processes

Article

Transformer Aided Adaptive Extended Kalman Filter for Autonomous Vehicle Mass Estimation

Hui Zhang [1,2], Zichao Yang [1,2], Huiyuan Xiong [1,2,*], Taohong Zhu [1,2], Zhineng Long [1] and Weibin Wu [3]

[1] Guangdong Provincial Key Laboratory of Intelligent Transportation System, School of Intelligent Systems Engineering, Sun Yat-sen University, Shenzhen 518107, China; yangzch5@mail2.sysu.edu.cn (Z.Y.)
[2] Dongguan Institute, Sun Yat-sen University, Dongguan 523808, China
[3] College of Engineering, South China Agricultural University, Guangzhou 510642, China
* Correspondence: xionghy@mail.sysu.edu.cn; Tel.: +86-755-2326-0062

Abstract: Vehicle mass is crucial to autonomous vehicles control. Affected by the nonlinearity of vehicle dynamics between vehicle states, it is still a tough issue to estimate vehicle mass precisely and stably. The transformer aided adaptive extended Kalman filter is proposed to further improve the accuracy and stability of estimation. Firstly, the transformer-based estimator is introduced to provide an accurate pre-estimation of vehicle mass, with the nonlinear dynamics among vehicle states being learned. Secondly, on the basis of comparing the real-time input and training data of neural network, the weight adjustment module is designed to present an adaptive law. Finally, the adaptive extended Kalman filter is proposed to meet the demand of accuracy and stability, where the pre-estimation of transformer-based estimator is integrated with the adaptive law. Dataset is collected by conducting heavy-duty vehicle simulation. The mean absolute percentage error, mean absolute error, root mean square error and convergence rate averaged over simulation tests are 0.90%, 256.47 kg, 357.01 kg and 184 steps, respectively. The results show the outperformance of the proposed method in terms of accuracy and stability.

Keywords: mass estimation; adaptive extended Kalman filter; transformer; autonomous vehicle

check for updates

Citation: Zhang, H.; Yang, Z.; Xiong, H.; Zhu, T.; Long, Z.; Wu, W. Transformer Aided Adaptive Extended Kalman Filter for Autonomous Vehicle Mass Estimation. *Processes* **2023**, *11*, 887. https://doi.org/10.3390/pr11030887

Academic Editor: Wen-Jer Chang

Received: 24 February 2023
Revised: 12 March 2023
Accepted: 14 March 2023
Published: 15 March 2023

1. Introduction

With the rapid emergence of various advanced technologies in different fields, such as computers and automobiles, the development of autonomous vehicles has become thriving in recent years. Autonomous vehicles have landed in various scenarios with the help and combination of those technologies. These vehicles are always equipped with different electronic control systems, which are driven with feedback regulation [1,2]. Therefore, the accurate estimation of vehicle real-time states and parameters can significantly improve the performance of vehicle control systems. As one of them, vehicle mass has a significant impact on the performance of systems that rely on vehicle dynamics, such as electronic stability program (ESP), anti-lock brake system (ABS), etc. [3]. Moreover, vehicle mass holds significant importance in vehicle dynamic modeling, and the estimation of other vehicle states, such as lateral velocity and sideslip angle, heavily depends on such dynamic model [4,5]. However, it is faced with a challenging problem since vehicle mass, especially for coaches and trucks, changes greatly depending on real-time payloads. Therefore, developing a method for determining the vehicle mass is critical to the safety and performance of autonomous vehicles, and has been extensively investigated by an increasing number of researchers in recent years so as to tackle the challenge.

Generally speaking, there are three types of parameter estimation methods for vehicle mass: sensor-based, model-based, and data-driven [6–12]. Sensor-based can be further divided into two kinds: fixed sensors and on-board sensors. Fixed sensors like weighbridges are the main weighing equipment used by factories, mines and businesses for bulk cargo measurement because of their benefits of sturdy construction and convenient use.

However, its application has been constrained by a lack of deactivation activity. On-board sensor research naturally evolved to address this issue [6]. With the installation of on-board sensors, mass weighing may be conducted while driving at any moment. However, the performance of these sensors is subject to the complex and changeable working environment, and the installation of additional sensors increases the cost of vehicle manufacturing.

The model-based approach primarily comprises three essential components, namely the sensor data, vehicle model, and the core algorithm, which are of lower cost compared to sensor-based ones [13–16]. In terms of sensor data, Rajamani and Hedrick adopted an additional linear variable differential transformer (LVDT) for observational purposes when constructing an adaptive vehicle states observer [17]. Jensen et al., in contrast, designed an approach that utilized solely parameters obtained from the on-board diagnostics (OBD) system to estimate mass [18]. Depending on the specific perspective chosen, various types of models can be established, including the lateral dynamic model, suspension dynamic model, longitudinal dynamic model, and others [17–20]. Given its dominant role throughout the driving process and exclusive use of pre-existing sensors, the longitudinal dynamic model facilitates the utilization of a larger amount of collected data for calculations and is frequently utilized by researchers.

With regard to the core algorithms, the majority of model-based approaches utilize filtering methods. Cai et al. presented a two-stage structured algorithm based on an extended Kalman filter (EKF) for estimating load and slope [21]. However, EKF suffers from inaccurate approximations for highly nonlinear models due to its reliance on first-order Taylor series expansions. In contrast, the unscented Kalman filter (UKF) offers a more precise approximation of nonlinear models using the unscented transform, even when they are highly nonlinear. Therefore, Jin et al. employed UKF in their dual-structure approach for online estimation of inertial parameters, including vehicle mass [22]. Nonetheless, it should be noted that Kalman filter-based suboptimal approximations are limited by their reliance on Gauss approximation. In an effort to overcome this limitation, Sun et al. utilized a particle filter (PF) to estimate the electric vehicle mass, which is a non-linear, non-Gauss filter and entirely unconstrained by the limitations of the Kalman filtering framework [23]. However, PF may suffer from particle degeneracy and sampling error. Hence, the pursuit of a universal and precise method that remains impervious to the vagaries of nonlinearity is of great importance for the accurate estimation of vehicle mass.

The data-driven approach presents an appealing alternative for vehicle state estimation, as it does not require costly additional hardware facilities and can learn nonlinear input-output relationships [24–26]. In a recent study, Korayem et al. compared a model-based method with a data-driven approach based on a multi-layer perceptron (MLP) for estimating the mass of a trailer tractor and found that the latter produced superior results [27]. However, these methods rely heavily on training data and can lead to unpredictable results if the distribution of test data does not match that of the training data, thus causing a reduction in accuracy and stability, and possibly leading to harm in control systems. To address this issue, Sieberg et al. suggested a hybrid estimating method using long short-term memory (LSTM) and UKF with a confidence level to enhance reliability in estimating other vehicle states [28].

Nevertheless, LSTM may suffer from vanishing gradients in long sequences, leading to diminished performance. As an alternative to LSTM, Transformer has gained increasing popularity in fields like computer vision (CV) and natural language processing (NLP), with its accuracy in time series regression having been proved [29]. While UKF is a popular choice of filter for state estimation, it requires more tuning parameters and may suffer from numerical stability issues. Moreover, the confidence level calculation method proposed in [28] can be time-consuming. To address these challenges and improve the accuracy and stability of autonomous vehicle mass estimation, we propose the transformer aided adaptive extended Kalman filter (TA-AEKF) method, which combines transformer and EKF with a fast confidence level calculation method. Our paper makes the following contributions:

1. A novel estimator based on transformer is introduced to provide an accurate pre-estimation of vehicle mass by learning nonlinear dynamics from vehicle data under diverse conditions, thus acting as a virtual observation for EKF.
2. A fast confidence level calculation method is designed for weight adjustment based on the conformity level between the real-time input and the training data, to adaptively determine the impact of transformer pre-estimation.
3. TA-AEKF is proposed to integrate the transformer pre-estimation into EKF with the adaptive weight adjustment in order to precisely and stably estimate vehicle mass.

The rest of this paper is organized as follows: In Section 2, the proposed method is explained in detail. In Section 3, simulation data are collected. In Section 4, experiments are conducted for method validation. Conclusions are made in Section 5.

2. Transformer Aided Adaptive Extended Kalman Filter

2.1. Framework

The proposed method comprises primarily three components. First, the acquired dataset is used to train the transformer neural network offline to dig deep information. When a new sequence of input variables is fed into the neural network for online estimation, a pre-estimated mass value will be produced. Second, the weight τ is determined by inputting the feature vector into the weight adjustment module at the same time. Finally, the pre-estimated mass value will be embedded into the EKF as the expanded observation quantity, and the weight τ will be utilized to balance the weight between the neural network and EKF. The algorithm structure of TA-AEKF is shown in Figure 1.

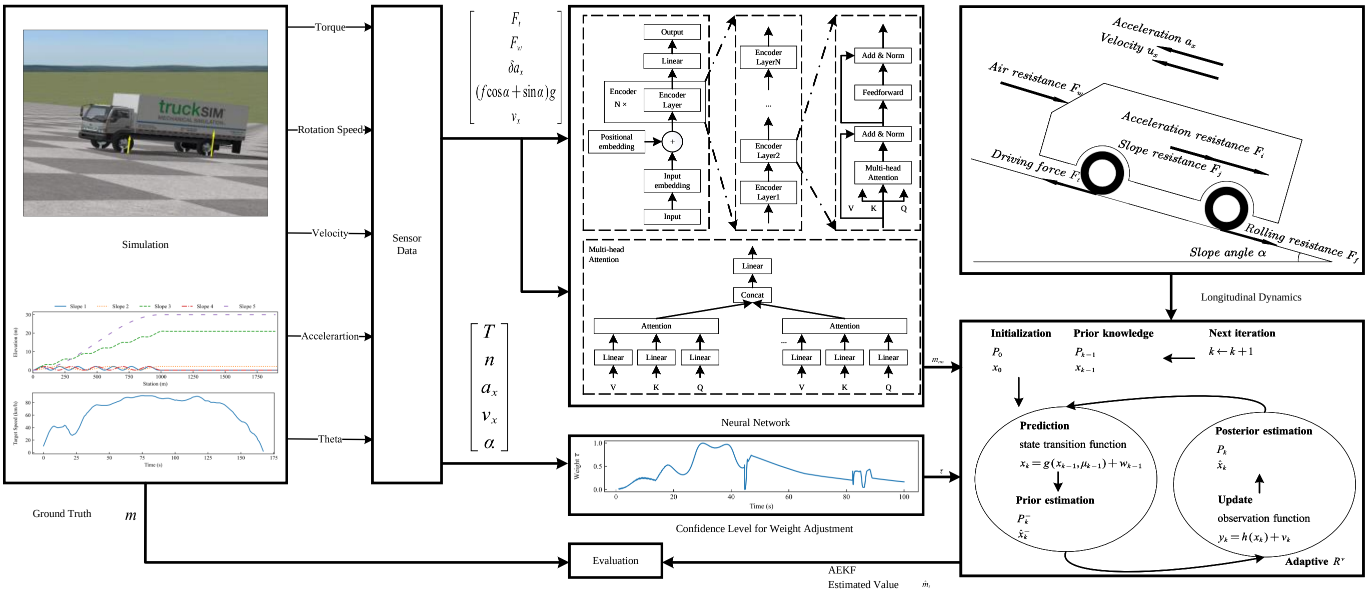


Figure 1. The framework of the proposed TA-AEKF estimator.

2.2. Transformer-Based Vehicle Mass Pre-Estimation

2.2.1. Preliminaries

A neural network is used for vehicle mass estimation and adaptive EKF (AEKF) design in this paper. Different kinds of neural networks, such as MLP, LSTM have been used for parameter estimation or even controller design in various fields for their excellence in approximating nonlinear functions [30–33]. Compared with MLP, LSTM is more frequently employed in time series regression problems to extract more hidden information. However, LSTM can suffer from vanishing gradients in long sequence data. Shown in Figure 2, a transformer neural network, where no decoder is adopted to solve this problem, for time series regression is proposed, which is different from traditional ones originally adopted in NLP with the encoder-decoder structure [34].

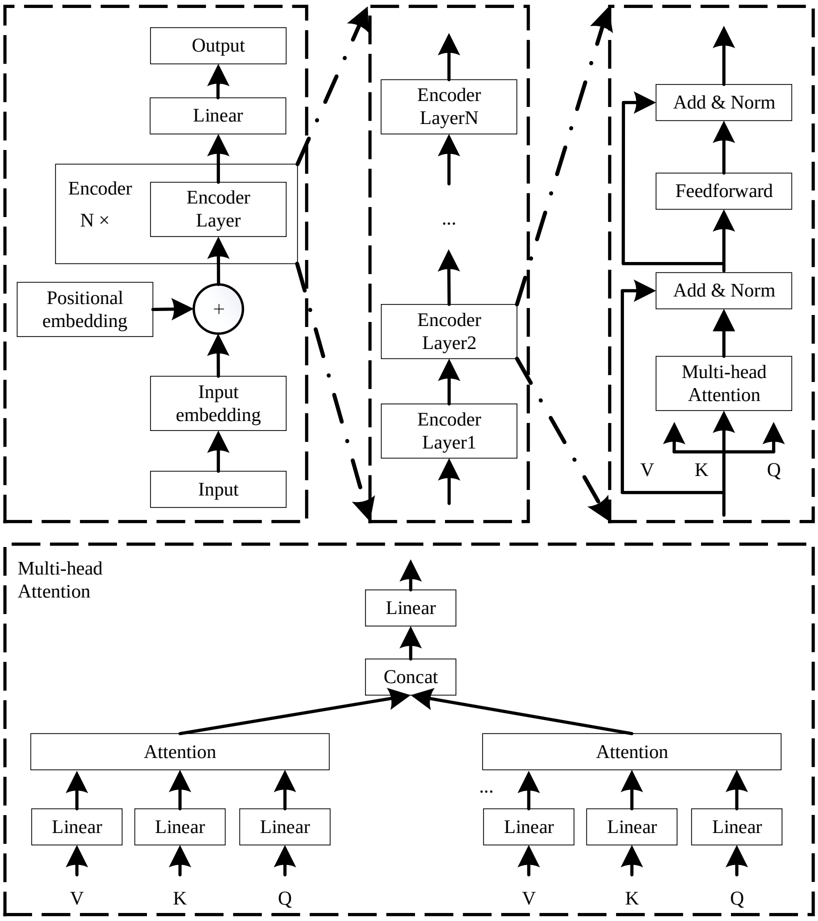


Figure 2. The flow chart of the proposed transformer model.

Firstly, the input feature will be projected from w-dimension features into n-dimension. z_t, the sum of the embeddings of the input signal and positional embeddings of the input signal, will then be passed into the encoder block. Here adopt the learnable positional embeddings in order to mine the sequential information in the time series. The encoder block is composed of several encoder layers. Each encoder layer mainly includes the fully connected layer and the self-attention layer, the core component of the transformer. The output Z of attention is calculated by:

$$Z = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

where Q, K, and V are the queries, keys, and values matrix, respectively, and d_k the dimension of the tensor, is used to scale the attention output.

In multi-head attention, the output of all attention heads will be concatenated and then multiplied by the weight matrix W to produce the output of the encoder layer Z_C:

$$Z_C = \text{ Concatenate } (Z_1, \ldots, Z_h)W \tag{2}$$

where h is the number of attention heads.

Features extracted by the encoder $E = [e_1; ...; e_t; ...; e_w] \in \mathbb{R}^{n \times r}$ are input to a fully connected layer in the linear block.

$$E^{'} = flatten(E) \in \mathbb{R}^{n \cdot r} \tag{3}$$

$$\hat{y} = W_o E^{'} + b_o \tag{4}$$

where $W_o \in \mathbb{R}^{l \times (n \cdot r)}$ and $b_o \in \mathbb{R}^l$ are used to project the $(n \cdot w)$-dimension features into l-dimension, $\hat{y} \in \mathbb{R}^l$ is the estimated regression output of neural network.

For the regression problem, the quadratic loss function will be adopted:

$$L = \|\hat{y} - y\|^2 \tag{5}$$

where y is the ground truth of the output variable, in this case, the total mass of the vehicle (m_{total}) and $l = 1$.

2.2.2. Input Feature Selection

Input features should be first decided properly to better study the nonlinear relationships between vehicle states. During the entire driving process, vehicle longitudinal motions is more often than lateral one. Then the vehicle longitudinal dynamic model can be chosen to select neural network features and design the adaptive EKF subsequently.

With the help of the transmission system, automobile driving torque T_{tq} can finally go to the driving wheels. The corresponding reaction force of the ground will propel the vehicle move forward. The various forces on the vehicle during the uphill process are as shown in Figure 3. Considering only the longitudinal dynamics, the model of the vehicle can be developed as follows:

$$F_t = F_f + F_w + F_j + F_i \tag{6}$$

where F_t, F_f, F_w, F_j, and F_i refer to the driving force, rolling resistance, air resistance, acceleration resistance and slope resistance, respectively.

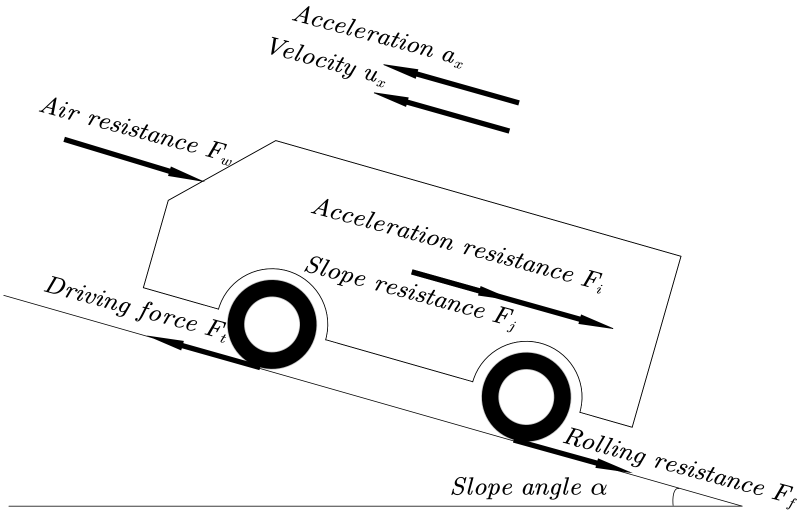


Figure 3. The force analysis of a vehicle moving on a slope.

The driving force F_t can be expressed as:

$$F_t = \frac{T_{tq} i_g i_0 \eta_T}{r} \tag{7}$$

where i_g is the gear ratio of the transmission, i_0 is the gear ratio of the final drive, η_T is the transmission mechanical efficiency, r is the effective radius of rotating tire, respectively.

The acceleration resistance F_j is defined as:

$$F_j = \delta m a_x \tag{8}$$

where δ is the rotating mass conversion factor, m is the total vehicle mass, a_x is the vehicle longitudinal acceleration, respectively.

The air resistance F_w when there is no wind is as follows:

$$F_w = \frac{1}{2}\rho C_d A_f u_x^2 \tag{9}$$

where ρ is the air density, C_d is the aerodynamic drag coefficient, A_f is the frontal area of the vehicle, u_x is the vehicle longitudinal speed, respectively.

The rolling resistance F_f is defined as:

$$F_f = mgf\cos\alpha \tag{10}$$

where g is the acceleration of gravity, f is the coefficient of rolling resistance, and α is the angle of slope, respectively.

The slope resistance F_i is defined as:

$$F_i = mg\sin\alpha \tag{11}$$

Integrate formulas of each force into (6) yields:

$$\frac{T_{tq} i_g i_0 \eta_T}{r} = \delta m a_x + \frac{1}{2}\rho C_d A_f u_x^2 + mgf\cos\alpha + mg\sin\alpha \tag{12}$$

Given that the transmission ratio is not always available in advance and that the effective tire radius varies and is challenging to determine, the transmission to wheel ratio is approximated here using the vehicle's speed and engine speed:

$$i_r = \frac{i_g i_0}{r} = \frac{\pi}{30}\frac{n}{u_x} \tag{13}$$

where i_r is defined as the ratio of $i_g i_0$ to r.

Therefore, (12) is transformed into:

$$\delta m a_x = \frac{\pi}{30}\frac{T_{tq}\eta_T n}{u_x} - \frac{1}{2}\rho C_d A_f u_x^2 - mg(f\cos\alpha + \sin\alpha) \tag{14}$$

As seen in (14), at the very least, driving force, longitudinal acceleration, longitudinal velocity, and slope angle should be gathered in order to determine vehicle mass. They can be collected using sensors, such as IMU and GNSS, which are frequently used in autonomous vehicles.

The neural network input feature vector is expressed as follows:

$$A_{input} = \begin{bmatrix} F_t & F_w & \delta a_x & (f\cos\alpha + \sin\alpha)g & v_x \end{bmatrix}^T \tag{15}$$

where v_x is velocity expressed in km/h.

The transmission of information in a classic feedforward neural network is unidirectional, with the current input being transmitted forward layer by layer through the neural network, corresponding to the current output. In real tasks, however, the output of the current time is influenced not just by the current time input, but also by the historical time. As a result, the back step of A_{input} is taken into account and supplied into the neural network.

2.2.3. Neural Network Architecture

For multi-head attention, 16 parallel attention heads are adopted. The amount of dropout applied to all linear layers is 0.3. The encoder block applied consists of 3 encoder layers.

2.3. Confidence Level for Weight Adjustment

The weight between EKF and the transformer neural network estimation results is represented by the weight factor, which impacts their weight in the final estimation results.

The confidence level is calculated by comparing the training and test datasets for similarity. A subspace partition of training data and mapping of test data is presented in [28]. However, it necessitates a significant amount of computing. A new approach is being developed here. Assume the variable $x^{'} = \begin{bmatrix} F_t & a_x & v_x & \alpha \end{bmatrix}^T$ in the training dataset's state quantity vector x follows the multivariate normal distribution:

$$p(x^{'};\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(x^{'}-\mu)^T\Sigma^{-1}(x^{'}-\mu)\right) \tag{16}$$

where μ is the mean vector, Σ is the covariance vector, n represents the number of features of x, and $|\Sigma| \equiv \det\Sigma$ is the determinant of Σ. Therefore, the probability density at μ of the training data is:

$$p_{max} = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \tag{17}$$

During the application of estimation, every time a new datapoint is fed into the neural network, substitute its value into (16), and get the corresponding probability density:

$$p_k = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(x_k^{'}-\mu)^T\Sigma^{-1}(x_k^{'}-\mu)\right) \tag{18}$$

Here, the probability density of the test dataset point will be divided by the maximum probability density in the training dataset, i.e., p_{max} to get the final confidence level:

$$\tau_k = \frac{p_k}{p_{max}} = \exp\left(-\frac{1}{2}(x_k^{'}-\mu)^T\Sigma^{-1}(x_k^{'}-\mu)\right) \tag{19}$$

2.4. AEKF

In comparison to RLS, KF is a correction-based recursive algorithm. It realizes the parameter estimation of a linear system with the minimum variance criterion. KF and its variants are frequently adopted for state estimation in linear systems [35,36]. For systems with non-linear transition equations or measurement equations, EKF is derived to linearize the equations. Different Kalman filter extensions (e.g., UKF, cubature KF (CKF)) have been adopted in various fields [37–39]. However, EKF performs better in terms of real-time performance, computing efficiency and parameter tuning in engineering applications. It is assumed that the nonlinear system has the following nonlinear state transition equation and observation equation:

$$x_k = g(x_{k-1},\mu_{k-1}) + w_{k-1} \tag{20}$$

$$y_k = h(x_k) + v_k \tag{21}$$

where x_k is the state quantity at step k, y_k is the observation quantity, μ is the control quantity, g is the transition function, h is the observation function, w is process noise, subject to $N(0,Q^w)$, v is observation noise, subject to $N(0,R^v)$.

The procedure of EKF is a series of successive operations of prediction based on state input and update based on measurement output.

First, the prior state estimation of state quantity $\hat{x}_k^-$ is calculated by:

$$\hat{x}_k^- = g(\hat{x}_{k-1},\mu_{k-1}) \tag{22}$$

and then the prior state estimation of covariance matrix P_k^- is calculated by:

$$P_k^- = AP_{k-1}A^T + Q^w \tag{23}$$

The Kalman gain K_k is calculated by:

$$K_k = P_k^- H^T (H P_k^- H^T + R^v)^{-1} \tag{24}$$

Therefore, the posterior state estimation of state quantity $\hat{x}_k$ is corrected with measurements by:

$$\hat{x}_k = \hat{x}_k^- + K_k \big(y_k - H \hat{x}_k^- \big) \tag{25}$$

and the posterior state estimation of covariance matrix P_k is updated by:

$$P_k = (I - K_k H) P_k^- \tag{26}$$

Here, A and H are the Jacobian matrices, calculated by the partial derivative of state transition function g and observation function h to state quantity x, respectively.

$$A = \frac{\partial g}{\partial x} \tag{27}$$

$$H = \frac{\partial h}{\partial x} \tag{28}$$

In order to apply EKF, the state space model is established. The longitudinal speed and vehicle mass are selected as the state quantity:

$$x_k = \begin{bmatrix} v_x & \frac{1}{m} \end{bmatrix}^T \tag{29}$$

The state equation form is derived to be:

$$\begin{bmatrix} v_{xk} \\ \frac{1}{m_k} \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} + w_{k-1} = \begin{bmatrix} v_{k-1} + a_x \Delta T \\ \frac{1}{m_{k-1}} \end{bmatrix} + w_{k-1} \tag{30}$$

The observation quantity is chosen as follows:

$$y_k = \begin{bmatrix} v_{x_k} \end{bmatrix} \tag{31}$$

The Jacobian matrix is:

$$A = \begin{bmatrix} \frac{\partial g_1}{\partial v} & \frac{\partial g_1}{\partial \frac{1}{m}} \\ \frac{\partial g_2}{\partial v} & \frac{\partial g_2}{\partial \frac{1}{m}} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ 0 & 1 \end{bmatrix} \tag{32}$$

where

$$A_{11} = 1 - \frac{3.6 \Delta T}{\delta m} \left(\frac{3\pi}{25} \frac{T_{tq} \eta_T n}{v^2} + \frac{1}{12.96} \rho C_D A v_x \right) \tag{33}$$

$$A_{12} = \frac{3.6 \Delta T}{\delta} \left(\frac{3\pi}{25} \frac{T_{tq} \eta_T n}{v} - \frac{1}{25.92} \rho C_D A v_x^2 \right) \tag{34}$$

The new observation quantity is chosen by expanding the original observation y_k with the transformer pre-estimation y_{nn_k} as follows:

$$\tilde{y}_k = \begin{bmatrix} y_k & y_{nn_k} \end{bmatrix} \tag{35}$$

$$y_{nn_k} = \frac{1}{m_{nnk}} \tag{36}$$

where m_{nnk} will be the estimated mass value of neural network.

Based on the weight adjustment module, an adaptive law is utilized to integrate the pre-estimation of the transformer into the AEKF. The flow chart of AEKF is shown in Figure 4. The pseudocode of TA-AEKF is shown in Algorithm 1, where $R^v_{nn} = \sigma^2_{m_{nn}}$

represents the observation covariance matrix increased by the neural network. Instead of using τ_k and $(1-\tau_k)$ as the weight factor, here divide them by τ_k and use $(\tau_k)/\tau_k = 1$ and $(1-\tau_k)/\tau_k$ to adjust the observation covariance matrix of neural network.

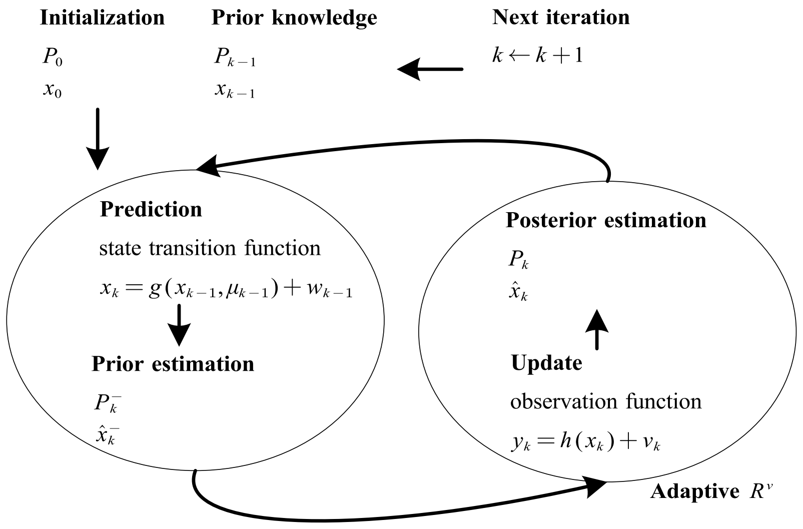


Figure 4. The architecture of AEKF.

Algorithm 1 Transformer Aided Adaptive Extended Kalman Filter

```
function DEEP ADAPTIVE EKF
    $\hat{x}_0 \leftarrow$ Initialization
    $P_0 \leftarrow$ Initialization
    for $k \leftarrow 1$ to $N$ do
        $\hat{x}_k^- \leftarrow g(\hat{x}_{k-1}, \mu_{k-1})$
        $\tilde{Q}_k^w \leftarrow Q_k^w$
        $P_k^- \leftarrow AP_{k-1}A^T + \tilde{Q}_k^w$
        $\tilde{R}_k^v \leftarrow \begin{bmatrix} R^v & 0 \\ 0 & (1-\tau_k)/\tau_k R_{nn}^v \end{bmatrix}$
        $K_k \leftarrow P_k^- H^T (HP_k^- H^T + \tilde{R}_k^v)^{-1}$
        $\tilde{y}_k \leftarrow \begin{bmatrix} y_k & y_{nn_k} \end{bmatrix}^T$
        $\hat{x}_k \leftarrow \hat{x}_k^- + K_k(\tilde{y}_k - H\hat{x}_k^-)$
        $P_k \leftarrow (I - K_k H)P_k^-$
    end for
    return $\hat{x}_N, P_N$
end function
```

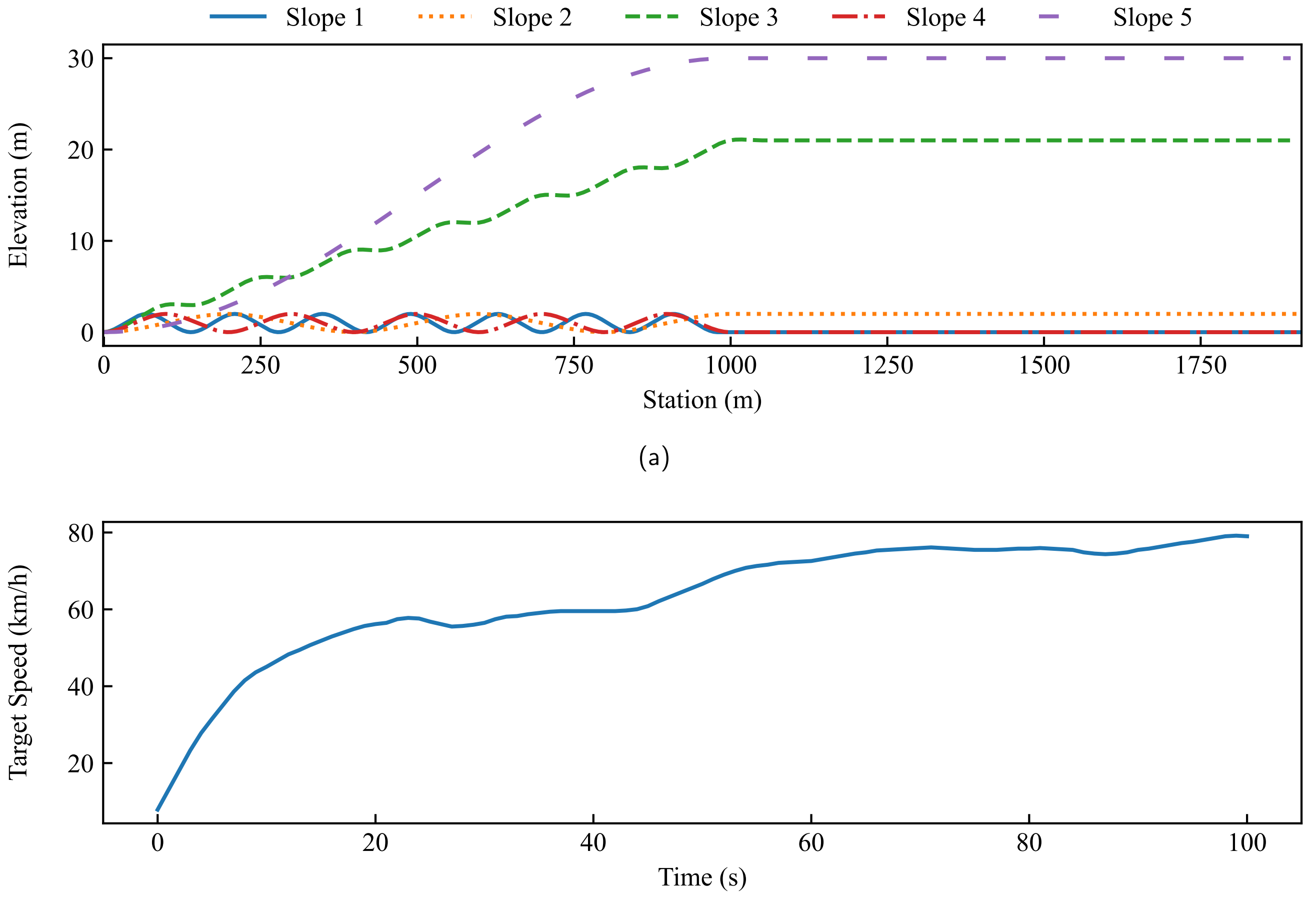
As $\tau \in (0,1]$, therefore $(1-\tau_k)/\tau_k \in [0,+\infty)$. In case when the new input data of the neural network is far from the original dataset, τ approaches zero. $(1-\tau_k)/\tau_k$ approaches $+\infty$. The posterior state $\hat{x}_k$ relies more on the matching degree between the physical model and the actual driving condition. As the similarity grows, the value τ increases gradually. If τ equals one, the observation covariance part based on the neural network equals zero. The final estimation result is completely based on a neural network.

3. Data Acquisition

For a data-driven approach such as the transformer, the amount and richness of the data may directly affect how the network functions. The dataset must include an adequate amount of data samples and should resemble actual ones. The neural network is tested using simulation data produced by the high-fidelity TruckSim model.

By configuring the settings for the truck model and the testing environment in TruckSim, gather sensor data in the virtual environment, and assess the accuracy of the parameter estimate technique. The main parameters of simulation vehilc e is shown in Table 1. The simulated vehicle's curb weight is 6042 kg. A range of payloads is selected, ranging from 4000 to 40,000 kg every 4000 kg.

Different slope conditions are used to generate enough simulation data. Five kinds of slope conditions are shown in Figure 5a, and the highway fuel economy driving schedule (HWFET) speed profile condition shown in Figure 5b are adopted. Each payload value and slope condition is combined to collect enough information. Each simulation is 100 s long.



(a)

(b)

Figure 5. Simulation conditions: (**a**) Slopes profiles. (**b**) Speed profiles.

Table 1. Main parameters of simulation vehicle.

Parameter	Value
vehicle curb weight	6042 kg
vehicle payload	4000 to 40,000 per 4000 kg
Frontal area	6.8 m^2
efficiency	0.99
sample time	0.01 s

In order to verify the accuracy and stability of the proposed method, two simulation tests are conducted. The gross vehicle weight is set to be 28,542 kg. The other parameters remain the same as the vehicle used for simulation data collection. Two kinds of slope conditions are used for test data generation, shown in Figure 6.

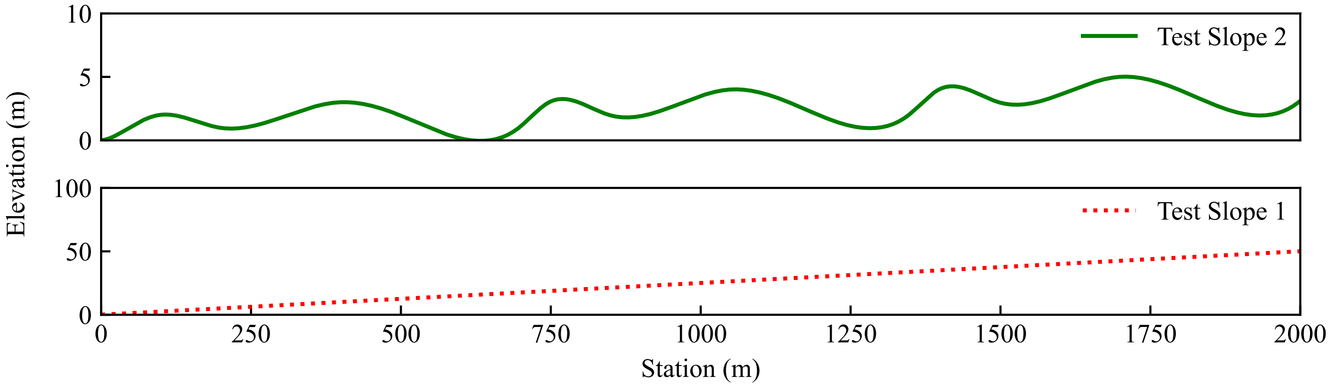


Figure 6. Slopes of simulation test conditions.

4. Results

The proportion of the training set, test set, and validation set are 6:2:2. The training of the neural network is implemented in Pytorch and Tsai [40]. The experiments were conducted on Ubuntu 20.04 operating system computer, which is equipped with Intel(R) Core(TM) i5-10600 CPU @ 3.30 GHz CPU, 64 G RAM and 12GB GeForce RTX 3080.

4.1. Evaluation Metrics

The root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) are used as evaluation metrics to assess the accuracy of mass estimate results:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{m}_i - m)^2} \tag{37}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{m}_i - m| \tag{38}$$

$$\text{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\hat{m}_i - m}{m}\right| \tag{39}$$

where $\hat{m}_i$ is the estimated value of mass, m is the ground truth of mass, n is the total number of estimated mass values.

In order to evaluate the stability of estimation results, here the convergence rate (CR) is adopted to evaluate the stability of estimation [41]. The estimation result is deemed to be convergent when the following requirements are satisfied:

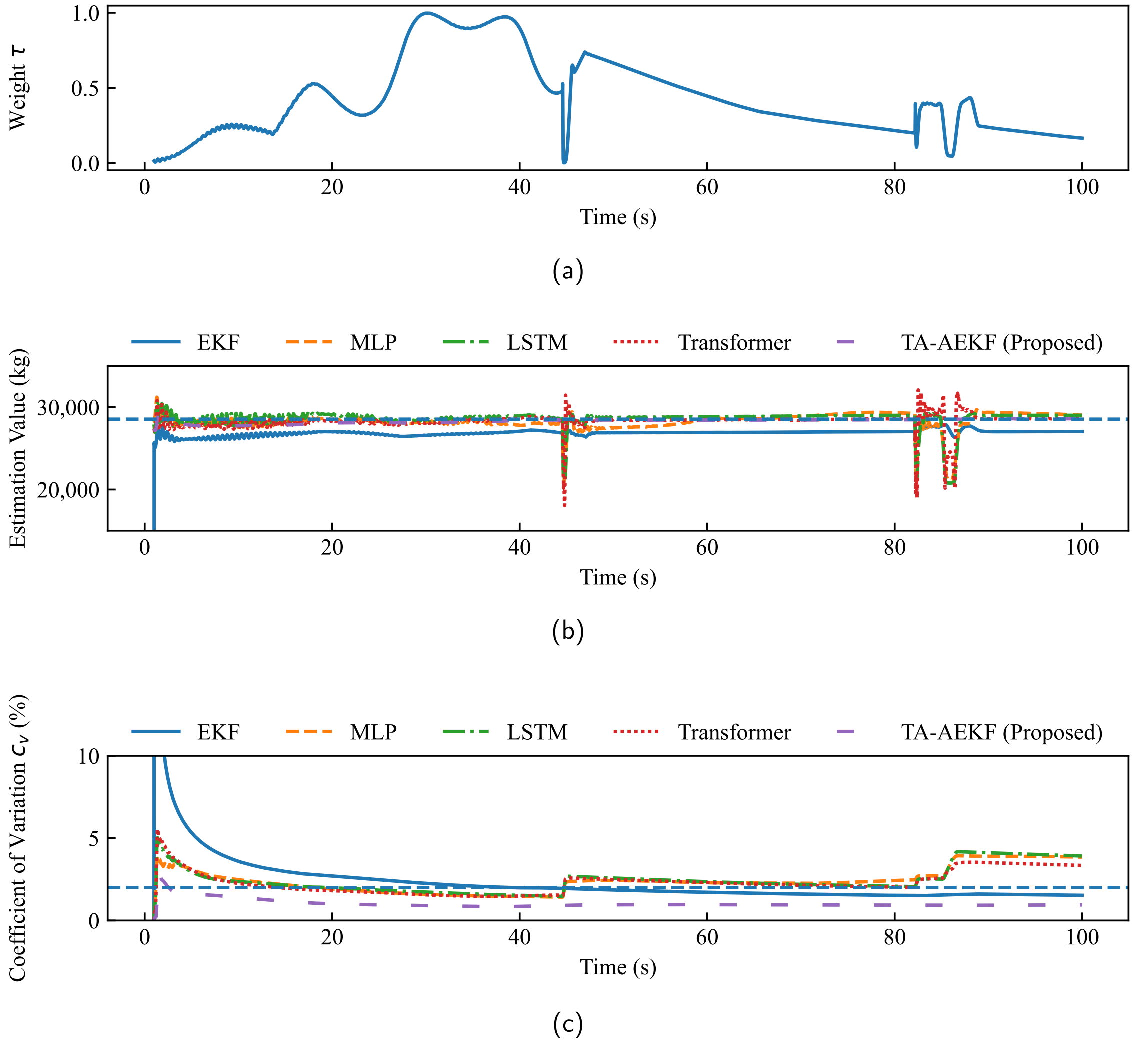
$$c_v(\hat{m}) = \frac{\sigma(\hat{m})}{\mu(\hat{m})} < c_{v0} \tag{40}$$

$$\frac{|\hat{m}_i - m|}{m} < 2\% \tag{41}$$

where c_v is the coefficient of variation, defined as the standard variance of estimation $\sigma(\hat{m})$ divided by the mean of estimation $\mu(\hat{m})$, and c_{v0} is the threshold of c_v and set to be 0.02. CR is defined as the calculation steps cost when c_v converges to lower than c_{v0}.

4.2. Performance Evaluation

The estimation results of each test are shown in Figures 7 and 8. The blue dashed line represents the result of ground truth in the middle subgraph and $c_{v0} = 2\%$ in the bottom subgraph, respectively.



(a)

(b)

(c)

Figure 7. Simulation result of test 1: (**a**) Weight calculation. (**b**) Estimation value of mass. The blue dashed line represents the result of the ground truth. (**c**) Evaluation with c_v. The blue dashed line represents $c_{v0} = 2\%$.

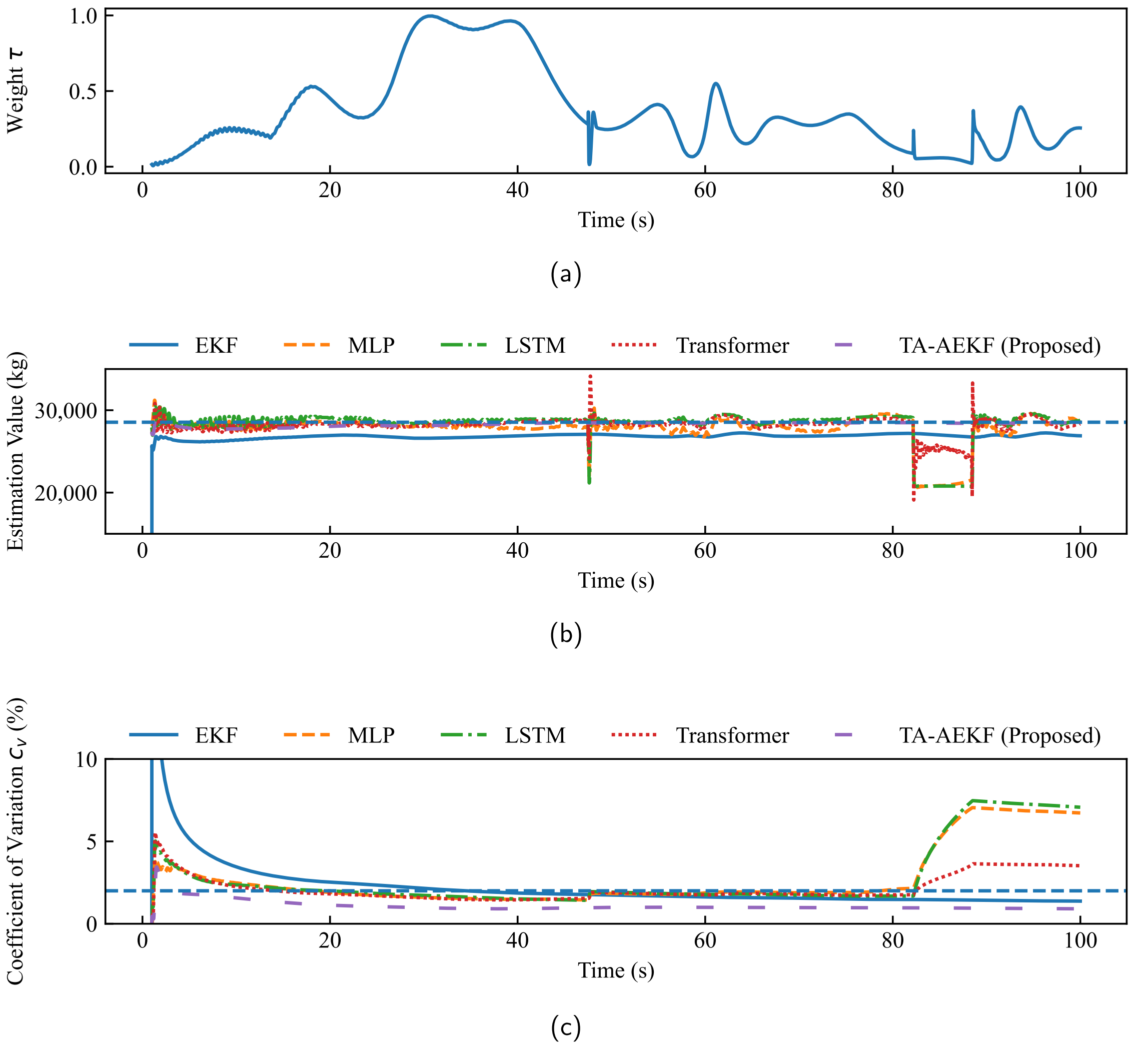


Figure 8. Simulation result of test 2: (**a**) Weight calculation. (**b**) Estimation value of mass. The blue dashed line represents the result of ground truth. (**c**) Evaluation with c_v. The blue dashed line represents $c_{v0} = 2\%$.

In Figure 7a, the weight level of test 1 changes due to the difference between the training set and the test set. Shown in Figure 7b, the estimation results of the neural network estimator vary and fluctuate occasionally. The transformer shows higher oscillations, followed by LSTM and MLP. However, these oscillations are not allowed in the control system as they would break the stability of the system. On the other hand, the estimation result of EKF is more stable. To start with, the EKF result undershoots to some high value but then begins to converge to ground truth gradually, showing more stability. The estimation results of the proposed method neither overshoot nor oscillates. Figure 7c shows the coefficient of variation and demonstrates the stability of each method. c_v of neural network methods gradually rises and is higher than 2% in the end, while the result of EKF overshoots and converges to lower than 2%. The result of the proposed method rises like neural network estimators, but will finally converge to a stable value lower than 2%.

In Figure 8a, the weight level of Test 2 changes more frequently, and the results of neural network estimators will also oscillate, which also shows the instability of neural network results, and might do harm to control systems. The changing trend of results also shows the advantages of the proposed method.

The metrics listed in Tables 2–4 demonstrate the superiority of the proposed method in terms of accuracy, with lower values in MAPE, RMSE, and MAE. RMSE and MAE are used to measure the deviation between the ground truth value and the predicted value, whereas MAPE measures the relative size of the deviation. RMSE and MAE have the same dimensions as the quantity, while MAPE is dimensionless. RMSE is more sensitive to outliers and can magnify prediction errors due to the square operation. In contrast, MAE and MAPE are more robust to extreme values.

Table 2. Evaluation metrics of simulation test 1 results.

Method	**MAPE** (%)	**MAE (kg)**	**RMSE (kg)**	**CR (steps)**
EKF	5.85	1669.77	1719.58	3790
MLP	2.28	652.05	1117.12	nan
LSTM	1.80	514.51	1124.30	nan
Transformer	1.42	404.91	972.35	nan
TA-AEKF (Proposed)	0.89	255.41	356.42	156

Table 3. Evaluation metrics of simulation test 2 results.

Method	**MAPE** (%)	**MAE (kg)**	**RMSE (kg)**	**CR (steps)**
EKF	5.98	1706.40	1745.78	3338
MLP	3.57	1018.14	2031.73	nan
LSTM	2.95	843.07	2018.19	nan
Transformer	2.15	613.11	1068.24	nan
TA-AEKF (Proposed)	0.90	257.53	357.61	212

Table 4. Evaluation metrics of simulation test 1 & 2 average results.

Method	**MAPE** (%)	**MAE (kg)**	**RMSE (kg)**	**CR (steps)**
EKF	5.91	1688.08	1732.68	3564
MLP	2.93	835.09	1574.42	nan
LSTM	2.38	678.79	1571.25	nan
Transformer	1.78	509.01	1020.29	nan
TA-AEKF (Proposed)	0.90	256.47	357.01	184

In test 1, the MAE of LSTM is smaller than that of MLP, but LSTM has a higher value of RMSE, indicating that there are higher extreme values in LSTM's estimation. As for test 2, EKF has the highest value of MAE, indicating the maximum number of estimation errors. However, LSTM and MLP have higher values of RMSE, indicating higher deviations in the results of neural network estimators. MAPE is easier to understand, even when the ground truth values are different, whereas MAE and RMSE need to combine the real value to judge the difference. These discrepancies are also apparent from Figures 7 and 8.

In terms of the time needed to converge to the threshold of efficiency variation, the proposed method is faster, demonstrating its stability advantage. Furthermore, among the three neural network estimators, the transformer neural network demonstrates superior performance. However, it should be noted that all neural network estimators exhibit instability, as they are unable to converge below the threshold of 2%. These further prove the necessity of integrating EKF with the transformer.

4.3. Input Feature Demonstration

To prove that the input feature is selected for the neural network, here the permutation method is used. In addition to the original features, lateral acceleration a_y and yaw rate

ω are also fed into the neural network. The MAPE increase when a feature is removed is calculated, shown in Figure 9. The removal of features included in A_t would result in higher MAPE than a_y and ω. Therefore, just selecting the input features in A_t is enough to recharacterize the nonlinear dynamics.

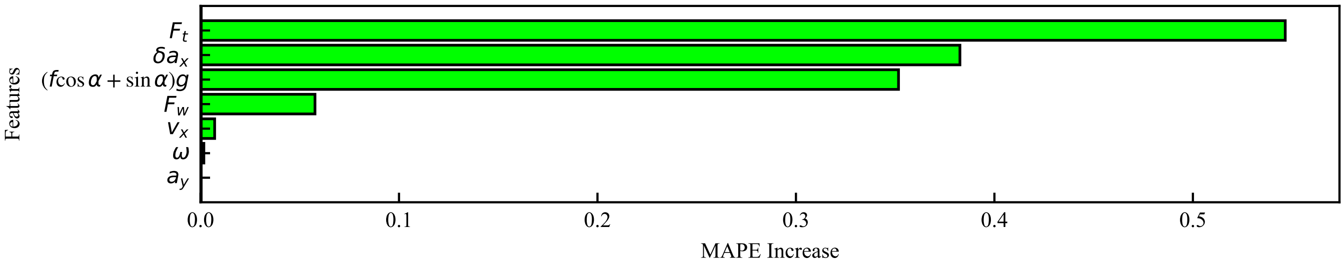


Figure 9. Permutation feature importance shown by MAPE increases when the corresponding feature is removed.

4.4. Confidence Level Demonstration

4.4.1. Calculation Speed Demonstration

To better demonstrate the calculation speed of the confidence level module, the computing time of each method is calculated as shown in Table 5. Results prove the outperformance of the proposed method in terms of calculation speed of confidence level.

Table 5. Evaluation computing time of weight adjustment.

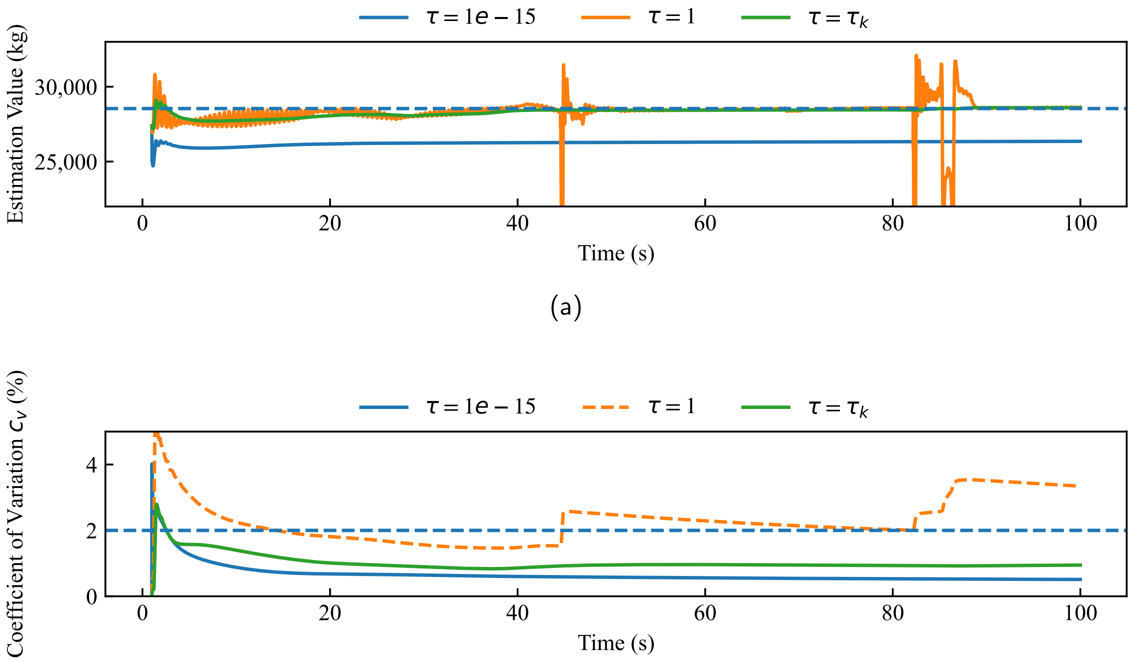
Method	**Computing Time (s)**		
	Test1	**Test2**	**Average of Test1 & Test2**
Method 1 ([28])	280.21	293.68	286.94
Method 2 (Proposed)	1.00	0.99	1.00

4.4.2. Improtance Demonstration

To better demonstrate the importance of the weight adjustment module, here different values are chosen to balance the relationship between the transformer and AEKF. The weight factor is set to be 10^{-15}, 1 and the adaptive weight τ_x, respectively. Furthermore, the results are shown in Figure 10 and Table 6. When τ equals one, $(1-\tau_k)/\tau_k$ equals zero. In this case, the result of TA-AEKF is the same as the estimation of the transformer. $\lim_{x\to 0}$ is represented by setting $\tau = 10^{-15}$. Furthermore, the result would be more reliant on the dynamic model. TA-AEKF estimator results when $\tau = \tau_k$ has the lowest MAPE, MAE, and RMSE. As for convergence steps, the values of $\tau = 10^{-15}$ and $\tau = \tau_k$ are very close. Therefore, the adaptive $\tau = \tau_x$ does work.

Table 6. Evaluation metrics of different weight factors.

Method	**MAPE (%)**	**MAE (kg)**	**RMSE (kg)**	**CR (steps)**
$\tau = 1e-15$	8.06	2301.15	2305.04	**154**
$\tau = 1$	1.42	404.91	972.35	nan
$\tau = \tau_k$	0.89	255.41	356.42	156



(a)

(b)

Figure 10. Simulation result of test 1 data using different values of weight to adjust: (**a**) Estimation value of mass. The blue dashed line represents the result of ground truth. (**b**) Evaluation with c_v. The blue dashed line represents $c_{v0} = 2\%$.

4.5. Real Time Test

To better evaluate the real-time performance of the proposed method, the corresponding computing time is calculated. For neural network estimators, the model can be trained offline and loaded before estimation. As shown in Table 7, LSTM costs the lowest time and TA-AEKF costs the highest. Besides, for a certain driving process, the total mass remains constant. Therefore, the calculation of mass can be done just once by collecting enough data. In this case, the computing time of the proposed method meets the need for the requested real-time performance.

Table 7. Evaluation computing time of simulation test.

Method	**Computing Time (s)**		
	Test1	**Test2**	**Average of Test1 & Test2**
EKF	0.53	0.57	0.55
MLP	0.17	0.21	0.19
LSTM	0.17	0.18	0.17
Transformer	0.44	0.46	0.45
TA-AEKF (Proposed)	1.07	1.09	1.08

5. Conclusions

In order to further improve the accuracy and robustness of autonomous vehicle mass estimation, the transformer-aided adaptive extended Kalman filter is proposed by integrating the transformer pre-estimation into an extended Kalman filter using a weight adjustment module. Simulation tests are conducted to validate the appropriateness of neural network feature selection, the adaptive law of the proposed weight adjustment module and the accuracy and stability of the proposed method.

In future work, we extend to collect more real vehicle data under more complex working conditions for performance verification. Furthermore, the proposed method will be extended to the estimation of other parameters in autonomous vehicles and different fields.

Author Contributions: Conceptualization, H.X. and Z.Y.; methodology, H.Z. and Z.Y.; investigation, Z.Y. and T.Z.; writing—original draft preparation, Z.Y.; writing—review and editing, H.Z., H.X., T.Z., Z.L. and W.W.; supervision, H.X.; funding acquisition, H.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by the Key-Area Research and Development Program of Guangdong Province (2020B0909050004) and Natural Science and Technology Special Projects under Grant 2019-1496.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

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

References

1. Singh, K.B.; Arat, M.A.; Taheri, S. Literature Review and Fundamental Approaches for Vehicle and Tire State Estimation. *Veh. Syst. Dyn.* **2019**, *57*, 1643–1665. [CrossRef]
2. Xiong, H.; Tan, Z.; Zhang, R.; He, S. A New Dual Axle Drive Optimization Control Strategy for Electric Vehicles Using Vehicle-to-Infrastructure Communications. *IEEE Trans. Ind. Inform.* **2020**, *16*, 2574–2582. [CrossRef]
3. Xiong, H.; Zhu, X.; Zhang, R. Energy Recovery Strategy Numerical Simulation for Dual Axle Drive Pure Electric Vehicle Based on Motor Loss Model and Big Data Calculation. *Complexity* **2018**, *2018*, e4071743. [CrossRef]
4. Liu, W.; Xiong, L.; Xia, X.; Lu, Y.; Gao, L.; Song, S. Vision-Aided Intelligent Vehicle Sideslip Angle Estimation Based on a Dynamic Model. *IET Intell. Transp. Syst.* **2020**, *14*, 1183–1189. [CrossRef]
5. Xia, X.; Hashemi, E.; Xiong, L.; Khajepour, A. Autonomous Vehicle Kinematics and Dynamics Synthesis for Sideslip Angle Estimation Based on Consensus Kalman Filter. *IEEE Trans. Control. Syst. Technol.* **2023**, *31*, 179–192. [CrossRef]
6. Yang, S.K.; Liu, T.S.; Cheng, Y.C. Automatic Measurement of Payload for Heavy Vehicles Using Strain Gages. *Measurement* **2008**, *41*, 491–502. [CrossRef]
7. AG, C. Continental In-Tire Sensors Read Tread Depth . 2014. Available online: https://www.continental.com/en/press/press-releases/2014-05-07-tpms-profile-104542 (accessed on 14 April 2021).
8. Zhang, Y.; Zhang, Y.; Ai, Z.; Feng, Y.; Zhang, J.; Murphey, Y.L. Estimation of Electric Mining Haul Trucks' Mass and Road Slope Using Dual Level Reinforcement Estimator. *IEEE Trans. Veh. Technol.* **2019**, *68*, 10627–10638. [CrossRef]
9. Torabi, S.; Wahde, M.; Hartono, P. Road Grade and Vehicle Mass Estimation for Heavy-duty Vehicles Using Feedforward Neural Networks. In Proceedings of the 2019 4th International Conference on Intelligent Transportation Engineering (ICITE), Singapore, 5–7 September 2019 ; IEEE: Singapore, 2019; pp. 316–321. [CrossRef]
10. Wang, H.; Zheng, Y.; Yu, Y. Joint Estimation of SOC of Lithium Battery Based on Dual Kalman Filter. *Processes* **2021**, *9*, 1412. [CrossRef]
11. Feng, W.; Dong, W.; Zhai, S.; Zhang, G.; Sun, X.; Ji, Y. A Deep Reinforcement Learning Method for Freight Train Driving Based on Domain Knowledge and Mass Estimation Network. In Proceedings of the 2021 6th International Conference on Machine Learning Technologies, Jeju Island, Republic of Korea, 23–25 April 2021; ACM: New York, NY, USA, 2021; pp. 41–46. [CrossRef]
12. Ritter, A.; Widmer, F.; Vetterli, B.; Onder, C.H. Optimization-Based Online Estimation of Vehicle Mass and Road Grade: Theoretical Analysis and Experimental Validation. *Mechatronics* **2021**, *80*, 102663. [CrossRef]
13. Li, X.; Ma, J.; Zhao, X.; Wang, L. Intelligent Two-Step Estimation Approach for Vehicle Mass and Road Grade. *IEEE Access* **2020**, *8*, 218853–218862. [CrossRef]
14. Zhang, F.; Wang, Y.; Hu, J.; Yin, G.; Chen, S.; Zhang, H.; Zhou, D. A Novel Comprehensive Scheme for Vehicle State Estimation Using Dual Extended H-Infinity Kalman Filter. *Electronics* **2021**, *10*, 1526. [CrossRef]
15. Mei, C.; Peng, J.; Yuhui, H. An Online Mass Estimation Algorithm for Single-shaft Parallel Hybrid Commercial Vehicle. *Acta Armamentarii* **2021**, *42*, 1838–1846. [CrossRef]
16. Zhang, Z.; Yin, G.; Wu, Z. Joint Estimation of Mass and Center of Gravity Position for Distributed Drive Electric Vehicles Using Dual Robust Embedded Cubature Kalman Filter. *Sensors* **2022**, *22*, 10018. [CrossRef]
17. Rajamani, R.; Hedrick, J.K. Adaptive Observers for Active Automotive Suspensions: Theory and Experiment. *IEEE Trans. Control. Syst. Technol.* **1995**, *3*, 86–93. [CrossRef]
18. Jensen, K.M.; Santos, I.F.; Clemmensen, L.K.H.; Theodorsen, S.; Corstens, H.J.P. Mass Estimation of Ground Vehicles Based on Longitudinal Dynamics Using IMU and Can-Bus Data. *Mech. Syst. Signal Process.* **2022**, *162*, 107982. [CrossRef]
19. Reina, G.; Paiano, M.; Blanco-Claraco, J.L. Vehicle Parameter Estimation Using a Model-Based Estimator. *Mech. Syst. Signal Process.* **2017**, *87*, 227–241. [CrossRef]

20. Jeong, D.; Kim, S.; Lee, J.; Choi, S.B.; Kim, M.; Lee, H. Estimation of Tire Load and Vehicle Parameters Using Intelligent Tires Combined With Vehicle Dynamics. *IEEE Trans. Instrum. Meas.* **2020**, *70*, 1–12. [CrossRef]
21. Cai, L.; Wang, H.; Jia, T.; Peng, P.; Pi, D.; Wang, E. Two-Layer Structure Algorithm for Estimation of Commercial Vehicle Mass. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2020**, *234*, 378–389. [CrossRef]
22. Jin, X.; Yang, J.; Li, Y.; Zhu, B.; Wang, J.; Yin, G. Online Estimation of Inertial Parameter for Lightweight Electric Vehicle Using Dual Unscented Kalman Filter Approach. *IET Intell. Transp. Syst.* **2020**, *14*, 412–422. [CrossRef]
23. Sun, S.; Zhang, N.; Walker, P.; Lin, C. Intelligent Estimation for Electric Vehicle Mass with Unknown Uncertainties Based on Particle Filter. *IET Intell. Transp. Syst.* **2020**, *14*, 463–467. [CrossRef]
24. Chen, J.; Feng, X.; Jiang, L.; Zhu, Q. State of Charge Estimation of Lithium-Ion Battery Using Denoising Autoencoder and Gated Recurrent Unit Recurrent Neural Network. *Energy* **2021**, *227*, 120451. [CrossRef]
25. Barati, K.; Shen, X.; Li, N.; Carmichael, D.G. Automatic Mass Estimation of Construction Vehicles by Modeling Operational and Engine Data. *J. Constr. Eng. Manag.* **2022**, *148*, 04021208. [CrossRef]
26. Xiong, H.; Wang, Z.; Wu, G.; Pan, Y.; Yang, Z.; Long, Z. Steering Actuator Fault Diagnosis for Autonomous Vehicle With an Adaptive Denoising Residual Network. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–13. [CrossRef]
27. Korayem, A.H.; Khajepour, A.; Fidan, B. Trailer Mass Estimation Using System Model-Based and Machine Learning Approaches. *IEEE Trans. Veh. Technol.* **2020**, *69*, 12536–12546. [CrossRef]
28. Sieberg, P.M.; Blume, S.; Reicherts, S.; Maas, N.; Schramm, D. Hybrid State Estimation-A Contribution Towards Reliability Enhancement of Artificial Neural Network Estimators. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 6337–6346. [CrossRef]
29. Hannan, M.A.; How, D.N.T.; Lipu, M.S.H.; Mansor, M.; Ker, P.J.; Dong, Z.Y.; Sahari, K.S.M.; Tiong, S.K.; Muttaqi, K.M.; Mahlia, T.M.I.; et al. Deep Learning Approach towards Accurate State of Charge Estimation for Lithium-Ion Batteries Using Self-Supervised Transformer Model. *Sci. Rep.* **2021**, *11*, 19541. [CrossRef]
30. Pang, H.; Liu, F.; Xu, Z. Variable Universe Fuzzy Control for Vehicle Semi-Active Suspension System with MR Damper Combining Fuzzy Neural Network and Particle Swarm Optimization. *Neurocomputing* **2018**, *306*, 130–140. [CrossRef]
31. Xing, Y.; Lv, C. Dynamic State Estimation for the Advanced Brake System of Electric Vehicles by Using Deep Recurrent Neural Networks. *IEEE Trans. Ind. Electron.* **2020**, *67*, 9536–9547. [CrossRef]
32. Lin, C.; Wang, H.; Fu, M.; Yuan, J.; Gu, J. A Gated Recurrent Unit-Based Particle Filter for Unmanned Underwater Vehicle State Estimation. *IEEE Trans. Instrum. Meas.* **2020**, *70*, 1–12. [CrossRef]
33. Wang, M.; Pang, H.; Luo, J.; Liu, M. On an Enhanced Back Propagation Neural Network Control of Vehicle Semi-Active Suspension with a Magnetorheological Damper. *Trans. Inst. Meas. Control.* **2023**, *45*, 512–523. [CrossRef]
34. Zerveas, G.; Jayaraman, S.; Patel, D.; Bhamidipaty, A.; Eickhoff, C. A Transformer-Based Framework for Multivariate Time Series Representation Learning. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Singapore, 14–18 August 2021; ACM: New York, NY, USA, 2021; pp. 2114–2124. [CrossRef]
35. Hong, S.; Lee, C.; Borrelli, F.; Hedrick, J.K. A Novel Approach for Vehicle Inertial Parameter Identification Using a Dual Kalman Filter. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 151–161. [CrossRef]
36. Xiong, L.; Xia, X.; Lu, Y.; Liu, W.; Gao, L.; Song, S.; Yu, Z. IMU-Based Automated Vehicle Body Sideslip Angle and Attitude Estimation Aided by GNSS Using Parallel Adaptive Kalman Filters. *IEEE Trans. Veh. Technol.* **2020**, *69*, 10668–10680. [CrossRef]
37. Xiong, H.; Liu, J.; Zhang, R.; Zhu, X.; Liu, H. An Accurate Vehicle and Road Condition Estimation Algorithm for Vehicle Networking Applications. *IEEE Access* **2019**, *7*, 17705–17715. [CrossRef]
38. Yang, F.; Zhang, S.; Li, W.; Miao, Q. State-of-Charge Estimation of Lithium-Ion Batteries Using LSTM and UKF. *Energy* **2020**, *201*, 117664. [CrossRef]
39. Liu, W.; Xia, X.; Xiong, L.; Lu, Y.; Gao, L.; Yu, Z. Automated Vehicle Sideslip Angle Estimation Considering Signal Measurement Characteristic. *IEEE Sens. J.* **2021**, *21*, 21675–21687. [CrossRef]
40. Oguiza, I. *tsai - A State-of-the-Art Deep Learning Library for Time Series and Sequential Data*; Github: San Francisco, CA, USA, 2022.
41. Kidambi, N.; Harne, R.L.; Fujii, Y.; Pietron, G.M.; Wang, K.W. Methods in Vehicle Mass and Road Grade Estimation. *SAE Int. J. Passeng. Cars-Mech. Syst.* **2014**, *7*, 981–991. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.