Research Article

Soft Sensor of Vehicle State Estimation Based on the Kernel Principal Component and Improved Neural Network

Haorui Liu,1 Juan Yang,2 Heli Yang,1 and Fengyan Yi3

1 School of Automotive Engineering, Dezhou University, Dezhou 253023, China
2 School of Economics and Management, Dezhou University, Dezhou 253023, China
3 Automotive Engineering College, Shandong Jiaotong University, Jinan 250023, China

Correspondence should be addressed to Fengyan Yi; yi_fengyan@163.com

Received 6 March 2016; Revised 5 August 2016; Accepted 30 August 2016

Academic Editor: Jesus Corres

Copyright © 2016 Haorui Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the car control systems, it is hard to measure some key vehicle states directly and accurately when running on the road and the cost of the measurement is high as well. To address these problems, a vehicle state estimation method based on the kernel principal component analysis and the improved Elman neural network is proposed. Combining with nonlinear vehicle model of three degrees of freedom (3 DOF), longitudinal, lateral, and yaw motion, this paper applies the method to the soft sensor of the vehicle states. The simulation results of the double lane change tested by Matlab/SIMULINK cosimulation prove the KPCA-IENN algorithm (kernel principal component algorithm and improved Elman neural network) to be quick and precise when tracking the vehicle states within the nonlinear area. This algorithm method can meet the software performance requirements of the vehicle states estimation in precision, tracking speed, noise suppression, and other aspects.

1. Introduction

Electronic control systems are increasingly being applied to modern vehicles to improve its safety, stability, and comfort. Electronic stability program (ESP), four wheel steering (4WS), and active steering control (ASC) improve the handling stability of a vehicle by controlling the body lateral and yawing motion. Active roll control (ARC) and active body control (ABC) improve the driving comfort by controlling the vertical movement and roll stability. The premise condition to realize the particular functions of these electronic control systems is to obtain accurate state parameters of vehicle when car is running on the road. Those parameters include vertical speed, lateral acceleration, yaw velocity, sideslip angle, and so on. Due to the measuring technology and its cost, it is quite a challenge to measure all the important state variables directly. At present, the soft sensor technology based on state estimation has been successfully applied to the estimation of vehicle dynamics and kinematics parameter information. Venhovens and Naab [1] and Gustafsson et al. [2] estimated the lateral acceleration, yaw velocity, tire force, and road tire friction coefficient by using Kalman filter (KF). Wenzel et al. [3, 4] estimate vehicle’s state variables and parametric variables by adopting dual extended Kalman filter parallel estimating method. Some other scholars use Luenberger observer and sliding mode observer to estimate sideslip angle and yawing force. Among all these estimation algorithms, the state estimation method based on KF is the most widely applied and has achieved relatively better results in field tests. However, the design difficulties of the KF algorithm are (1) the requirement of accurate knowledge for the model of the system observed; (2) the requirement of the statistical properties of the known system noise and the measurement noise.

Principal component analysis (PCA) is a kind of linear algorithm. It only considers the second-order statistics and can only extract the linear relation of the data. When non-linear relation exists among a large number of variables of a nonlinear system, the principal component analysis (PCA) will not meet the requirements any more. While the kernel principal component analysis (KPCA) is a nonlinear extension of the principal component analysis (PCA) [5–7].
2. Vehicle Model and Principles of KPCA-IENN

2.1. Nonlinear Dynamic Model of Vehicle. The 2-DOF model used to analyze the vehicle’s yaw motion and lateral motion is a linear model, which is on the assumption that the vehicle is driving in uniform velocity. But in reality, longitudinal velocity is subject to change at any time and this changing longitudinal velocity has a significant impact on the yaw and lateral motion and constitutes a nonlinear relation among the state variables. This paper established the vehicle state estimation soft sensor model based on the longitudinal, lateral, and yaw 3-DOF nonlinear vehicle model. Figure 1 is the 3-DOF vehicle dynamic model.

As is shown in Figure 1, $X_oO_xY_o$, is an absolute coordinate system fixed in the horizontal plane, and the $xoy$ is the vehicle coordinate fixed on the gravity center, where the $x$ axis overlaps with the vehicle’s longitudinal axis and is positive toward the direction of the vehicle movement; $y$ axis through the vehicle gravity center $o$ and is positive to the left. All the angles and torques within the plane of $X_oO_xY_o$ are positive counter-clockwise and all the components of vectors are positive with the axis direction. The state equations and measurement equations of vehicle nonlinear dynamic model of 3 DOF are shown as

$$
\dot{\omega} = \frac{\left(l_f^2C_f + l_r^2C_r\right)}{I_z} \dot{\omega} + \frac{\left(l_fC_f + l_rC_r\right)}{I_z} \beta + \frac{l_fC_f}{I_z} \delta_f,
$$

$$
\dot{\beta} = \frac{\left(l_fC_r - l_fC_f - mv_y^2\right)}{m} \dot{\omega} + \frac{C_f + C_r}{v_x} \beta + \frac{C_f \delta_f}{m v_x},
$$

$$
\dot{v}_x = \omega \beta v_x + a_x,
$$

$$
\dot{a}_x = \frac{\left(l_fC_r - l_fC_f\right)}{m} \omega + \frac{\left(l_f + l_r\right)}{m} \beta + \frac{l_f}{m} \delta_f.
$$

In the equation, $\omega$ is the yaw velocity; $\beta$ is the sideslip angle; $v_x$ is the longitudinal velocity; $\delta_f$ is the front wheel angle; $a_x$ is the lateral acceleration; $a_x$ is the longitudinal acceleration; $m$ is the vehicle mass; $I_z$ is the moment of inertia about the $z$-axis; $l_f$ is the front wheelbase; $l_r$ is the rear wheelbase; $C_f$ is the front wheel effective cornering stiffness; $C_r$ is the rear wheel cornering stiffness.

With the steering wheel angle as control input and the steering wheel to the front wheel angle transmission ratio $i = 16$, we get

$$
\delta_{sw} = i \delta_f.
$$

2.2. Soft Sensor Technology. Soft sensor technology is also named soft instrument technology [7]. The basic principle is to get the estimation value of the primary variables through the establishment of the mathematical model by using those easily detectable assistant variables that are closely related to those primary variables of difficult detection. The commonly used methods for soft sensor technology include the mechanism analysis modeling, the regression analysis, state estimation, pattern recognition, neural networks, fuzzy mathematics, process tomography, the correlation analysis, and so forth. The state estimation based soft sensor technique adopted in this thesis deducted the estimation of the primary variables using detectable information and the algorithm (the kernel principal component analysis and the improved Elman neural network). The technique realized the measurement for...
the state parameters that are otherwise hard to measure by conventional method. Figure 2 is the vehicle state estimation system based on soft sensor technique. In Figure 2, $w$, $v$, and $u$ represent measurable process noise, measurement noise, and control variable, respectively. $y(k)$ and $\hat{y}(k)$ are the real value and estimated value of the object outputs, respectively. $\tilde{x}(k)$ is the estimated value of the object state with the discrete moments $k = 0, 1, 2, \ldots$

2.3. The Reduction of the Feature Parameters Based on the Kernel Principal Component Analysis. The basic idea of the kernel principal component analysis is to implicitly map the data in input space to feature space through kernel function and then to process the kernel principal component analysis in the feature space. This is a powerful approach of extracting nonlinear features, realizing the reduction of the feature parameters, reducing of data redundancy, and getting the primary feature vectors with low classification error rate and high sensitivity with the driving conditions.

2.3.1. The Selection of Feature Parameters. There are many feature parameters that can be used to represent the vehicle status, such as average running time, the average acceleration time, the average deceleration time, the average time of driving at uniform velocity, the average idling time, the average running distance, the maximal driving speed, the maximum acceleration, the maximum vehicle deceleration, the average acceleration during the acceleration period, the average deceleration during the deceleration period, the standard deviation of the acceleration, and the changing rate of the throttle opening. Firstly, this thesis reduces the feature parameters representing vehicle status via correlation analysis. In other words, it uses sequences $x(n)$ and $y(n)$ to represent the feature parameters pair-wise respectively, in the equation, $n = 1, 2, \ldots, a, \ldots, b, \ldots$. Choose any two time series when $n = a \sim b$ and then compare the level of similarity of the two sequences according to

$$r = \frac{\sum_{i=a}^{b} x(n) y^*(n)}{\left( \sum_{n=a}^{b} |x(n)|^2 \sum_{n=a}^{b} |y(n)|^2 \right)^{1/2}}.$$  

(3)

In the equation, “$*$” is the conjugation of $y(n)$.

Sort the above results of operation from highest irrelevance to the lowest and then make correlation analysis on the first 15 parameters and the categories of working condition. Then rate the correlation from high to low, and finally we get the 10 feature parameters as the classification standard for working conditions; they are the maximal driving speed, average speed, the minimum deceleration, the average acceleration, the uniform velocity time ratio, idling time ratio, the average acceleration during the acceleration period, the average deceleration during the deceleration period, the average degree of the throttle opening, and the time when the throttle opening degree is zero.

2.3.2. The Reduction of the Feature Parameters Based on the Kernel Principal Component Analysis. Through nonlinear mapping $\phi$, the kernel principal component analysis transfers the original data $x_i \in \mathbb{R}^d (i = 1, 2, \ldots, M)$ from data space into the $\phi(x_i)$ in a new high dimensional feature space $F$ and then gets the nonlinear principal components which enables the original data to have better divisibility by using principal component analysis (PCA) on these after mapping data.

The covariance matrix for the after mapping data $\phi(x_i)$ is as follows:

$$C = \frac{1}{M} \sum_{i=1}^{M} \phi(x_i) \phi(x_i)^T.$$  

(4)

If we set the eigenvalue for the covariance matrix $C$ to be $\lambda$ and the relative feature vector to be $v$, then

$$\lambda v = Cv.$$  

(5)

For all the $i = 1, 2, \ldots, M$, do dot-product of each after mapping sample $\phi(x_i)$ with both sides of (5), respectively,

$$\lambda \langle \phi(x_i) \cdot v \rangle = \langle \phi(x_i) \cdot Cv \rangle.$$  

(6)

When $\lambda \neq 0$, there is coefficient $a_i$ ($i = 1, 2, \ldots, M$); use linear combination of $\phi(x_i)$ to represent feature vector $v$

$$v = \sum_{i=1}^{M} a_i \phi(x_i).$$  

(7)

Substitute (4) and (7) into the above inner product formula, and then get

$$\lambda \sum_{i=1}^{M} a_i \langle \phi(x_k) \cdot \phi(x_i) \rangle = \frac{1}{M} \sum_{i=1}^{M} a_i \left( \phi(x_k) \cdot \sum_{j=1}^{M} \phi(x_j) \phi(x_j)^T \phi(x_i) \right).$$  

(8)

To avoid complex direct calculations, in the feature space, we define the elements $K_{ij}$ of the symmetric matrix $K$ as kernel function

$$K_{ij} = \langle \phi(x_i) \cdot \phi(x_j) \rangle.$$  

(9)

In the equation, $j = 1, 2, \ldots, M$.

As $K$ is symmetric matrix, so (5) can be simplified as

$$M\lambda a = Ka.$$  

(10)
Solve (10); we will get the required eigenvalues \( \lambda_k \) and the feature vectors \( a_k \) \((k = 1,2,\ldots,M)\). Reduce the system inputted feature parameters via the method of keeping the first \( p \) feature vectors. The eigenvalues of the diagonalized matrix \( K \) are denoted as \( \lambda_{k1} \geq \lambda_{k2} \geq \cdots \geq \lambda_{kM} \), if the number of the smallest nonzero eigenvalues for \( K \) is set to be \( p \), and then with (9) and (10), we could get

\[
a_k^T a_k = \frac{1}{\lambda_k}.
\]

In the equation, \( k = 1,2,\ldots,p \).

The projection of the after mapping data on the feature vectors \( v^k \) \((k = 1,2,\ldots,p)\) is the nonlinear principal components of the original data \( x_i \) obtained through nonlinear mapping \( \varphi \)

\[
\rho_k = \langle v_k \cdot \varphi(x) \rangle = \sum_{i=1}^{M} a_k^i \langle \varphi(x_i) \cdot \varphi(x) \rangle
= \sum_{i=1}^{M} a_k^i K(x_i,x).
\]

(12)

In the equation, \( a_k^i \) is the \( i \)th coefficient of the \( k \)th eigenvalue of the matrix \( K \) for feature vector \( a_k \).

This thesis chooses \( K_{ij} \) as Gaussian based radial basis function

\[
K_{ij} = \exp \left[ -\frac{\|x_i-x_j\|^2}{(2\sigma^2)} \right].
\]

In the equation, \( \sigma \) is the standard deviation.

If the mapped data are non-zero mean, mean value method needs to be used on kernel function

\[
\overline{K}_{ij} = (K - I_M K - K I_M + I_M K I_M)_{ij} = \langle \varphi(x_i) \cdot \varphi(x_j) \rangle.
\]

In the equation, \( I_M \) are the \( M \times M \) constant matrices with \( 1/M \) as its element; \( \varphi(x_i) \) and \( \varphi(x_j) \) are mapped data after being treated with mean value method.

2.4. Improved Elman Neural Network

2.4.1. Modified Elman Neural Network. Elman regression neural network is a kind of typical dynamic neural network. Based on the feed forward neural network, it adds one more node, the context node. Through the delay and storage of the output of the hidden node by the context node, the output of the hidden node can self-link to the input of the hidden node, thereby enabling the network to be sensitive to the historical data, increasing the capacity of the network's processing capability of dynamic information and realizing dynamic modeling.

The characteristics of the existing modified Elman neural network is an added self-feedback connection with a fixed gain as \( \alpha \), so that the output of the context node at the time \( k \) equals the output of the hidden node at time \( k-1 \) plus \( \alpha \) times of the output of the context node at the time \( k-1 \) [12].

Figure 3 shows the network structure. The difference between the original and the improved Elman network is the different values for \( \alpha \). But both are fixed values. This thesis proposes a determination method for variable gain \( \alpha \) based on real time error analysis, which gives the network a better real time performance.

Comparing with the standard Elman network, the modified Elman network not only has a more flexible approximation to high order system essentially but can also achieve the same effect as gained by the standard Elman network when it adopts dynamic BP algorithm training.

The expressions between each layer of the modified Elman neural network are

\[
x(k) = f\left(w^{I1} x_{c}(k) + w^{I2} u(k-1)\right),
\]

\[
x_{c}(k) = \alpha x_{c}(k-1) + x(k-1),
\]

(15)

\[
y(k) = g\left(w^{O} x(k)\right).
\]

In the equation, connection weight \( w^{I1} \) is the connection weight matrix of the context node and the hidden node; \( w^{I2} \) is the connection weight matrix of the input node and the hidden node; \( w^{O} \) is the connection weight matrix of the hidden node and the output node; \( x_{c}(k) \) and \( x(k) \) represent the output of context node and the hidden node, respectively; \( y(k) \) represents the output of the output node, with \( 0 \leq \alpha \leq 1 \) as the self–join feedback gain factor. \( f(x) \) and \( g(x) \) are often hyperbolic tangent functions or sigmoid functions.

2.4.2. The \( \alpha \) Value Determination of the Modified Elman Neural Network. By analyzing the equation (15), we know that the bigger the \( \alpha \) value is, the greater the influence the output of the context node at time \( k-1 \) has on the output of the node at time \( k \). Suppose the calculation result is \( y(k) \) at the time \( k \), and the true value is \( y_d(k-1) \) and \( y_d(k) \) at the time \( k-1 \) and the time \( k \), respectively; then discuss the following 3 circumstances:

(1) when \( y_d(k-1) < y_d(k) \): if \( y(k) < y_d(k-1) \), make \( a(k+1) = 0 \); if \( y_d(k-1) < y(k) < y_d(k) \), make \( a(k+1) = ((y(k) - y_d(k-1))/(y_d(k) - y_d(k-1)))a(k) \);
if \( y_d(k) < y(k) \), make \( a(k + 1) = ((y(k) - y_d(k - 1))/(y_d(k) - y_d(k - 1)))a(k) \); if \( a(k + 1) > 1 \), make \( a(k + 1) = 1 \).

(2) when \( y_d(k - 1) < y_d(k) \): if \( y(k) < y_d(k - 1) \), make \( a(k + 1) = 0 \); if \( y_d(k - 1) < y(k) < y_d(k) \), make \( a(k + 1) = ((y(k) - y_d(k - 1))/(y_d(k) - y_d(k - 1)))a(k) \); if \( y_d(k) < y(k) \), make \( a(k + 1) = ((y(k) - y_d(k - 1))/(y_d(k) - y_d(k - 1)))a(k) \); if \( a(k + 1) > 1 \), make \( a(k + 1) = 1 \).

(3) when \( y_d(k - 1) = y_d(k) \): if \( y(k) < y_d(k) \), make \( a(k + 1) = 0 \); if \( y(k) > y_d(k) \), make \( a(k + 1) = (y(k)/y_d(k))a(k) \); if \( a(k + 1) > 1 \), make \( a(k + 1) = 1 \).

2.4.3. The Modified Elman Neural Network Learning Algorithm. The modified Elman neural network overall error objective function can be expressed as

\[
E(k) = \sum_{p=1}^{m} E_p(k),
\]

\[
E_p(k) = \frac{1}{2} (y_d(k) - y(k))^T (y_d(k) - y(k)).
\]

Calculating the partial derivatives of \( E_p(k) \) to weight values \( w_{ij}^{(1)}, w_{ij}^{(2)} \), and \( w_{ij}^{(3)} \), we can get the gradient descent algorithm of the modified Elman neural network:

\[
\Delta w_{ij}^{(3)} = \eta_3 \delta_i^0 x_j(k) \quad (i = 1, 2, \ldots, m; \ j = 1, 2, \ldots, n),
\]

\[
\Delta w_{ji}^{(2)} = \eta_2 \delta_j^h u_i(k - 1) \quad (j = 1, 2, \ldots, n; \ q = 1, 2, \ldots, r),
\]

\[
\Delta w_{jl}^{(1)} = \eta_1 \sum_{i=1}^{m} (\delta_i^0 w_{ij}^{(3)}) \frac{\partial x_j(k)}{\partial w_{jl}^{(1)}} \quad (j = 1, 2, \ldots, n; \ l = 1, 2, \ldots, n),
\]

\[
\delta_i^0 = (y_d(k) - y_i(k)) \varphi_i(\cdot),
\]

\[
\delta_j^h = \sum_{i=1}^{m} (\delta_i^0 w_{ij}^{(3)}) f_j'(\cdot)
\]

\[
\frac{\partial x_j(k)}{\partial w_{jl}^{(1)}} = f_j'(\cdot) x_l(k - 1) + a \frac{\partial x_j(k - 1)}{\partial w_{jl}^{(1)}} \quad (j = 1, 2, \ldots, m; \ l = 1, 2, \ldots, n).
\]

In the equation, \( \eta_1, \eta_2, \) and \( \eta_3 \) are the learning steps of \( w_{ij}^{(1)}, w_{ij}^{(2)} \), and \( w_{ij}^{(3)} \), respectively.

2.5. Modeling Based on Kernel Principal Component Analysis and Modified Elman Neural Network. Combining the Mahalonobis distance, kernel principal component analysis (KPCA), and improved Elman neural network, after Mahalonobis distance getting rid of the fault data among the sampling data, we can select the kernel principal components using KPCA and establish the neural network.

Specific steps are shown in Figure 4 and as follows: (1) generate the sample data set \( \{x_1, x_2, \ldots, x_n\} \) and preprocess the sample data (the data transform and error handling); (2) calculate the Mahalonobis distance \( \text{xd}_i \) between each sample data point and the nearest point, and the distance set of all the nearest points is expressed with \( \{\text{xd}_i\} \), where \( i = 1, 2, \ldots, n \). Then calculate the median value \( md \) of \( \{\text{xd}_i\} \) and compare each value of \( \{\text{xd}_i\} \) with \( 2md \); if any \( \text{xd}_i > 2md \), then the sample contains gross error data; otherwise the sample is normal. Sort the gross error data out from the new sample data set. (3) Select principle components by KPCA. There are total \( n \) principle components, when the accumulative contribution rate accounted for over 90%, and choose the first \( k \) principle components. (4) Use the \( k \) principle components as the input of the neural network, initialize parameters to be \( (\omega_{ij}^{(1)}, w_{ij}^{(2)}, c_{ij}, \sigma_{ij}) \), and choose a batch of data to train the neural network. (5) Use the last batch of data as prediction set. Keep the best parameters when the predicting error satisfies the requirements. Substitute these parameters (driving state) into the measurement and control applications. Use them as the initial weights of the network model (driving state) and get into the on-line learning.

3. Results and Discussion

To validate the KPC-IENN algorithm to be accurate and reliable, the double lane change test is adopted. Using SUV’s real vehicle parameters, virtual simulation of the sharp double lane change test is based on the ISO3888-1-1999 standard with sampling period of 0.025 s. During the test, the vehicle passes quickly through the benchmarks set on the road, and the passage is shown in Figure 5. Turn the steering wheel, with the observation of lateral acceleration to estimate the yaw velocity, the sideslip angle, and so on.

To stimulate the external disturbances that affect the vehicle, suppose the vehicle was affected by an instantaneous crosswinds gust shown in Figure 6 when passing through the double lane.

The samples were analyzed by KPCA. The accumulative contribution rate has already reached 90.09% as seen in Table 1. \( O(N_2) \) is the complexity to compute the kernel principal components.

Figures 7 and 8 are the estimated results of the vehicle longitudinal velocity, the yaw velocity, the sideslip angle, and other key state parameters based on the KPC-IENN.
Table 1: Cumulative contribution ratio of different principles.

<table>
<thead>
<tr>
<th>Principle number</th>
<th>Eigenvalues</th>
<th>Difference value</th>
<th>Percent variance (%)</th>
<th>Cumulative contribution ratio (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Z_1 )</td>
<td>5.123895</td>
<td>2.960055</td>
<td>47.4027</td>
<td>47.4027</td>
</tr>
<tr>
<td>( Z_2 )</td>
<td>2.16384</td>
<td>0.56343</td>
<td>21.19649</td>
<td>68.59919</td>
</tr>
<tr>
<td>( Z_3 )</td>
<td>1.60041</td>
<td>1.006635</td>
<td>15.67728</td>
<td>84.27647</td>
</tr>
<tr>
<td>( Z_4 )</td>
<td>0.59367</td>
<td>0.2184</td>
<td>5.81585</td>
<td>90.09232</td>
</tr>
<tr>
<td>( Z_5 )</td>
<td>0.375375</td>
<td>0.088725</td>
<td>3.67631</td>
<td>93.76905</td>
</tr>
<tr>
<td>( Z_6 )</td>
<td>0.286545</td>
<td>0.128625</td>
<td>2.80704</td>
<td>96.57613</td>
</tr>
<tr>
<td>( Z_7 )</td>
<td>0.15792</td>
<td>0.05124</td>
<td>1.54177</td>
<td>98.1233</td>
</tr>
<tr>
<td>( Z_8 )</td>
<td>0.106785</td>
<td>0.04179</td>
<td>1.04574</td>
<td>99.16905</td>
</tr>
<tr>
<td>( Z_9 )</td>
<td>0.06489</td>
<td>0.03822</td>
<td>0.588857</td>
<td>99.75791</td>
</tr>
<tr>
<td>( Z_{10} )</td>
<td>0.02667</td>
<td>0</td>
<td>0.242095</td>
<td>100</td>
</tr>
</tbody>
</table>

Judging from Figures 7 and 8, the vehicle nonlinear state based on the KPC-IENN algorithm is in line with the reality. The estimated values of this algorithm are very close to the stimulation values and the errors are under control. When \( t = 4 \sim 10 \) s, the vehicle is entering into the lane driving state. The vehicle longitudinal velocity, the yaw velocity, the sideslip angle, and other state parameters change drastically in this time interval. Judging from Figures 6, 7, and 8, the KPC-IENN algorithm used multiple suboptimal fading factor to force the estimated residual to have orthogonality and increased the tracking efficiency for the states with abrupt changes, thereby tracking the changing state values effectively.

4. Conclusions

This thesis proposed a KPC-IENN algorithm which is applicable to the states with abrupt changes and established a soft sensor method for the key states needed in the controlling of the lateral, longitudinal, and yaw stability of the vehicle body. In the sharp double lane change test, the estimated values gained by the KPC-IENN algorithm were proved to be highly accurate and satisfied the software performance requirements of the vehicle state estimators. The stimulation results of the Matlab/SIMULINK cosimulation showed that the KPC-IENN algorithm can estimate the speed, sideslip angle, and yaw velocity with accuracy and strong real time performance as well as an effective tracking speed and an outstanding property of restraining noises. It is suitable for online estimation of vehicle states. This thesis validated the accuracy and effectiveness of the KPC-IENN soft sensor algorithm using real vehicle parameters in stimulated environment. The next step is to use the established soft sensor model and the estimation algorithm in field test through hardware in loop simulation technique, which can further verify its practicability and reliability.

Competing Interests

The authors declare no conflict of interests.

Authors’ Contributions

All authors contributed in the theory and analysis developed in the manuscript and in finalizing the manuscript. All authors have read and approved the final manuscript.
**Figure 7:** Comparison of yaw rate between estimated value and simulated value (a) vehicle states not affected by crosswind (b) vehicle states affected by crosswind.

**Figure 8:** Comparison of side slip angle between estimated value and simulated value (a) vehicle states not affected by crosswind (b) vehicle states affected by crosswind.

**Acknowledgments**

The present work is supported by Shandong Province Crucial R&D Plan Project, China (no. 2015GGX105008), and Shandong Provincial Science and Technology Development Plan Project, China (no. 2014GGX105001).

**References**


