Hindawi
Complexity
Volume 2021, Article ID 5124841, 8 pages
https://doi.org/10.1155/2021/5124841

Research Article

A Nonlinear System State Estimation Method Based on Adaptive Fusion of Multiple Kernel Functions

Daxing Xu[1] **Aiyu Hu,**[2] **Xuelong Han,**[1] **and Lu Zhang**[1]

[1]*College of Electrical and Information Engineering, Quzhou University, Quzhou 324000, China*
[2]*Zhejiang Juhua Group Co., Ltd., Quzhou 324000, China*

Correspondence should be addressed to Daxing Xu; daxingxu@163.com

Received 12 April 2021; Accepted 18 June 2021; Published 24 June 2021

Academic Editor: Carlos Aguilar-Ibanez

With the development of the industry, the physical model of controlled object tends to be complicated and unknown. It is particularly important to estimate the state variables of a nonlinear system when the model is unknown. This paper proposes a state estimation method based on adaptive fusion of multiple kernel functions to improve the accuracy of system state estimation. First, a dynamic neural network is used to build the system state model, where the kernel function node is constructed by a weighted linear combination of multiple local kernel functions and global kernel functions. Then, the state of the system and the weight of the kernel functions are put together to form an augmented state vector, which can be estimated in real time by using high-degree cubature Kalman filter. The high-degree cubature Kalman filter performs adaptive fusion of the kernel function weights according to specific samples, which makes the neural network function approximate the real system model, and the state estimate follows the real value. Finally, the simulation results verify the feasibility and effectiveness of the proposed algorithm.

1. Introduction

With the development of industrial technology, the physical model of control object is also becoming more and more complicated [1, 2]. The linear system model can no longer accurately describe the system model, leading to new challenges for linear control theory. Although the classical control theory for linear systems is relatively mature, we are often faced with different types of nonlinear systems in practice; the characteristics include model nonlinearity, time variance, uncertain terms, and so on [3–5]. The state estimation of nonlinear systems has attracted more attention in recent decades. For nonlinear systems with unknown parameters, adaptive control with parameter approximation has been fully studied, and a reasonable adaptive law is designed to achieve stable control effects. For nonlinear systems whose models are completely unknown, the function approximation theory of neural networks makes it an effective approximation tool.

The kernel function is the core of the neural network [6–11]. The commonly used kernel functions include Gaussian kernel function, Fourier kernel function, linear kernel function, polynomial kernel function, and sigmoid kernel function. For the selection of the kernel function type, the most commonly used method is the cross-validation method, which uses different kernel functions to train the samples and selects the kernel function with the smallest overall error as the optimal kernel function. Machine training based on a single kernel function in a single feature space has great defects when dealing with uneven sample distribution. For example, an actual sample feature is a fusion of two basic features, where the first feature obeys a polynomial distribution and the second feature obeys the normal distribution. A single kernel function can only describe the characteristics of a certain aspect of the data, and it cannot properly represent the characteristics of different distributions [12]. Literatures [13–15] use a hybrid kernel with strong local and global information processing capabilities to deal with classification problems. This method can make up for the shortcomings of a single kernel function in processing sample local and global information, but it lacks an effective method to optimize the weighting coefficients of the two basic kernel functions. Kernel functions are divided into local kernel functions and global kernel functions according to their local and global capabilities. The local kernel function has a strong learning ability, while the global kernel function has a strong extrapolation ability [16, 17]. There are a variety of existing kernel functions, each with its own characteristics, and different kernel functions have different nonlinear processing capabilities. On the basis of maintaining the basic characteristics of the original kernel function, multiple local kernel functions and global kernel functions are linearly combined into a new kernel function, that is, the multiple kernel functions absorb the advantages of the local kernel function and the global kernel function. It can accurately reflect the characteristics of the actual sample [18, 19]. However, how to choose the weights of multiple kernel functions is a challenging problem.

In 1960, Kalman introduced state variables into the filtering theory and proposed the famous Kalman filter. But it is suitable for the linear time-invariant system model. For nonlinear systems, researchers have successively proposed extended Kalman filter, unscented Kalman filter, cubature Kalman filter, and so on [20–22]. For the problem of state estimation of nonlinear systems with unknown models, many scholars have combined the unscented Kalman filter algorithm with neural networks to solve practical problems. Literatures [21, 23, 24] use a neural network to establish a one-dimensional nonlinear time series model. The input and output of the model are the value of the time series at the current time and the next time, respectively. The unscented Kalman filter algorithm is used to simultaneously update the network weights and time series, and the method is compared with the standard neural network learning algorithm and a separate unscented Kalman filter estimation algorithm. It proves that the estimation effect of this method is better than other methods. However, when the dimensionality of the system is high, the unscented Kalman filter faces the problem of dimensionality catastrophe, which causes its estimation performance to be greatly reduced. In order to further improve the filtering accuracy, literature [25] proposed a high-degree cubature Kalman with arbitrary order volume rules. The filtering algorithm uses radial integration rules to optimize sigma points and weights, which greatly enhances the ability to handle high-dimensional nonlinear states, and the estimation accuracy and stability are also significantly improved.

The selection of multiple function fusion coefficients is actually a process of constant weight adjustment. This paper regards the fusion coefficients as part of the augmented system state. Since the kernel function is often nonlinear, the training of neural networks can be regarded as the state estimation of the nonlinear system. The problem is that the estimation of the fusion coefficient of the multiple function and the system state can be regarded as the optimal estimation of the state vector in filtering. Therefore, the main contributions of this paper include the following:

(1) Building the unknown system state model using the weighted linear combination of multiple local functions and global functions. Then, establishing a nonlinear system model with augmented state.

(2) By using the high-degree cubature Kalman filter to estimate the fusion coefficient and the system state in real time, a nonlinear system state estimation method based on adaptive fusion of multiple functions is proposed, and the optimal fusion coefficients are selected to improve the accuracy of state estimation.

2. Problem Formulation

Denote $K_l^{t1}(x_i, x_j)$ $(t1 = 1, 2, \ldots, M)$ and $K_g^{t2}(x_i, x_j)$ $(t2 = 1, 2, \ldots, N)$ as the local kernel functions and the global kernel functions, respectively. wl^{t1} and wg^{t2} represent the corresponding weight coefficients for the above kernel function, respectively. Then, the multiple kernel functions can be expressed as follows:

$$K_m(x_i, x_j) = \sum_{t1=1}^{M} wl^{t1} \cdot K_l^{t1}(x_i, x_j) + \sum_{t2=1}^{N} wg^{t2} \cdot K_g^{t2}(x_i, x_j). \tag{1}$$

Further, the structure of the neural network state space model based on the multiple kernel function is shown in Figure 1.

$x_1, x_2, \ldots, x_n$ represent the input sample nodes, and W_{ij}, W_{ki} stand for the weight coefficients among all layers. $y_1, y_2, \ldots, y_m$ denote the output sample nodes. The neural network model structure has three node layers, namely, input layer, hidden layer, and output layer. It is connected by weight coefficients, the input and output layers are at both ends, and the number of nodes in the middle hidden layer is selected according to actual requirements.

Since the system model is unknown, this paper uses a neural network based on multiple kernel functions to approximate it. Specifically, the nonlinear systems can be described as follows:

$$\begin{aligned} x_k &= \mathbf{f}(x_{k-1}) + w_k, \\ z_k &= \mathbf{h}(x_k) + v_k, \end{aligned} \tag{2}$$

where x_k is an n-dimensional state vector, z_k is an m-dimensional observation vector, functions $\mathbf{f}$ and $\mathbf{h}$ are known nonlinear functions, and $\{w_k\}$ and $\{v_k\}$ are independent zero-mean Gaussian white noise.

For general nonlinear systems, under the Gaussian hypothesis, the basic theory of Bayesian estimation can be combined with any order cubature rule to derive a high-order cubature Kalman filter. Similar to the unscented Kalman filter structure, high-order cubature Kalman filter is also divided into two steps: state prediction (time update) and measurement update. The high-degree cubature Kalman filter uses the phase difference cubature rule to solve the problem of dimensional explosion in high-dimensional systems. High-degree cubature rules satisfy

$$\begin{aligned} I_{U_n} = \overline{w}_{s1} \sum_{j=1}^{n(n-1)} \left(g_s(s_j^+) + g_s(-s_j^+) + g_s(s_j^-) + g_s(-s_j^-) \right) \\ + \overline{w}_{s2} \sum_{j=1}^{n} \left(g_s(e_j) + g_s(-e_j) \right), \end{aligned} \tag{3}$$

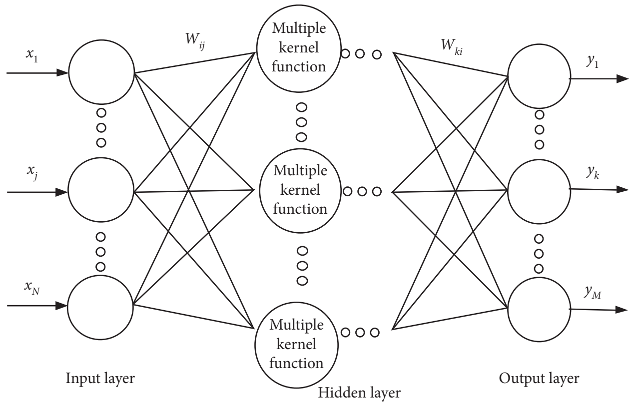

FIGURE 1: Neural network model structure based on multiple kernel function.

where $s = [s_1, s_1, \ldots, s_n]^T, U_n = \{s \in R^n\colon s_1^2 + s_2^2 + \cdots + s_n^2 = 1\}$, and e_j is the j-th column of the unit vector matrix of n-dimensional space R^n. $g_s\,(\,)$ is a general nonlinear function that has different forms in different filtering steps. s_j^+ and s_j^- are the sets of points as shown below:

$$\begin{aligned} \{s_j^+\} &= \left\{\sqrt{\frac{1}{2}}\,(\mathbf{e}_k + \mathbf{e}_l)\colon k < l, k, l = 1, 2, \ldots, n,\right\}, \\ \{s_j^-\} &= \left\{\sqrt{\frac{1}{2}}\,(\mathbf{e}_k - \mathbf{e}_l)\colon k < l, k, l = 1, 2, \ldots, n\right\}. \end{aligned} \tag{4}$$

The weight coefficients $\overline{w}_{s1}$ and $\overline{w}_{s2}$ are

$$\begin{aligned} \overline{w}_{s1} &= \frac{A_n}{n(n+2)}, \\ \overline{w}_{s2} &= \frac{(4-n)A_n}{(2n(n+2))}, \end{aligned} \tag{5}$$

where $A_n = 2\sqrt{\pi^n}/\Gamma(n/2)$ is the surface area of the unit sphere and $\Gamma(z) = \int_0^\infty \exp(-\lambda)\lambda^{z-1}\mathrm{d}\lambda$. According to the moment matching method, when $n = 2$, the weight is

$$\begin{cases} w_1 = \dfrac{\Gamma(n/2)}{(n+2)}, \\ w_2 = \dfrac{n\Gamma(n/2)}{(2(n+2))}. \end{cases} \tag{6}$$

In this paper, we study how to combine high-degree cubature filter and neural networks to model unknown nonlinear systems and how to estimate the state of the system. Consequently, the addressed problem in this paper can be summarized as follows:

(1) For the multiple kernel function in (1), how to construct a unified model of the combination of multiple kernel function weights and state variables to satisfy the requirements of the filter.

(2) How to design a high-degree cubature filter to adaptively estimate the system state and kernel function weights.

3. Main Results

3.1. Establishment of Nonlinear System Model. When the model of the system is unknown, the neural network is used to approximate the system model, and then the optimal network node weight coefficients need to be solved. Meanwhile, the state is also unknown, and the state and weight coefficients are related. Therefore, we combine the original state x_{k-1} and weight coefficient of the kernel functions wl^{t1}, wg^{t2} as a new state $x_k^a = [x_{k-1}, wl_k^1, \ldots, wl_k^M, wg_k^1, \ldots, wg_k^N]$. Then, the original system equation and the augmented equation of the weight coefficient equation are considered as a new system model:

$$x_k^a = \begin{bmatrix} x_k \\ wl_k^1 \\ . \\ wl_k^M \\ wg_k^1 \\ . \\ wg_k^N \end{bmatrix} = \begin{bmatrix} f^1(x_{k-1}) \\ f^2(x_{k-1}) \\ . \\ wl_{k-1}^1 \\ . \\ wl_{k-1}^M \\ wg_{k-1}^1 \\ . \\ wg_{k-1}^N \end{bmatrix} + w_{k-1} = \begin{bmatrix} f(x_{k-1}) \\ . \\ wl_{k-1}^M \\ wg_{k-1}^1 \\ . \\ wg_{k-1}^N \end{bmatrix} + w_{k-1} = f^a(x_{k-1}^a) + w_{k-1}, \tag{7}$$

$$z_k = \mathbf{h}(x_k^a) + v_k, \tag{8}$$

where $f^j(x_{k-1})$ is the mathematical model established by the neural network for the nonlinear system:

$$f^{j}(x_k)=\sum_{l=i}^{L}\left(W_{j,l}^{2}\left(g\left(\sum_{i=1}^{N}x_{i,k}W_{i,l}^{1}\right)\right)\right),\quad j=1,2,\ldots,N, \tag{9}$$

where $g(x)$ is the sigmoid kernel function of the neural network, which has been proved to have good global classification performance in the application of neural network because it is a smooth function that is convenient to find derivatives [26]. W_k is the weight coefficient of the neural network. The process noise w_k and observation noise v_k of the new system are independent zero-mean Gaussian white noise, and the corresponding covariance matrices are $\mathbf{Q}_k$ and $\mathbf{R}_k$.

Remark 1. Since the neural network based on multiple kernels is used to approximate the nonlinear function, it is necessary to solve the weight coefficients of the local kernel function and the global kernel function. By assuming that the weight coefficients are disturbed by Gaussian white noise, the coefficients and the state can be combined into an augmented state vector, so that a nonlinear system model based on the augmented state can be established.

3.2. Adaptive Fusion Filtering. In the past two decades, the extended Kalman filter has been widely used in the training of a neural network and as an optimizer of fuzzy membership functions for fuzzy classifiers. Tuning of multiple parameters of SVM can be viewed as an identification problem of a nonlinear dynamic system. Due to the truncation error introduced by the extended Kalman filter when linearizing the nonlinear system, the state estimation accuracy is low. The high-degree cubature Kalman filter has higher estimation accuracy than the extended Kalman filter because it uses radial integration rules to optimize sigma points and weights. Therefore, the high-degree cubature Kalman filter is exploited to estimate the the augmented state.

The parameter estimation model is established in the previous section, and the adaptive selection method of fusion coefficients is given below. The estimation process of the entire augmented state is shown in Figure 2. First, select some local kernel functions with learning ability and some global kernel functions with generalization ability from the commonly used kernel functions to form a multiple kernel function. Then, the weighted fusion coefficient and the original state are combined to form an augmented state vector, then the high-degree cubature Kalman filter is used for time update, and then the real output value of the data set is used for the high-degree cubature Kalman filter measurement update.

The specific algorithm is given as follows:

Update the state:

(1) At time k, assume that the error covariance $\mathbf{P}_{k-1|k-1}$ at time k-1 is known, and the factorization is

$$\mathbf{P}_{k-1|k-1}=\mathbf{S}_{k-1|k-1}\mathbf{S}_{k-1|k-1}^{T}, \tag{10}$$

where the vector $\mathbf{S}_{k-1|k-1}$ is the Cholesky factorization of $\mathbf{P}_{k-1|k-1}$.

(2) Compute the cubature points:

$$X_{i,k-1|k-1}^{a}=\mathbf{S}_{k-1|k-1}\boldsymbol{\xi}_i+\widehat{x}_{k-1|k-1}^{a},\quad (i=1,2,\ldots,m), \tag{11}$$

where $m=2n$, and the vector ξ_i is

$$\boldsymbol{\xi}_i=\begin{cases}[0,0,\ldots,0]^{T}, & i=0,\\ \beta s_i^{+}, & i=1,2,\ldots,\dfrac{n(n-1)}{2}\\ -\beta s_{i-(n(n-1)/2)}^{+}, & i=\dfrac{n(n-1)}{2}+1,\ldots,n(n-1),\\ \beta s_{i-n(n-1)}^{-}, & i=n(n-1)+1,\ldots,\dfrac{3n(n-1)}{2},\\ -\beta s_{i-3n(n-1)/2}^{-}, & i=\dfrac{3n(n-1)}{2}+1,\ldots,2n(n-1),\\ \beta e_{i-2n(n-1)}, & i=2n(n-1)+1,\ldots,n(2n-1),\\ -\beta e_{i-n(2n-1)}, & i=n(2n-1)+1,\ldots,2n^{2},\end{cases} \tag{12}$$

where $\beta=\sqrt{n+2}$, $\mathbf{e}_i$ represents an n-dimensional unit vector, and its i-th element is 1. s_j^{+},s_j^{-} are

$$\begin{cases}s_j^{+}=\sqrt{\dfrac{1}{2}}\left(e_p+e_q\right), & p<q,\,p,q=1,2,\ldots,n,\\ s_j^{-}=\sqrt{\dfrac{1}{2}}\left(e_p-e_q\right), & p<q,\,p,q=1,2,\ldots,n.\end{cases} \tag{13}$$

(3) Calculate the cubature points after propagation of state equation $(i=1,2,\ldots,m)$:

$$\mathbf{X}_{i,k|k-1}^{a*}=\mathbf{f}\left(X_{i,k-1|k-1}^{a}\right). \tag{14}$$

(4) Compute one-step state prediction:

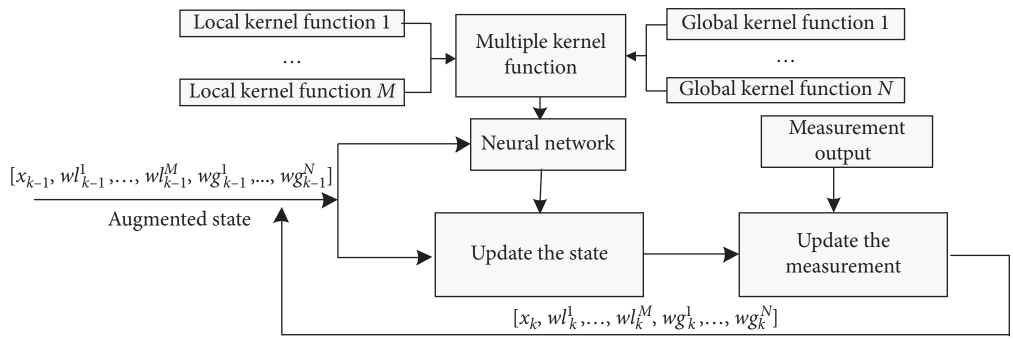

Figure 2: State estimation system.

$$\hat{x}^a_{k|k-1} = \sum_{i=1}^{m} w_i \mathbf{X}^{a*}_{i,k|k-1}, \tag{15}$$

where the weights w_i are

$$w_i = \begin{cases} \dfrac{2}{n+2}, & i = 0, \\ \dfrac{1}{(n+2)^2}, & i = 1, 2, \ldots, 2n(n-1), \\ \dfrac{(4-n)}{(n+2)^2}, & \\ i = 2n(n-1)+1, 2n(n-1)+2, \ldots, 2n^2. \end{cases} \tag{16}$$

(5) Calculate the one-step prediction error covariance matrix:

$$\mathbf{P}_{k|k-1} = \sum_{i=1}^{m} w_i \mathbf{X}^{a*}_{i,k|k-1} \mathbf{X}^{a*T}_{i,k|k-1} - \hat{x}^a_{k|k-1} \left(\hat{x}^a_{k|k-1}\right)^T + \mathbf{Q}_{k,k-1}. \tag{17}$$

Update the measurement:

(1) Factorization:

$$\mathbf{P}_{k|k-1} = \mathbf{S}_{k|k-1} \mathbf{S}^T_{k|k-1}. \tag{18}$$

(2) Calculate the state cubature point after update:

$$\mathbf{X}_{i,k|k-1} = \mathbf{S}_{k|k-1} \xi_i + \hat{x}^a_{k|k-1}, \quad (i = 1, 2, \ldots, m). \tag{19}$$

(3) Compute the cubature point after the measurement equation has propagated:

$$\mathbf{Z}_{i,k|k-1} = \mathbf{h}\left(\mathbf{X}^a_{i,k|k-1}\right). \tag{20}$$

(4) Calculate one-step measurement and prediction at time k:

$$\hat{z}_{k|k-1} = \sum_{i=1}^{m} w_i \mathbf{Z}_{i,k|k-1}. \tag{21}$$

(5) Calculate the innovation covariance matrix:

$$\mathbf{P}_{zz,k|k-1} = \sum_{i=1}^{m} w_i \mathbf{Z}_{i,k|k-1} \mathbf{Z}^T_{i,k|k-1} - \hat{z}_{k|k-1} \hat{z}^T_{k|k-1} + \mathbf{R}_k. \tag{22}$$

(6) Compute the one-step prediction cross covariance matrix:

$$\mathbf{P}_{xz,k|k-1} = \sum_{i=1}^{m} w_i \mathbf{X}_{i,k|k-1} \mathbf{Z}^T_{i,k|k-1} - \hat{x}^a_{k|k-1} \hat{z}^T_{k|k-1}. \tag{23}$$

(7) Calculate the gain matrix:

$$\mathbf{K}_k = \mathbf{P}_{xz,k|k-1} \mathbf{P}^{-1}_{zz,k|k-1}. \tag{24}$$

(8) Update state as follows:

$$\hat{x}_{k|k} = \hat{x}^a_{k|k-1} + K_k\left(\mathbf{z}_k - \hat{z}_{k|k-1}\right). \tag{25}$$

(9) Error covariance matrix can be obtained by

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{P}_{zz,k|k-1} \mathbf{K}^T_k. \tag{26}$$

Remark 2. For the known nonlinear system described in formulas (10) and (11), given the initial state of the state, the high-degree cubature Kalman filter can be performed according to the above two steps of time update and measurement update to obtain an augmented state vector value.

4. Simulation Example

The neural network approximation of the system model using the nonlinear filtering algorithm based on the Kalman filter framework has many practical applications, for example, the tracking problem of a moving target at a constant speed in a two-dimensional plane [22], the state estimation problem of the concentration and temperature of the reactant in the non-isothermal chemical stirring tower reactor [27], etc. The example considered in this paper is a commonly used discrete model of a nonlinear system as follows [28]:

$$
\begin{aligned}
x(k+1) &= \begin{bmatrix} 0.8 + 0.05\sin(0.1k) & 0.06 \\ 0.1 & -0.3 + 0.2\sin(0.1k) \end{bmatrix} \\
&\quad * x(k) + w(k), \\
y(k) &= x_1(k) + x_2(k) + v(k),
\end{aligned}
\tag{27}
$$

where $w(k)$ and $v(k)$ are independent zero-mean Gaussian white noises, with the variances $\mathbf{Q}(k) = \begin{bmatrix} 2.3478 & 0.7314 \\ 0.7314 & 2.6532 \end{bmatrix}$ and $R(k) = 0.8$, respectively. The initial state is $x_0 = [\,9.5 \;\; 4.5\,]^T$, and its estimate is set as $\widehat{x}_0 = [\,6.5 \;\; 2.2\,]^T$. The initial state error covariance matrix is $P_0 = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.3 \end{bmatrix}$.

The neural network model has two input nodes, two output nodes, and two hidden nodes. The simulation environment is Intel i5 CPU with 4G memory, and the simulation software uses Matlab R2013a.

In this simulation, the linear combination of Gaussian kernel function, Fourier kernel function, and linear kernel function is selected as the multiple kernel function, and the weight coefficients are, respectively, denoted as wl^1, wl^2, and wg^1. For convenience of comparison, denote MAEE as mean absolute estimation error, and we simply mark the algorithms as follows:

EAFMKF: estimation algorithm based on adaptive fusion of multiple kernel functions.

ESKF: estimation algorithm based on single sigmoid kernel function.

The simulation results are shown in Figures 3–7 and Table 1.

From the estimation curves of Figures 3 and 4, both EAFMKF and ESKF can perform a good tracking estimation on the two states, indicating that both algorithms are effective. From the estimation error curves of Figures 5 and 6, the error curve of ESKF is generally above EAFMKF, which means that the error of ESKF is obviously greater than that of EAFMKF. From the statistics of Table 1, it can be seen that the time consumption of EAFMKF is slightly higher than that of ESKF, but the accuracy of the state estimation of EAFMKF is much higher than that of ESKF. Specifically, estimation error of ESKF is more than twice that of the EAFMKF. This is mainly because the multiple function can accurately describe the characteristics of the sample by adaptively adjusting the weight coefficients, thereby making the established state model more accurate. As shown in Figure 7, the weights wl^1, wl^2, and wg^1 of the neural network are an adaptive adjustment process in the whole estimation process, and they quickly stabilize to the corresponding

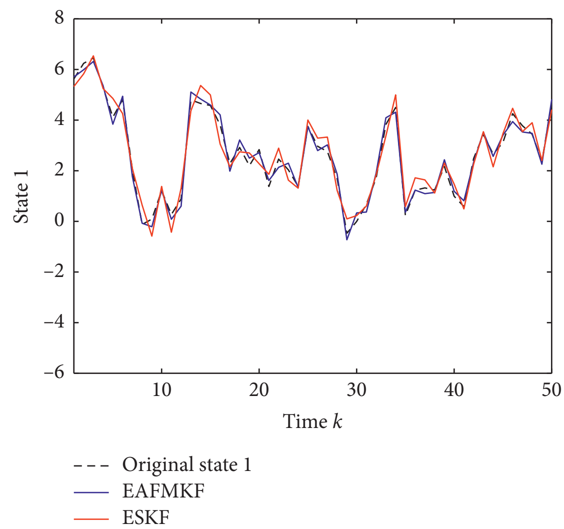

Figure 3: Estimation curve for state 1.

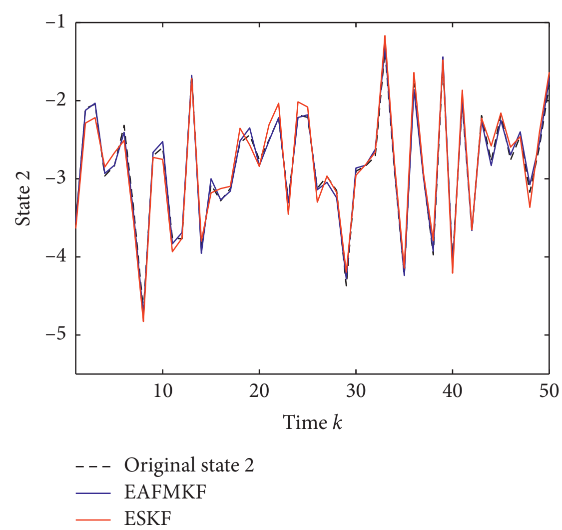

Figure 4: Estimation curve for state 2.

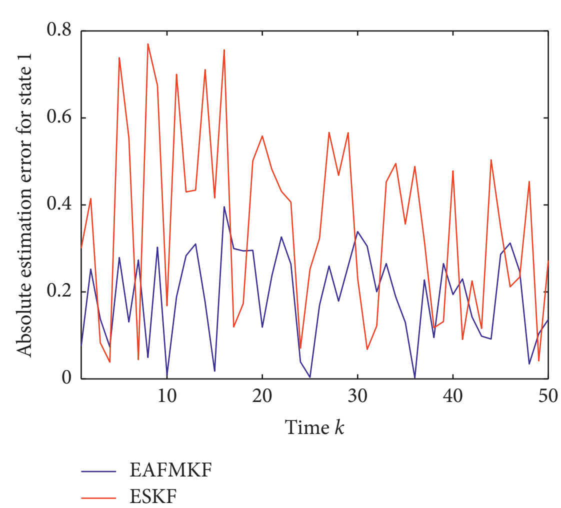

Figure 5: Estimation error curve for state 1.

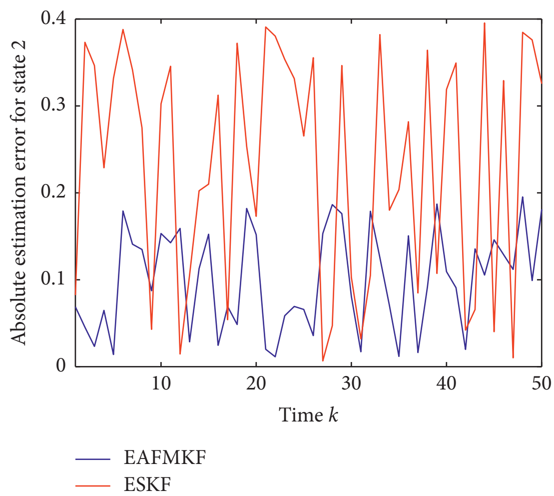

Figure 6: Estimation error curve for state 2.

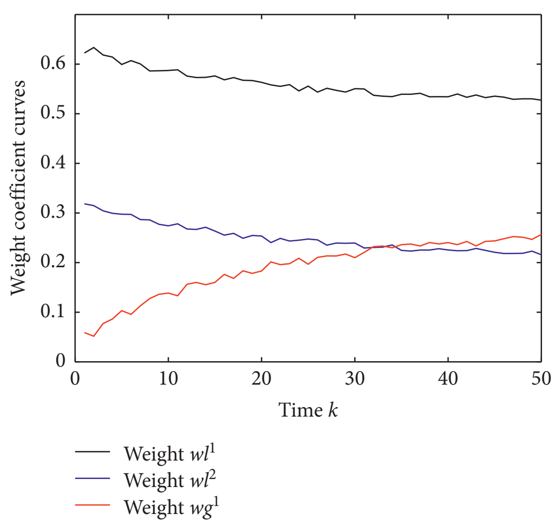

Figure 7: Weight coefficient curves.

Table 1: Comparison of estimation errors and time consumption.

Algorithm	MAEE of state 1	MAEE of state 2	Algorithm time consumption
EAFMKF	0.1811	0.0984	0.912
ESKF	0.3323	0.2290	0.610

values of 0.52, 0.21, and 0.27. These demonstrate the effectiveness of cubature Kalman filtering and neural network estimation algorithms.

5. Conclusions

This paper proposed a state estimation algorithm based on adaptive fusion of multiple kernel function for nonlinear systems with unknown state model. The system state model is built by using multiple kernel function, which is constructed by some local kernel functions and global kernel functions. Under this case, the characteristics of the actual sample can be fully characterized. Then, we put the weights of the multiple kernel function and the original state together as a augmented state. Futher, the high-degree cubature Kalman filter algorithm is used to estimate the augmented state in real time. Thus, we can obtain the optimal weight coefficients by the adaptive fusion of multiple kernel function, and the accuracy of the original states is significantly improved. Finally, a simulation example verifies the effectiveness of the proposed algorithm. In some practical applications, the state dimensionality is often very high. The next research content is how to choose a suitable approximate neural network to establish the state transition equation when the state dimension is high. Under this case, good state estimation results can be obtained while reducing the dimensionality of the estimation problem.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

[1] H. B. Zhu and J. B. Ding, “A dynamic variance-based triggering scheme for distributed cooperative state estimation over wireless sensor networks,” *Complexity*, vol. 2021, 12 pages, Article ID 7851080, 2021.

[2] H. Wang, S. S. Xie, W. X. Wang, L. Wang, and J. B. Peng, “Investigation of unmeasured parameters estimation for distributed control systems,” *Complexity*, vol. 2020, 15 pages, Article ID 7518039, 2020.

[3] L. B. Prasad, H. O. Gupta, and B. Tyagi, “Intelligent control of nonlinear inverted pendulum dynamical system with disturbance input using fuzzy logic systems,” *Recent Advances in Electrical and Electronic Engineering*, pp. 136–141, 2011.

[4] W. Zhao, L. Tang, and Y. J. Liu, “Disturbance observer-based adaptive neural network control of marine vessel systems with time-varying output constraints,” *Complexity*, vol. 2020, 12 pages, Article ID 6641758, 2020.

[5] G. Khanna, S. K. Chaturvedi, and S. Soh, “Two-terminal reliability analysis for time-evolving and predictable delay-tolerant networks,” *Recent Advances in Electrical and Electronic Engineering*, vol. 13, no. 3, pp. 396–404, 2020.

[6] M. A. Islas, J. de Jesus Rubio, S. Muniz et al., “A fuzzy logic model for hourly electrical power demand modeling,” *Electronics*, vol. 10, no. 4, p. 448, 2021.

[7] J. de Jesus Rubio, “SOFMLS: online self-organizing fuzzy modified least-squares network,” *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 6, pp. 1296–1309, 2009.

[8] H. S. Chiang, M. Y. Chen, and Y. J. Huang, “Wavelet-based EEG processing for epilepsy detection using fuzzy entropy and associative petri net,” *IEEE Access*, vol. 7, pp. 103255–103262, 2019.

[9] J. J. de Rubio, “Stability analysis of the modified levenberg-marquardt algorithm for the artificial neural network training,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2020.

[10] J. A. Meda-Campaña, “On the estimation and control of nonlinear systems with parametric uncertainties and noisy outputs,” *IEEE Access*, vol. 6, pp. 31968–31973, 2018.

[11] F. Furlán, E. Rubio, H. Sossa et al., “CNN based detectors on planetary environments: a performance evaluation,” *Frontiers in Neurorobotics*, vol. 14, p. 85, 2020.

[12] C.W. Hsu, C. C. Chang, and C.J. Lin, “A practical guide to support vector classification,” pp. 1–16, Department of Computer Science, National Taiwan University, Taiwan, China, 2020, Doctoral Thesis.

[13] E. B. Huerta, B. Duval, and J.K. Hao, “A hybrid GA/SVM approach for gene selection and classification of microarray data,” *Workshops on Applications of Evolutionary Computation,* pp. 34–44, Berlin, Heidelberg, 2006.

[14] J. Zhou, T. Bai, and C. Suo, “The SVM optimized by culturegenetic algorithm and its application in forecasting share price,” in *Proceedings of the 2015 IEEE Conference on Granular Computing*, pp. 838–843, Article ID 4664698, Hangzhou, China, August 2008.

[15] J. Zhou, T. Bai, A. Zhang, and J. Tian, “The integrated methodology of wavelet transform and GA based-SVM for forecasting share price,” in *Proceedings of the 2016 IEEE Conference on Information and Automation*, pp. 729–733, Article ID 4608094, Changsha, China, June 2008.

[16] F. Kuang, S. Zhang, Z. Jin et al., “A novel SVM by combining kernel principal component analysis and improved chaotic particle swarm optimization for intrusion detection,” *Soft Computing*, pp. 1–13, 2015.

[17] Z. Xu, Z. Y. Dong, and W. Q. Liu, “Short-term electricity price forecasting using wavelet and SVM techniques,” in *Proceedings of the Third International DCDIS Conference on Engineering Applications and Computational Algorithms*, pp. 15–18, Guelph, Canada, May 2003.

[18] S. K. Aggarwal, L. M. Saini, and A. Kumar, “Electricity price forecasting in deregulated markets: a review and evaluation,” *International Journal Electric Power Energy System*, vol. 31, no. 1, pp. 13–22, 2019.

[19] T. Mu and A. K. Nandi, “Automatic tuning of L2-SVM parameters employing the extended Kalman filter,” *Expert Systems*, vol. 26, no. 2, pp. 160–175, 2009.

[20] H. E. Yao, C. Zhou, and L. Y. Zheng, “Detection method against false data injection attack based on extended Kalman filter,” *Electric Power*, vol. 50, no. 10, pp. 35–40, 2019.

[21] X. Wu and Y. Wang, “Extended and unscented Kalman filtering based feedforward neural networks for time series prediction,” *Applied Mathematical Modelling*, vol. 36, no. 3, pp. 1123–1131, 2016.

[22] Z. T. Hu, G. Y. Yuan, and Y. M. Hu, “Training method of neural network based on cubature Kalman filter,” *Control and Decision*, vol. 31, no. 2, pp. 355–360, 2016.

[23] H. L. Li, J. Wang, Y.Q. Chen et al., “On neural network-aided training algorithm based on the unscented Kalman filter,” in *In Proceedings of the 29th Chinese Control Conference*, pp. 1447–1450, Beijing, China, July 2010.

[24] Z. Ronghui and J. Wan, “Neural network-aided adaptive unscented Kalman filter for nonlinear state estimaiion,” *IEEE Signal Processing Letter*, vol. 13, no. 7, pp. 445–448, 2018.

[25] B. Jia, M. Xin, and Y. Cheng, “High-degree cubature Kalman filter,” *Automatica*, vol. 49, no. 2, pp. 510–518, 2013.

[26] Y. Ito, “Representation of functions by superpositions of a step or sigmoid function and their applications to neural network theory,” *Neural Networks*, vol. 4, no. 3, pp. 385–394, 1991.

[27] K. Salahshoor and A. S. Kamalabady, “On-line multivariable identification by adaptive RBF neural networks based on UKF learning algorithm,” in *In Proceedings of the IEEE 2008 Chinese Control and Decision Conference*, pp. 4754–4759, Yantai, China, July 2008.

[28] J. L. Zhou, D. H. Zhou, H. Wang et al., “Distribution function tracking filter design using hybrid characteristic functions,” *Automatica*, vol. 46, no. 1, pp. 101–109, 2010.