P_{p}^{-}(t)H_{p}^{T}(t) + R_{p}\right]^{-1}$$
(36)

$$\hat{x}_{p}(t) = \hat{x}_{p}^{-}(t) + K_{p}(t) \left[y(t) - h \left(\hat{x}_{s}^{-}(t) \right) \right]$$
(37)

$$P_p(t) = P_p^{-}(t) - K_p(t)H_p(t)P_p^{-}(t)$$
(38)

where the parameter vector $\hat{x}_p = \begin{bmatrix} C_{\alpha f} & C_{\alpha r} \end{bmatrix}^T$, state vector $\hat{x}_s = \begin{bmatrix} v_x & v_y & v_z & \dot{\phi} & \dot{\theta} & \psi & \phi \end{bmatrix}^T$; P_p and P_s are the error covariance matrices for parameters and states, respectively; Q_p and Q_s are the process noise covariance matrices for parameter and state estimators, respectively; R_p and R_s are the output noise covariance matrices for parameter and state estimators, respectively. R_p and R_s are the same because both the parameter and state estimator have same output y. K_p and K_s are the Kalman gain matrices for parameter and state estimators, respectively. F_s and H_s are the Jacobian matrices for the state and output equations, respectively, and are expressed as follows:

$$F_{s} = \begin{bmatrix} \frac{\partial f_{1}}{\partial x_{1}} & \cdots & \frac{\partial f_{1}}{\partial x_{8}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{8}}{\partial x_{1}} & \cdots & \frac{\partial f_{8}}{\partial x_{8}} \end{bmatrix}$$
(39)

$$H_{s} = \begin{bmatrix} \frac{\partial h_{1}}{\partial x_{1}} & \cdots & \frac{\partial h_{1}}{\partial x_{8}} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_{5}}{\partial x_{1}} & \cdots & \frac{\partial h_{5}}{\partial x_{8}} \end{bmatrix}$$
(40)

In Equations (36) and (38), H_p is the Jacobian matrix of state output for the parameter estimator and can be expressed as:

$$H_{p} = \frac{\partial y}{\partial \hat{x}_{p}} = \begin{bmatrix} \frac{\partial a_{x}}{\partial C_{\alpha f}} & \frac{\partial a_{x}}{\partial C_{\alpha r}} \\ \vdots \\ \frac{\partial \theta}{\partial C_{\alpha f}} & \frac{\partial \theta}{\partial C_{\alpha r}} \end{bmatrix}$$
(41)

 R_p and R_s can be determined by the engineer's tuning based on sensor noise. Q_s and Q_p can also be determined by engineer's tuning but Q_p 's square of each elements is tuned to approximately 1% of the initial values of the each actual parameter.

4. Results and Analysis

This section presents and discusses the experimental verification results of the algorithms mentioned in Section 3. Neural network and DEKF are discussed separately in Sections 3.1 and 3.2, respectively. The performance is compared with that of commercial sensors.

4.1. Roll and Pitch Estimator (Neural Network)

4.1.1. Dataset

Sensor data from real-car experiment data were used for training and validating the neural network. The neural network input data set contained information of a car's chassis sensors, and the label data set of roll and pitch angles was obtained from the high-accuracy GPS-inertial navigation system (INS) RT-3002 (Oxford Technical Solutions Ltd., Bicester, UK). For training the neural network, a total of 176,259 data sets were used, which logged about 30 min at 10 ms intervals in various situations and offline validation was performed in scenarios as shown in Table 3 with the same vehicle. The software used Python and the framework used TensorFlow 1.6.

Table 3. Roll/pitch estimator validation scenario.

	Condition	Description
Case 1	-	A stationary situation on the uphill
Case 2	Acceleration ± 0.5 g or higher Steering ± 5 deg or higher	Rapid acceleration with steering on downhill slope-quick deceleration
Case 3	Steering $\pm 50 \deg$ or higher Yaw rate $\pm 30 \deg/s$ or higher	Accelerate-turn-deceleration with steering on flat road
Case 4	-	Common driving

4.1.2. Validation Result and Analysis

The estimation performance was validated using offline sensor data logged in various cases, as shown in Table 3. The root mean square error (RMSE) was calculated by comparison with RT-3002, which was treated as a reference, and with the datasheet of SST810, which is a commercial inclinometer sensor (Vigor Technology Co., Ltd., Xiamen, China). Figures 5–8 show the scenario and roll/pitch estimation results for Case 1–4, respectively.



Figure 5. Validation results of Case 1. (a) Speed and steering profile. (b) Roll/pitch estimation results.



Figure 6. Validation results of Case 2. (a) Speed and steering profile. (b) Roll/pitch estimation results.



Figure 7. Validation results of Case 3. (a) Speed and steering profile. (b) Roll/pitch estimation results.



Figure 8. Validation results of Case 4. (a) Speed and steering profile. (b) Roll/pitch estimation results.

Table 4 shows the accuracy of the SST810 datasheet and RMSE calculation results of the estimation results for Case 1–4. Figures 5–8 show that the value of pitch and roll angles between 0 and 2 s is fixed at zero. This is due to the structure described in Section 3.1.2, requiring 2 s of sequential data for the input; thus, the values in the initial 2 s cannot be calculated. The RMSE was therefore calculated in the time zone excluding the initial 2 s.

	Case 1		Ca	Case 2 Ca		ase 3 Case 4		se 4
	Roll	Pitch	Roll	Pitch	Roll	Pitch	Roll	Pitch
RMSE (deg) Commercial sensor accuracy (deg) ¹	$0.1133 \leq \pm$ (static si	0.4188 0.05 tuation)	0.0573	0.5422	0.1359 ≤∃ (dynamic	0.0958 ₌0.5 situation)	0.5140	0.5283

Table 4. Root mean square error (RMSE) calculation results of roll/pitch estimator and accuracy of SST810 datasheet.

¹ From SST810 inclinometer datasheet of Vigor Technology Co., Ltd.

Case 1 shows that the roll RMSE is approximately 0.1° and the pitch RMSE is approximately 0.4°. There was some offset error, although the vehicle's speed was zero, i.e., a stationary scenario. For commercial sensors, the error rate is 0.05° at the static scenario, but the estimation results show a larger error compared with the commercial sensors. The neural network estimator in this study uses data of only the chassis sensors of the vehicle; thus, the performance of sensors has a significant impact on the estimation performance. In particular, it can be expected that the IMU's characteristic bias error affected the offset errors in the estimation results.

Cases 2–4 include scenarios in which rapid changes in vehicle attitude occur in areas where acceleration or deceleration occurs. The validation results show high estimation accuracy in these scenarios. In addition, the noise reduction effect compared to RT-3002 is noticeable in the 15–60 s duration for Case 3; thus, the neural network estimator can estimate stable output. The performance of the estimator is thus superior compared to the commercial sensors.

4.2. Acceleration and Angular Velocity Estimator (DEKF)

4.2.1. Validation Environment

The acceleration and angle velocity estimator was validated by simulation using CarSim, which is a commercial vehicle simulator software. To create an environment similar to the actual vehicle, Gaussian white noise was added to the sensor value, as shown in Table 5. To ensure that the errors of the acceleration sensor due to roll and pitch have been corrected, simulations were conducted in the scenarios shown in Table 6.

Table 5. Sensor configuration for simulation.

Sensor	Noise	Unit	
IMU (a_x, a_y)	0.1 (RMS) + 10 (%)	m/s ²	
IMU $(\dot{\psi})$	0.01 (RMS) + 10 (%)	rad/s	
Steering angle	0.05 (RMS) + 10 (%)	rad	
Engine torque	7 (RMS)	N⋅m	
Brake pressure	0.05 (RMS)	MPa	

Table 6. Acceleration/angular velocity estimator validation scenario.

	Condition	Description
Case 1	Roll $\pm 30 \text{ deg or lower}$ Pitch $\pm 3 \text{ deg or lower}$	U-turn with 30 degrees of bank angle
Case 2	Roll $\pm 30 \deg$ or lower Pitch $\pm 10 \deg$ or lower	Sharp turn at 30 degrees of bank angle after 10 degrees of uphill and downhill
Case 3	-	Common driving

4.2.2. Validation Results and Analysis

The correctness performance of a_x , a_y , and ψ was validated by comparing IMU sensor values, DEKF estimates, and reference values, while the estimates of roll rates and pitch rates without sensors were validated by comparing with only the reference values. The RMSE of the estimated results is calculated based on the reference and compared with the datasheet of SMI860, which is a commercial six-dimensional IMU from BOSCH Co., Ltd. In addition, to validate the effects of DEKF's parameter estimator, the estimated results of $C_{\alpha f}$ and $C_{\alpha r}$ for each case were noted, and the estimated results with and without cornering stiffness estimation for Case 3 were compared. Figures 9–12 show the simulation scenarios and results of DEKF.



Figure 9. Simulation results of Case 1. (a) Roll/pitch profile. (b) a_x , a_y and ψ estimation results. (c) ϕ and θ estimation results. (d) $C_{\alpha f}$ and $C_{\alpha r}$ estimation results.



Figure 10. Simulation results of Case 2. (a) Roll/pitch profile. (b) a_x , a_y , and ψ estimation results. (c) ϕ and θ estimation results. (d) $C_{\alpha f}$ and $C_{\alpha r}$ estimation results.



Figure 11. Simulation results of Case 3. (a) Roll/pitch profile. (b) a_x , a_y , and $\dot{\psi}$ estimation results. (c) $\dot{\phi}$ and $\dot{\theta}$ estimation results. (d) $C_{\alpha f}$ and $C_{\alpha r}$ estimation results.



Figure 12. Effect of cornering stiffness estimation in Case 3.

Table 7 shows the accuracy of the commercial sensor, obtained from its datasheet, and RMSE calculation results of the estimation values for case 1–3. In case 1, there was a fatal error of sensor value of a_y due to roll angle, and case 2 exhibited considerably varying roll and pitch angles; therefore, the value of a_x and a_y from sensors may not be accurate. Figures 9b and 10b confirm that these errors in sensor values are successfully corrected to obtain estimates close to the reference. In addition, in normal driving scenarios, as in the 60–65 s duration in Case 3, an a_x sensor error by pitch is noted, which has also been successfully corrected. Furthermore, in all cases, filter effects that reduce noise from existing sensors can be checked through a_x , a_y , and ψ . Compared with a commercial sensor, a_x and a_y show similar accuracy, whereas ψ shows a significantly higher accuracy. However, in the case of commercial sensors, there may be errors caused by roll and pitch angles; therefore, DEKF can have commercial sensor level or higher accuracy even though correcting those errors.

		Accuracy	
		DEKF (RMSE)	Commercial Sensor ¹
	Case 1	0.4325	
$a_x (m/s^2)$	Case 2	0.8075	$\leq \pm 0.5$
	Case 3	0.3087	
	Case 1	0.8232	
a_{ν} (m/s ²)	Case 2	0.4204	$\leq \pm 0.5$
y ()	Case 3	0.5085	
	Case 1	0.7391	
ψ (deg/s)	Case 2	0.3953	$\leq \pm 3$
1 (0.)	Case 3	0.5844	
	Case 1	5.8499	
ϕ (deg/s)	Case 2	5.1394	$\leq \pm 2$
1 (0, -)	Case 3	1.0542	
	Case 1	1.9251	
$\dot{\theta}$ (deg/s)	Case 2	1.7475	_ 2
(0, -)	Case 3	0.6704	

Table 7. RMSE calculation results of DEKF and accuracy obtained from SMI860 datasheet.

¹ From SMI860 IMU datasheet of BOSCH Co., Ltd. ² SMI860 cannot sense the pitch rate.

The roll and pitch rates could be estimated because they were included in the state vector, although the sensor values were not included. The roll rate accuracy was occasionally lower than that of commercial sensors depending on the case, and in the case of pitch rate, it is not comparable because there is no output of commercial sensors, but it can be confirmed from the RMSE values that the estimated values have a fairly high accuracy.

Figures 9d, 10d and 11d show the estimated cornering stiffness. Figure 12 and Table 8 show the results with and without cornering stiffness estimation in Case 3, which can improve accuracy by approximately 3–5%, particularly with a greater effect on a_y .

Table 8. RMSE of with and without cornering stiffness estimation in Case 3.

	RMS	SE
	Without Cornering Stiffness Estimation	With Cornering Stiffness Estimation
$a_y (\mathrm{m/s^2})$	0.5541	0.5085
ψ (deg/s)	0.6131	0.5844

5. Conclusions and Future Work

In this paper, we proposed a CNN-based neural network to estimate the roll and pitch angles of a vehicle. A DEKF was used to correct the gravitational effect caused by the roll and pitch for estimating the exact acceleration and angular velocity.

By using the vehicle's chassis sensor data as a time series, the neural network could estimate the roll and pitch angles of the vehicle without a GPS or six-dimensional IMU. Based on the estimated roll and pitch angels, we designed an extended Kalman filter (EKF) using the 6-DOF vehicle model. Another EKF was designed, and the two EKFs were used to estimate the cornering stiffness. We then constructed the DEKF.

Using experimental data obtained using a real car, the proposed roll and pitch estimator was validated, and the DEKF was validated in the CarSim simulation environment. The roll and pitch estimator showed an improved performance compared to the commercial sensors in dynamic scenarios and also reduced the noise. However, the performance in static scenarios was weaker. The acceleration and angular velocity estimator could effectively correct the acceleration sensor error due to roll and pitch with a de-noising effect. In addition, the roll and pitch rates that could not be obtained from sensors could be estimated with significant accuracy. By comparing the results before and after including the cornering stiffness, we found that the accuracy is improved if the cornering stiffness is considered.

On the other hand, our work has limitations and challenges that should be further discussed. We plan to consider fusion with other algorithms to improve the attitude estimation performance in static scenarios. In addition, the proposed method has not been checked in case of a change in the vehicle weight. It will also be necessary to verify the performance of the algorithm due to changes in vehicle weight. Furthermore, our proposed algorithm is hard to apply as an embedded system in vehicle because the neural network has large capacity. The future work should be conducted to enable algorithm to operate as real-time in vehicle through simplification and optimization.

Author Contributions: Conceptualization, M.O. and S.O.; methodology, M.O.; software, M.O.; validation, M.O. and S.O.; formal analysis, M.O. and S.O.; investigation, M.O.; resources, J.H.P.; data curation, S.O.; writing—original draft preparation, M.O.; writing—review and editing, S.O. and J.H.P.; visualization, M.O.; supervision, J.H.P.; project administration, S.O. and J.H.P.; funding acquisition, J.H.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Hyundai Motor Group Academy Industry Research Collaboration.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Restrictions apply to the availability of these data. Data was obtained from Hyundai Motor Company and are available with the permission of Autonomous Driving Center of Hyundai Motor Company R&D Division.

Acknowledgments: This work was supported by Autonomous Driving Center of Hyundai Motor Company R&D Division.

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

Appendix A. Feature Importance Analysis



Figure A1. Result of feature importance weight analysis.

Table A1. Feature name of each number

Feature Number	Feature Name	
1	Ax	
2	Ау	
3	Yaw Rate	
4	Brake Pressure	
5	Steering Angle	
6	Engine Throttle	
7	Σ Wheel Speed (Vx)	

All features have an importance weight higher than 0.8; therefore, feature reduction is not performed.

Appendix B. Hyperparameter of Neural Network

Part	Layer	Name	Filter Size	Activation	Output Size
	1-layer	Fully connected		Leaky ReLU	$(200 \times 200 \times 1)$
	2-layer	Convolution layer	(3 $ imes$ 3 $ imes$ 1) 8EA 1	Leaky ReLU	$(200 \times 200 \times 8)$
CNN	3-layer	Convolution layer	$(3 \times 3 \times 8)$ 16EA 1	Leaky ReLU	$(200 \times 200 \times 16)$
part	4-layer	Convolution layer	$(3 \times 3 \times 16)$ 32EA	Leaky ReLU	(200 × 200 × 32)
	5-layer	Convolution layer	$(5 \times 5 \times 32)$ 32EA	Leaky ReLU	$(40 \times 40 \times 32)$
	6-layer	Fully connected		Leaky ReLU	(256 × 1)
	1-layer	Fully connected		Leaky ReLU	(256×1)
FCL	2-layer	Fully connected		Leaky ReLU	(1024×1)
part	3-layer	Fully connected		Leaky ReLU	(1024×1)
	4-layer	Fully connected		Leaky ReLU	(256×1)
Final Layer	Final layer	Fully connected			(2 × 1)

Table A2. Details of neural network.

 1 1 \times 1 stride was used. 2 5 \times 5 strides were used.

References

- 1. Lee, H. Reliability indexed sensor fusion and its application to vehicle velocity estimation. *J. Dyn. Sys. Meas. Control* 2006, 128, 236–243. [CrossRef]
- Chu, L.; Shi, Y.; Zhang, Y.; Liu, H.; Xu, M. Vehicle lateral and longitudinal velocity estimation based on Adaptive Kalman Filter. In Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), Chengdu, China, 20–22 August 2010. [CrossRef]

- Wu, L.-J. Experimental study on vehicle speed estimation using accelerometer and wheel speed measurements. In Proceedings of the 2011 Second International Conference on Mechanic Automation and Control Engineering, Inner Mongolia, China, 15–17 July 2011. [CrossRef]
- 4. Klomp, M.; Gao, Y.; Bruzelius, F. Longitudinal velocity and road slope estimation in hybrid electric vehicles employing early detection of excessive wheel slip. *Veh. Syst. Dyn.* **2014**, 52 (Suppl. S1), 172–188. [CrossRef]
- Song, C.K.; Uchanski, M.; Hedrick, J.K. Vehicle speed estimation using accelerometer and wheel speed measurements. In Proceedings of the SAE International Body Engineering Conference and Automotive & Transportation Technology Conference, Paris, France, 9–11 July 2002. [CrossRef]
- Jin, X.; Yin, G.; Chen, N. Advanced estimation techniques for vehicle system dynamic state: A survey. Sensors 2019, 19, 4289. [CrossRef] [PubMed]
- 7. Zhang, X.; Zhao, J.; LeCun, Y. Character-level convolutional networks for text classification. *arXiv* 2015, arXiv:1509.01626.
- Eric Tseng, H.; Xu, L.; Hrovat, D. Estimation of land vehicle roll and pitch angles. *Veh. Syst. Dyn.* 2007, 45, 433–443. [CrossRef]
 Xiong, L.; Xia, X.; Lu, Y.; Liu, W.; Gao, L.; Song, S.; Han, Y.; Yu, Z. IMU-Based Automated Vehicle Slip Angle and Attitude Estimation Aided by Vehicle Dynamics. *Sensors* 2019, *19*, 1930. [CrossRef]
- 10. Lee, J.K.; Park, E.J.; Robinovitch, S.N. Estimation of attitude and external acceleration using inertial sensor measurement during various dynamic conditions. *IEEE Trans. Instrum. Meas.* **2012**, *61*, 2262–2273. [CrossRef]
- Suh, Y.-S.; Park, S.-K.; Kang, H.-J.; Ro, Y.-S. Attitude estimation adaptively compensating external acceleration. JSME Int. J. Ser. C Mech. Syst. Mach. Elem. Manuf. 2006, 49, 172–179. [CrossRef]
- 12. Ahmed, H.; Tahir, M. Accurate attitude estimation of a moving land vehicle using low-cost MEMS IMU sensors. *IEEE Trans. Intell. Transp. Syst.* **2016**, *18*, 1723–1739. [CrossRef]
- 13. Scholte, W.J.; Marco, V.R.; Nijmeijer, H. Experimental Validation of Vehicle Velocity, Attitude and IMU Bias Estimation. *IFAC-PapersOnLine* **2019**, *52*, 118–123. [CrossRef]
- 14. Caron, F.; Duflos, E.; Pomorski, D.; Vanheeghe, P. GPS/IMU data fusion using multisensor Kalman filtering: Introduction of contextual aspects. *Inf. Fusion* **2006**, *7*, 221–230. [CrossRef]
- 15. 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. [CrossRef]
- 16. Ryu, J.; Rossetter, E.J.; Gerdes, J.C. Vehicle sideslip and roll parameter estimation using GPS. In Proceedings of the AVEC International Symposium on Advanced Vehicle Control, Hiroshima, Japan, 9–13 September 2002.
- Ahmad, I.; Benallegue, A.; El Hadri, A. Sliding mode based attitude estimation for accelerated aerial vehicles using GPS/IMU measurements. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013. [CrossRef]
- 18. 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. [CrossRef]
- 19. Garcia Guzman, J.; Prieto Gonzalez, L.; Pajares Redondo, J.; Sanz Sanchez, S.; Boada, B.L. Design of Low-Cost Vehicle Roll Angle Estimator Based on Kalman Filters and an IoT Architecture. *Sensors* **2018**, *18*, 1800. [CrossRef] [PubMed]
- 20. Park, M.; Yim, S. Design of Robust Observers for Active Roll Control. IEEE Access 2019, 7, 173034–173043. [CrossRef]
- 21. Chung, T.; Yi, S.; Yi, K. Estimation of vehicle state and road bank angle for driver assistance systems. *Int. J. Automot. Technol.* 2007, *8*, 111–117.
- 22. Tseng, H.E. Dynamic estimation of road bank angle. Veh. Syst. Dyn. 2001, 36, 307–328. [CrossRef]
- 23. Oh, J.; Choi, S.B. Vehicle roll and pitch angle estimation using a cost-effective six-dimensional inertial measurement unit. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2013**, 227, 577–590. [CrossRef]
- 24. Kamal Mazhar, M.; Khan, M.J.; Bhatti, A.I.; Naseer, N. A Novel Roll and Pitch Estimation Approach for a Ground Vehicle Stability Improvement Using a Low Cost IMU. *Sensors* 2020, 20, 340. [CrossRef]
- 25. Vargas-Meléndez, L.; Boada, B.L.; Boada, M.J.L.; Gauchía, A.; Díaz, V. A sensor fusion method based on an integrated neural network and Kalman filter for vehicle roll angle estimation. *Sensors* **2016**, *16*, 1400. [CrossRef]
- Vargas-Melendez, L.; Boada, B.L.; Boada, M.J.L.; Gauchia, A.; Diaz, V. Sensor Fusion based on an integrated neural network and probability density function (PDF) dual Kalman filter for on-line estimation of vehicle parameters and states. *Sensors* 2017, 17, 987. [CrossRef] [PubMed]
- 27. González, L.P.; Sánchez, S.S.; Garcia-Guzman, J.; Boada, M.J.L.; Boada, B.L. Simultaneous Estimation of Vehicle Roll and Sideslip Angles through a Deep Learning Approach. *Sensors* 2020, 20, 3679. [CrossRef]
- 28. Taehui, L.; Sang Won, Y. Estimation of Vehicle Roll and Road Bank Angle based on Deep Neural Network. In Proceedings of the KSAE 2018 Annual Autumn Conference & Exhibition, Jeongseon, Korea, 14–17 November 2018.
- 29. Rajamani, R. Vehicle Dynamics and Control, 2nd ed.; Springer Science & Business Media: New York, NY, USA, 2011.
- Challita, N.; Khalil, M.; Beauseroy, P. New feature selection method based on neural network and machine learning. In Proceedings of the 2016 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET), Beirut, Lebanon, 2–4 November 2016. [CrossRef]
- 31. Yang, S.; Kim, J. Validation of the 6-dof vehicle dynamics model and its related VBA program under the constant radius turn manoeuvre. *Int. J. Automot. Technol.* **2012**, *13*, 593–605. [CrossRef]

- 32. Ray, L.R. Nonlinear tire force estimation and road friction identification: Simulation and experiments. *Automatica* **1997**, *33*, 1819–1833. [CrossRef]
- 33. Lee, D.-H.; Kim, I.-K.; Huh, K.-S. Tire Lateral Force Estimation System Using Nonlinear Kalman Filter. *Trans. Korean Soc. Automot. Eng.* **2012**, 20, 126–131. [CrossRef]
- 34. Wenzel, T.A.; Burnham, K.; Blundell, M.; Williams, R. Dual extended Kalman filter for vehicle state and parameter estimation. *Veh. Syst. Dyn.* **2006**, *44*, 153–171. [CrossRef]