



Article

A Neuron-Based Kalman Filter with Nonlinear Autoregressive Model

Yu-ting Bai ^{1,2} , Xiao-yi Wang ^{1,2,*}, Xue-bo Jin ^{1,2,*} , Zhi-yao Zhao ^{1,2} and Bai-hai Zhang ³ 

¹ School of Computer and Information Engineering, Beijing Technology and Business University, Beijing 100048, China; baiyuting@btbu.edu.cn (Y.-t.B.); zhaozy@btbu.edu.cn (Z.-y.Z.)

² Beijing Key Laboratory of Big Data Technology for Food Safety, Beijing Technology and Business University, Beijing 100048, China

³ School of Automation, Beijing Institute of Technology, Beijing 100811, China; smczhang@bit.edu.cn

* Correspondence: wangxy@btbu.edu.cn (X.-y.W.); jinxuebo@btbu.edu.cn (X.-b.J.)

Received: 3 December 2019; Accepted: 2 January 2020; Published: 5 January 2020



Abstract: The control effect of various intelligent terminals is affected by the data sensing precision. The filtering method has been the typical soft computing method used to promote the sensing level. Due to the difficult recognition of the practical system and the empirical parameter estimation in the traditional Kalman filter, a neuron-based Kalman filter was proposed in the paper. Firstly, the framework of the improved Kalman filter was designed, in which the neuro units were introduced. Secondly, the functions of the neuro units were excavated with the nonlinear autoregressive model. The neuro units optimized the filtering process to reduce the effect of the unpractical system model and hypothetical parameters. Thirdly, the adaptive filtering algorithm was proposed based on the new Kalman filter. Finally, the filter was verified with the simulation signals and practical measurements. The results proved that the filter was effective in noise elimination within the soft computing solution.

Keywords: kalman filter; nonlinear autoregressive; neural network; noise filtering

1. Introduction

In the typical control systems, the measurement is the primary component that senses the system status and provides the input information. The measurement accuracy influences the control effect directly. Especially in intelligent terminals, such as industrial robots, unmanned aerial vehicles and unmanned vehicles, various sensors are implemented to measure the motion state and working condition. The sensors are expected to be of high precision. However, the precision is limited by the manufacturing technique when the intelligent terminals are with small size and low cost relying on the micro-electromechanical sensors [1,2]. For the complicated noises, it is essential to filter the noises. Then the motion state and condition can be estimated accurately for the target tracking and control.

In the noise filtering and state estimation field, many methods have been studied, such as the wavelet filter [3], time-frequency peak filtering [4], empirical mode decomposition [5] and the Kalman filter [6]. These filtering and estimation algorithms are often based on the mathematical models and established using the iterative schemes [7–9] or recursive schemes [10,11]. Some filtering-based estimation algorithms use input-output representations [12,13], and others use state-space models [14–16]. Among these methods, Kalman filter is a state estimation algorithm based on the state-space model. It introduces the state space into the stochastic estimation theory and obtains the optimal estimation without requiring a vast amount of historical data. However, it is obvious that the Kalman filter depends highly on some assumptions that the system model is linear, the process and measurement noises are standard Gaussian, and their covariance matrixes are all known. When these assumptions are seriously limited in reality, two categories of methods have been explored. On the one hand,

the adaptive Kalman filter (AKF) was proposed focusing on the parameter adjustment to approximate the filtering process to the practical system. Then common AKF includes innovation-based adaptive estimation (IAE) [17], multiple model adaptive estimation (MMAE) [18] and adaptive fading Kalman filter (AFKF) [19]. On the other hand, some methods focus on nonlinear systems, such as extended Kalman filter (EKF) [20], unscented Kalman filter (UKF) [21], noise-robust filter [22] and other estimation methods [23,24]. The two categories of the methods try to describe and represent the system features with the variable approximation. The filters will be efficient if the alternative expression of the model and parameter is similar to the system dynamic characteristic.

The methods above mainly extract and represent the system characteristics with the existing information. While the characteristic extraction is the specialty of machine learning, the artificial neural network (ANN) has been introduced in noise filtering and state estimation. The leading solution is the distributed mode in which Kalman filter and ANN are applied separately in sequential order [25–27]. In the mode, the neural network mainly preprocesses or reprocesses the data before or after the filtering process. However, the inner relation in the Kalman filter has not been explored deeply with ANN. Then it becomes an issue on how to extract the relationship of parameters in the Kalman filter and optimize the filtering results with the limited existing information.

Because of the advantages in Kalman filter and the neural network, a new neuron-based Kalman filter is built in this paper. It mainly enhances the filtering process with the existing information. The potential numerical relation of the intermediate variables in the Kalman filter is explored with the feature extraction and nonlinear fitting ability of the neural network. In the paper, neurocomputing is integrated with the inner components of the Kalman filter. The nonlinear autoregressive model is introduced and constructed to predict and modify the critical intermediate variables in the Kalman filter. The simulation and practical experiments have verified the precision and feasibility of the proposed filter.

This paper is organized as follows: Section 2 introduces the underlying theory and related works on noise filtering. Section 3 presents the main proposed filter with the framework and network design. The simulation and experiment are designed and conducted in Section 4. The results and work are discussed in Sections 5 and 6 finally concludes the paper.

2. Related Work

As the typical filtering method, the Kalman filter is selected as the basic framework in this paper. The basic theory and developments of the Kalman filter are introduced firstly. Then the related work is presented on the integration of the filter and neural networks.

2.1. Kalman Filter and Its Improvement

Because of its clearness and convenience in computer calculation, the Kalman filter has been the classical method in the filtering and estimation of Gaussian stochastic systems [28,29]. It is applied widely in target tracking [30], integrated navigation [31], communication signal processing [32], etc. Kalman filter introduces the state space description in the time domain, in which the estimated signal is set as the output of the stochastic linear system in the action of white noise. Kalman filter is appropriate for the stationary process and the non-stationary Markov sequence.

For the detailed analysis in the paper, the main algorithm of the Kalman filter is presented here. The discrete model can be expressed as:

$$x(k+1) = A(k)x(k) + w(k) \quad (1)$$

$$z(k) = C(k)x(k) + v(k) \quad (2)$$

where $x(k)$ is the to-be-estimated variable or state variable, $z(k)$ is the measurement value from sensors, A is the state transition matrix or process matrix, C is the measurement matrix, $w(k)$ is the process noise, $v(k)$ is the measurement noise. The concrete Kalman filter algorithm is shown as follows:

(1) State estimation updating:

$$\hat{x}(k|k) = \hat{x}(k|k-1) + K(k)[z(k) - C(k)\hat{x}(k|k-1)] \quad (3)$$

(2) One step forward prediction:

$$\hat{x}(k|k-1) = A(k-1)\hat{x}(k-1|k-1) \quad (4)$$

(3) Filtering gain calculation:

$$K(k) = P(k|k-1)C^T(k)[C^T(k)P(k|k-1)C(k) + R(k)]^{-1} \quad (5)$$

(4) The variance of the state estimation calculation:

$$P(k|k-1) = A(k-1)P(k-1|k-1)A^T(k-1) + Q(k-1) \quad (6)$$

$$P(k|k) = [I - K(k)C(k)]P(k|k-1) \quad (7)$$

where $\hat{x}(k|k)$ is the posterior estimation, $\hat{x}(k|k-1)$ is the prior estimation which is also called the prediction, K is the filtering gain, P is the variance of the state estimation, Q is the variance of the process noise, R is the variance of the measurement noise.

There are assumptions in Kalman filter, namely that the process and measurement noises are standard Gaussian noises, and their covariance matrixes are all known. The assumptions deviate from the real systems. Then many studies have been carried out to improve Kalman filter from different solutions.

Some improvements were proposed for the nonlinear system, and the typical methods include EKF [20] and UKF [21]. In EKF, the Taylor expansion of the nonlinear function is truncated with the first-order linearization, and other higher-order terms are ignored. Then the nonlinear problem can be transformed into the linearity, which is suitable for the Kalman filter. In UKF, the prediction measurement values are represented with the sampling points, and the unscented transformation is used to deal with the nonlinear transfer of mean and covariance. EKF and UKF have been improved, as well as the integration with other methods [33–35].

Aside from nonlinear system methods, AKF methods have been studied to solve problems where mainly the settled and experiential parameters are given. The representative IAE [17], MMAE [18], and AFKF [19] are proposed based on the thought that the model parameter and noise statistics are modified with the observation and judgment during the filtering process. From a literature search, it was seen that some improvements in AKF [36–38] were presented recently. In the latest work [38], the colored noise is analyzed with the adaptive parameter. The second-order adaptive statistical model and Yule-Walker algorithm are used to recognize and filter the noises. The work is one of the latest representative improvements of AKF, and it can be set as a contrast in the experimental research.

The two categories of methods above, nonlinear and adaptive filters, mainly improve the filtering performance from the approximate system modeling and parameter adjustment. They are conducted based on inherent mathematics and statistic derivation. They provide an effective solution to promote the Kalman filter in the system mechanism analysis idea. The filtering and prediction are based on the mathematical models by assuming that the model parameters are known or estimated using some parameter identification methods, including the iterative algorithms [39–41], the particle-based algorithms [42–44] and the recursive algorithms [45–48].

2.2. Filter with Neural Network

The methods in Section 2.1 improve the Kalman filter by modifying the system model and parameters based on the mathematic mechanism. The idea can be carried out with another data-driven solution. For the filtering parameter adjustment, the core task is to find and express the relation between

parameters and process data, which meets the ability of neural networks. ANN has caused great concerns again with the trends of deep learning and artificial intelligence. ANN can fit the nonlinear model with excellent performance. It can solve the nonlinear and time-varying problems without a concrete internal mechanism model. For the problematic modeling of process and noise in the Kalman filter, ANN can be considered as a helpful tool to reconstitute the unknown elements in the filter. Scholars have made some efforts to explore the integrations of ANN and Kalman filter. The related research can be divided into two categories, including the distributed and crossed integration.

2.2.1. Distributed Integration of Kalman filter and ANN

For the distributed integration, Kalman filter and ANN are applied separately in sequential order. Liu et al. [25] smoothed the measurement value with the Kalman filter, and the filtered results were set as the input of the backpropagation neural network (BPNN). Hu et al. [49] estimated the target location with Kalman filter and the estimation was imported into BPNN to classify the targets. Liu et al. [50] utilized Kalman filter and fuzzy neural network (FNN) in a multi-source data fusion framework of an adaptive control system, in which data was processed firstly with Kalman filter, and the filtered results were set as the input of FNN. Others [26,27,51] used a Kalman filter and ANN in reverse order, in which ANN is constructed before the Kalman filter. Leandro et al. [26,27] built up BPNN to predict a variable, which is an important state variable in the Kalman filter. Cui et al. [51] proposed a radial basis function neural network (RBF) to train the GPS signals, and the RBF output is the input of adaptive Kalman filter, aiming at improving the processing precision.

2.2.2. Crossed Integration of Kalman Filter and ANN

Different from the methods above, Kalman filter and ANN are combined in a tight pattern, in which ANN is applied during the internal filtering procedure. Shang et al. [52] predicted the model error in the filter with FNN, and the error level was considered to confirm the measurement noise covariance, which was set as 0 or infinity. Li et al. [53] thought that the gain of EKF was usually modified with the erroneous measurement, which reduced the gain precision. They used BPNN to train the gain with the input of measurement, estimation, and error, and then the precision was increased. Deep neural networks [54] have been studied recently. Pei et al. [55] combined a deep neural network with the Kalman filter in the emotion recognition of the image and audio. The features extracted by the deep neural network were input into the switching Kalman filter to obtain the final estimation results.

In the research reported in the literature, more works are conducted in the first separately distributed mode, in which Kalman filter and ANN process the data respectively. The mode does not adjust the inner parameters of the Kalman filter. The works of tightly crossed integration are relatively few. They can be improved with the relation exploration in filter parameters with ANN. Besides, the category and structure of the neural network can be studied to meet the demand for filtering calculation procedures.

3. Neuron-Based Kalman Filter

3.1. Framework of Neuron-Based Kalman Filter

Kalman filter provides a feasible framework to filter the noises and estimate the system state. The components of the Kalman filter can be divided into two categories, namely the models and intermediate variables. The models describe the system dynamic and the measurement process, including the system process equation and measurement equation. The intermediate variables influence the filtering results seriously, but they are difficult to determine in practice. In this paper, the effect of the intermediate variables on the filtering is explored with neural networks. The neurons are integrated into the Kalman filter, and the neurons can help to optimize the filtering process with the limited existing information.

The main influence factors of Kalman filter include the process equation, process noise, and measurement noise, expressed as the matrix A , Q and R . Notably, the noise variances are the critical intermediate variables that affect the estimation results. The effect of noise variances is expressed in the filtering gain K , and the filtering gain determines the estimation result as an important weight. In the optimizing thought with the neurons, the influence relation of the filtering results and the variables should be explored. Then the framework of the integrated Kalman filter is designed firstly, shown in Figure 1, and the concrete design ideas will be interpreted later.

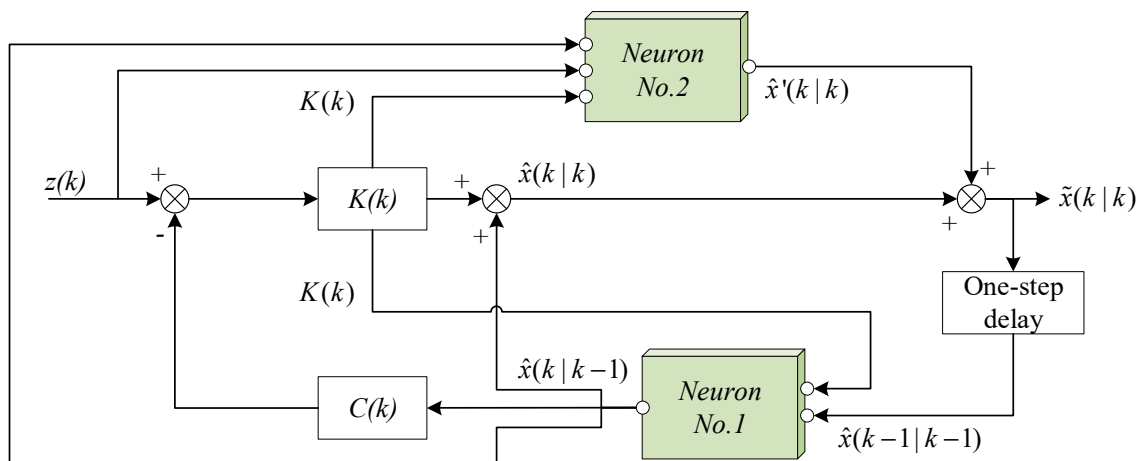


Figure 1. Framework structure of the neuron-based Kalman filter.

Considering the prediction process of the Kalman filter in Equation (4), the estimation result is affected by the precision of the process equation. The process equation describes the system change along the time, but the common simplified equation is difficult to model the actual system. In the view of data-driven, the core of the process equation is the time series relation of the system change. And the time series can be modeled well with neural computing. Then a neuro unit is introduced to model the process in a black-box thought. The first neuro unit can be expressed as:

$$\hat{x}(k) = f_1(K(k), K(k-1), K(k-2), \dots, \hat{x}(k-1), \hat{x}(k-2), \dots) \quad (8)$$

The inputs in the first neuro unit are the filtering gain K series and the state estimation value \hat{x} at the previous time points, represented with $k-1, k-2, \dots$. The output is the estimation \hat{x} at the time k . The key to the model is the fitting function f_1 which aims at the excavation of the time series features in the estimated variables and intermediate filtering variables. As a vital variable, the filtering gain K is obtained from the variance of process and measurement noise Q and R , and it can represent the noise features to some extent. Therefore, the filtering gain is set as an input to transmit the noise features to the system process model. The output estimation value is related to the state at the previous time and the filtering gain. Then the new estimation can be regarded as a more accurate predictive value of the system, and it replaces the initial prediction value in Equation (4) to continue the computing process of the Kalman filter.

For neural computing, it needs training with the existing data. In the training of the first neuro unit, the filtering process data and the final estimation result are collected as the training set. In detail, the series data of the filtering gain from time step $(k-m)$ to k and the estimation value from $(k-m)$ to $(k-1)$ are set as the training input data, where m is the prediction length set in the neural computing. The estimation value at k is set as the training output. In fact, the previous one-step prediction value is optimized with the final estimation value in the network. The fitting function f_1 can be obtained with the training data and the learning algorithm which will be discussed in Section 3.2.

Considering the final estimation process of Kalman filter in Equation (3), the estimation result is determined by two parts, of which one is the prediction value, and the other one is the measurement

residual error. The second neuro unit is built to discover the mapping relation between the two parts and the final estimation result, expressed as:

$$\hat{x}'(k) = f_2(K(k), K(k-1), \dots, z(k), z(k-1), \dots, \hat{x}(k), \hat{x}(k-1), \dots) \quad (9)$$

The inputs of the second neuro unit are the measurement value z , the prediction value \hat{x} and the filtering gain K at the previous time points, represented with $k-1, k-2, \dots$. The output is the final estimation value \hat{x}' at the time k . The final estimation synthesizes the measurement and the filtering intermediate variables such as the noise variance matrix (reflected by the filtering gain) and the prediction variables. The function f_2 is trained to realize the synthetization. In the training, the values of K, z, \hat{x} from $(k-m)$ to k are set as the input, and the final estimation value \hat{x}' at k is set as the output.

It can be considered that the estimation via the neuro units is an effective supplement and amendment of the estimation in Kalman filter. Then the new final estimation value can be obtained by synthesizing the two estimation values from the neural computing and Kalman filter:

$$\hat{x}(k|k) = (1 - \alpha)\hat{x}(k|k) + \alpha\hat{x}'(k|k) \quad (10)$$

where $\hat{x}(k|k)$ is from Kalman filter, $\hat{x}'(k|k)$ is from the neuro unit. $(1 - \alpha)$ and α are the weights of the two estimation values. α is determined by the validation error of the neuro unit, and:

$$\alpha = \frac{n}{\sum_{i=1}^n |(d_i^v - d_i) / d_i|} \quad (11)$$

where d is the validation set during the neuro unit training, d^v is the output of the neuro unit for the validation set, n is the number of data in the validation set.

3.2. Neuro Units Based on Nonlinear Autoregressive Model

In the framework of the neuron-based Kalman filter, the critical components are the neuro units which analyze the intermediate variables to support the final filtering result. Referring to the demand analysis of the two units, the two functions in Equations (8) and (9) should be able to fit the nonlinear relation in multiple variables. Moreover, they should excavate the time-series features in the data. With the two aspects of the demands, the nonlinear autoregressive model with exogenous input (NARX) can be the appropriate solution [56,57]. NARX derives from the time series autoregressive analysis, and it is effective in the reconstitution of the nonlinear systems. The availability of NARX has been proved by various applications [58–60].

NARX belongs to the recurrent neural network. It has a learning efficiency with the better gradient descent. The nonlinear relation between the inputs and outputs in NARX can be expressed as follows:

$$y(t+1) = \phi(y(t), y(t-1), \dots, y(t-n), u(t+1), u(t), \dots, u(t-m)) \quad (12)$$

where $y(t+1)$ is the output to be predicted, $y(t)$ to $y(t-n)$ are the historical outputs, $u(t+1)$ to $u(t-m)$ are the related inputs which last to the current moment. ϕ represents the nonlinear relation between inputs and outputs, and it also represents the structure of NARX, shown in Figure 2.

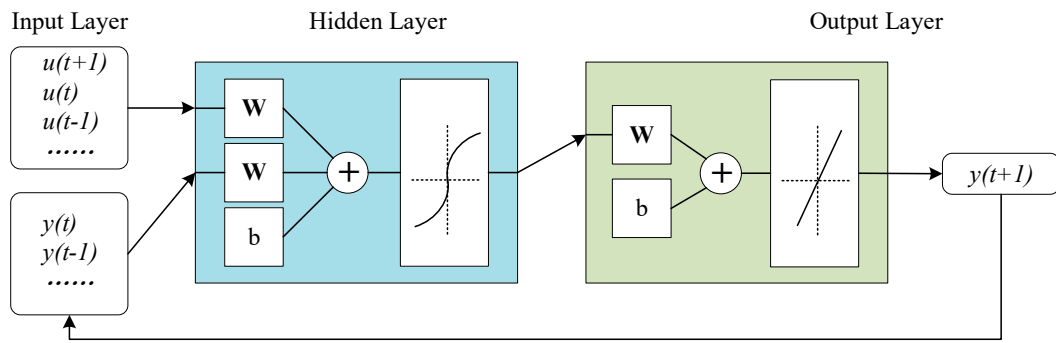


Figure 2. Structure of NARX.

NARX consists of the input layer, hidden layer, and output layer. The transfer function of NARX is similar to the backpropagation neural network, and a one-hidden-layer network is shown as follows:

$$f(\cdot) = g\left(\sum \omega_h h(\cdot)\right) \tag{13}$$

$$h(\cdot) = r\left(\sum \omega_i u_i\right) \tag{14}$$

where g and r are the activation functions of the output, ω_h is the node weight in the hidden layer, $h(\cdot)$ is the activation function of the hidden layer, ω_i is the weight of all inputs. The vector form of ω_h and ω_i are represented with W in Figure 2.

Based on the primary NARX structure, the two neuro units in the neuron-based Kalman filter can be designed concretely, which are shown in Figure 3.

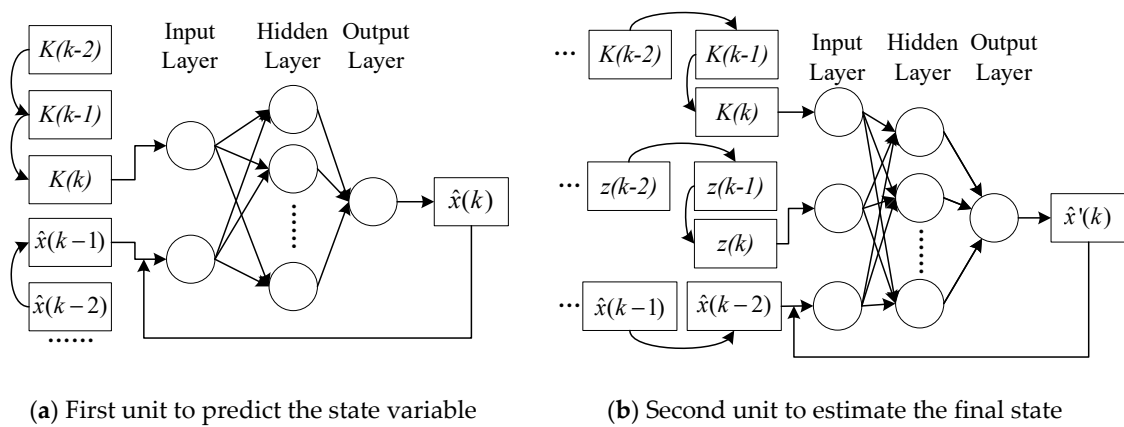


Figure 3. Concrete structures of the two neuro units in the proposed Kalman filter.

For the neuron units in Figure 3, the concrete scale can be determined with the traditional empirical mode in the shallow neural network. The number of the hidden layer is set as one according to the number of input and output variables. The number of hidden nodes can be determined with the equations, such as $n = \log_2 p$, $n = \sqrt{p+q} + a$, where n is the number of hidden nodes, p is the number of input nodes, q is the number of output nodes, and a is a constant between 1 and 10. Besides, the number of hidden nodes can be adjusted in the network training, following the network performance.

Based on the static construction of the neural network, the appropriate training method should be selected to obtain favorable dynamic performance. The gradient descent method is the core solution in neural network training. Some improved algorithms have been proposed. Levenberg-

Marquardt (L–M) [61,62] is a rapid training method that combines the basic gradient descent method and Gauss-Newton method. Its error target function is:

$$E(w) = \frac{1}{2} \sum_{i=1}^p \|Y_i - Y_i'\|^2 = \frac{1}{2} \sum_{i=1}^p e_i^2(w) \quad (15)$$

where Y_i is the expected output, Y_i' is the actual output, $e_i(w)$ is the error, p is the number of samples, w is the vector consisting of network weights and threshold values.

The k -th iterative vector of weights and threshold values is w^k , and the new vector is:

$$w^{k+1} = w^k + \Delta w \quad (16)$$

and the increment in L–M is calculated as:

$$\Delta w = [J^T(w)J(w) + \mu I]^{-1} J^T(w)e(w) \quad (17)$$

where I is the unit matrix, μ is the learning rate, $J(w)$ is the Jacobian matrix, and:

$$J(w) = \begin{bmatrix} \frac{\partial e_1(w)}{\partial w_1} & \frac{\partial e_1(w)}{\partial w_1} & \dots & \frac{\partial e_1(w)}{\partial w_1} \\ \frac{\partial e_1(w)}{\partial w_1} & \frac{\partial e_1(w)}{\partial w_1} & \dots & \frac{\partial e_1(w)}{\partial w_1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_1(w)}{\partial w_1} & \frac{\partial e_1(w)}{\partial w_1} & \dots & \frac{\partial e_1(w)}{\partial w_1} \end{bmatrix} \quad (18)$$

With the training method, the neuro units can be built with the anticipative functions. Then the improved Kalman filter can be established with the functional neuro units.

3.3. Adaptive Filtering Algorithm

In the framework of neuron-based Kalman filter, two neuron units are introduced into the basic consistent of Kalman filter. The input, output, and inner structure of the neuro unit are designed to improve the filtering. Finally, the adaptive filtering algorithm based on the improved Kalman filter is proposed here, in which the neuro units are trained firstly and applied to the filter. The flow of the algorithm is presented in Figure 4.

As shown in Figure 4, the algorithm consists of two parts, namely the training process on the left and the filtering process on the right. The concrete flow of the algorithm is as follows:

(A) Training process

- (1) The system and measurement equations are established according to the object. The parameters in the Kalman filter can be initialized with empirical values.
- (2) The primary calculation of the Kalman filter is conducted iteratively following Equations (1)–(7). The measurement vectors are imported into the filter along with time. The intermediate and final values are recorded, including the one-step prediction value, Kalman gain, measurement, estimation result, etc. The recorded values are all labeled with a time stamp. Meanwhile, the iterative steps should be no less than about 150 for the following neuro unit training. The number of sample steps may be adjusted according to the complexity of signals.
- (3) With the filtering values in step 2, they are marked with the step number to form the time series sets. Then the prediction value and filter gain are imported into the first neuro unit. The prediction value, filter gain, and measurement are imported into the second neuro unit. The estimation result is set as the reference output of the two units.

- (4) The neuro units are trained with the learning method L–M in Section 3.2. The trained neuro units are obtained when the preset iteration conditions are met, including the numbers of iteration or the convergence error.
- (B) Filtering process with trained neuro units
- (5) Based on the model equations and the initialized parameters in Kalman filter, the initial variable and filter gain are imported into the first neuro unit, and the prediction value is outputted and set as the basis of prediction error.
- (6) The filter gain is updated and used to calculate the estimation value with the measurement. Meanwhile, the prediction value, filter gain, and measurement are imported into the second neuro unit to obtain another estimation value.
- (7) The two estimation values are fused following Equation (10).
- (8) The step moves forward to conduct steps (5)–(7) iteratively. In the iteration, the measurement vectors are calculated along with time.

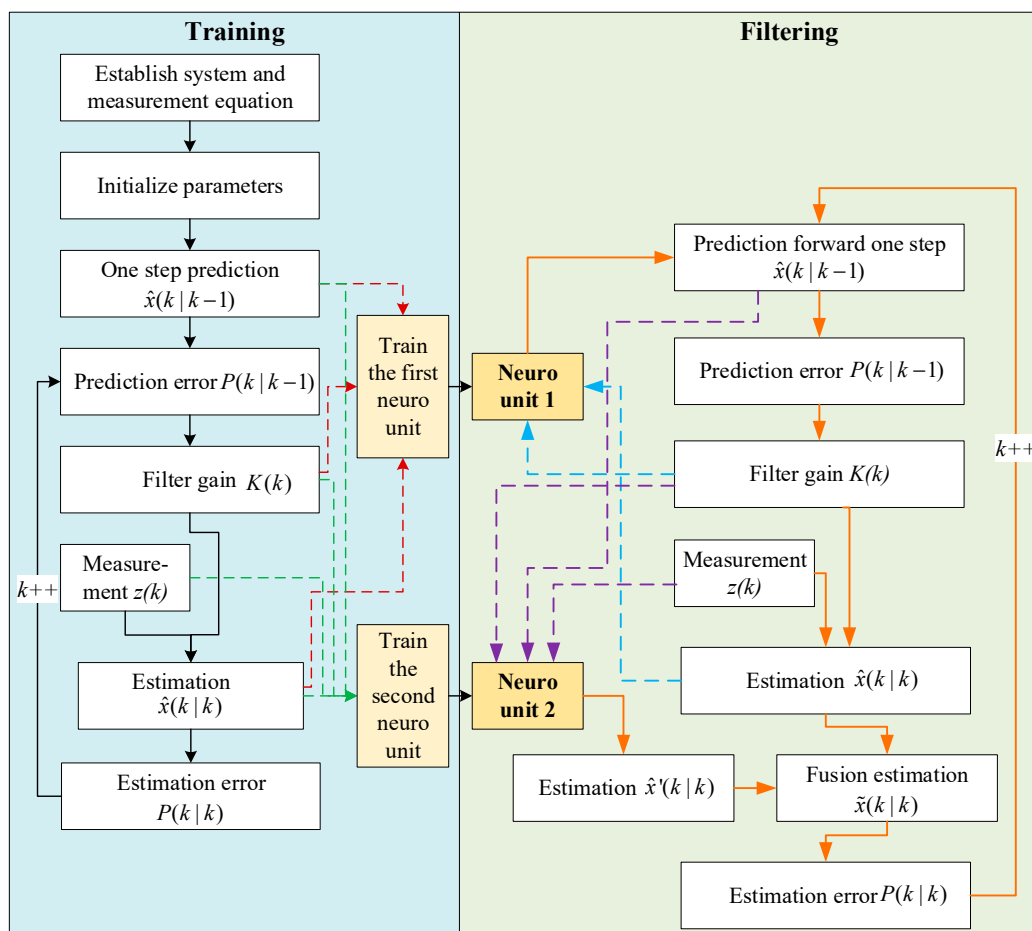


Figure 4. Algorithm flow of the adaptive filtering with the neuron-based Kalman filter.

4. Experiment and Result

The simulation and practice experiments are conducted to test the filter proposed. In the simulation, different noises are generated to simulate the complex noises in the sensors. In the experiment, the wheeled robot path is measured with low-cost GPS. All the computing runs with MATLAB 2017a on a PC equipped with an Intel Core i5-6200U CPU@2.30 GHz and 8 GB RAM. The experiment setting and results are presented in this section.

4.1. Simulation and Result

The common noises in sensors are the white noise and color noise. The signals with the two kinds of noises are generated in the simulation. The two sets of the signals are:

$$x_1(k) = \sigma(k)f(k) \quad (19)$$

$$x_2(k) = G(z^{-1})f(k) \quad (20)$$

where $f(k)$ is the Gaussian white noise. $\sigma(k)$ is the standard deviation of $f(k)$, and $\sigma(k) = (L + k)/L$, L is the number of signal samples, and k is the sample number. $G(z^{-1})$ is the transfer function of a system which can be second order or third order to simulate the noise change process.

The first set is the approximately linear noises, and the second one is the sinusoidal noises, corresponding to the white noise and color noise, respectively. In the simulation, the sampling interval is 0.02 s. The numbers of signal samples are all 2000. That is, the sampling time is the 40 s. The simulation signals are shown in Figure 5, in which the true values of x_1 and x_2 are 0.

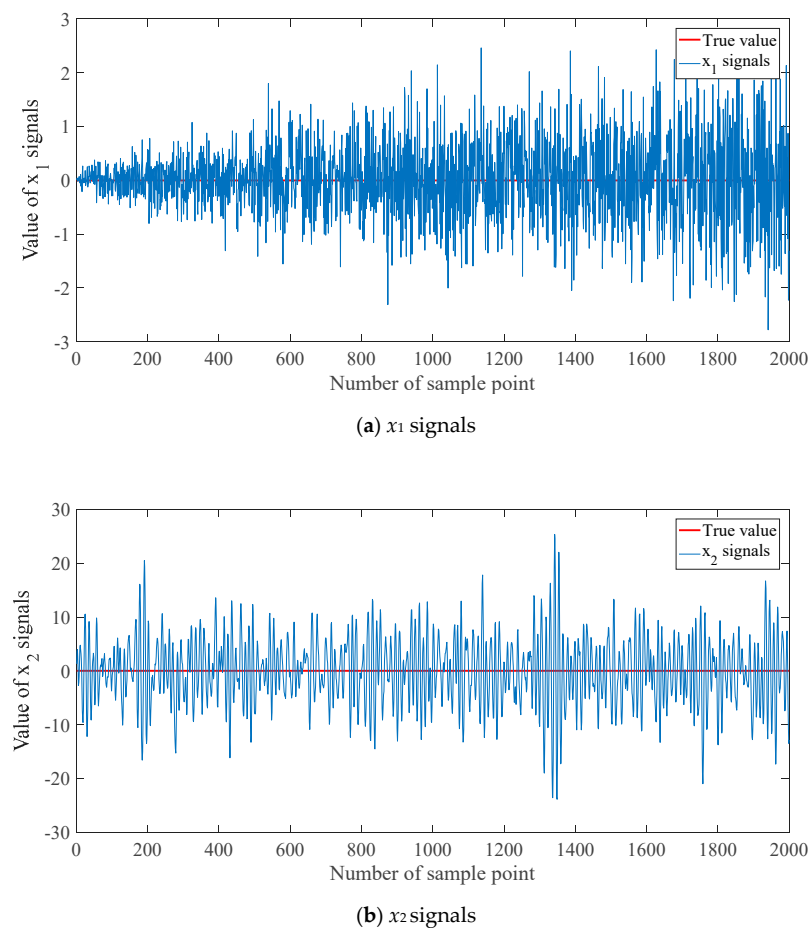


Figure 5. Simulation signals with different noises.

In the filtering of the simulation signals, the system model is established with the classical Jerk model, which also can be replaced with other motion models such as constant velocity, constant acceleration, Singer, interacting multiple model algorithms, and so on. For the Kalman filter, the initial state estimate x_0 and covariance P_0 are assumed to be $x_0 = [0 \ 0 \ 0]^T$ and $P_0 = 1000 \cdot \text{eye}(4)$.

Because the neuro unit needs the training, the first 70% of the data are set as the training data, and the rest is used to test the filtering result. In the setting of the two neuro units, the number of

hidden nodes are set as 3 and 6, respectively. Other settings are also tested to obtain the optimal performance. The training results of NARX are shown in Figure 6.

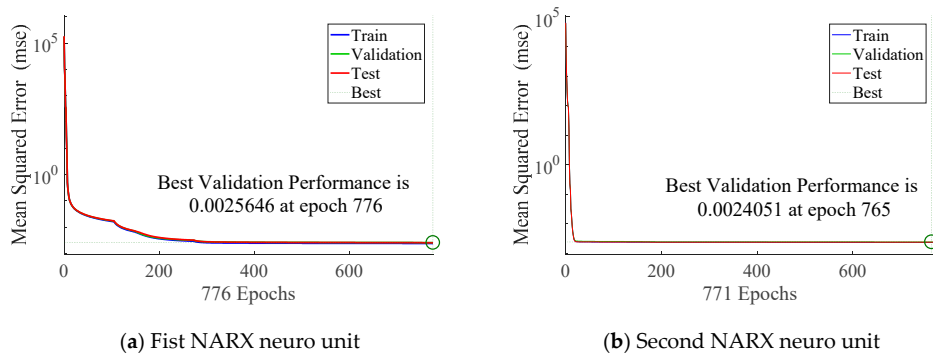
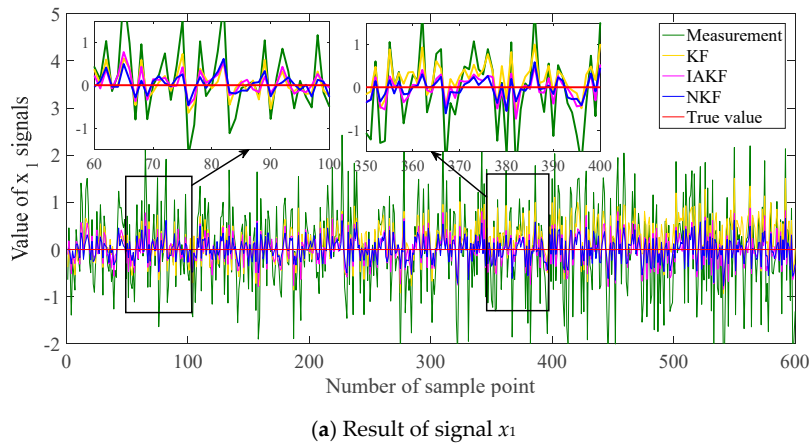
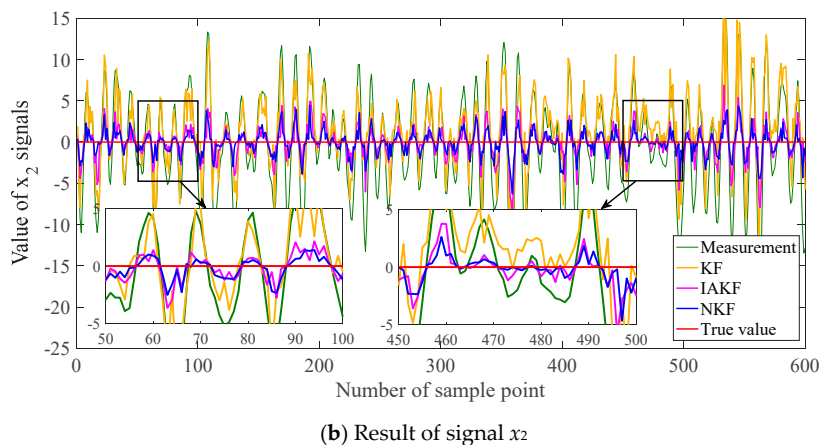


Figure 6. Training results of NARX in integrated filter.

Based on the trained neuro units, the data are imported into the proposed filter to estimate the variable values. For verifying the estimation performance, the traditional Kalman filter is set as a contrast, abbreviated as KF. Moreover, the proposed filter can be regarded as a solution to adaptive filtering. Then one of the latest improvements of AKF in [38] is also set as the contrast, abbreviated as IAKF. The proposed filter in this paper is abbreviated as NKF. The filtering results are shown in Figure 7. For the quantitative evaluation of errors, the mean absolute error (MAE) and root-mean-square error (RMSE) are calculated and listed in Table 1.



(a) Result of signal x_1



(b) Result of signal x_2

Figure 7. Filtering results of simulation signals.

Table 1. Evaluation of filtering errors.

		KF	IAKF	NKF
Signal x_1	MAE	0.3692	0.2550	0.2004
	RMSE	0.4577	0.3170	0.2507
Signal x_2	MAE	3.3379	1.2678	1.0294
	RMSE	4.4763	1.8295	1.4429

For the first set of signals which are of the white noise, the results of the three methods are relatively similar in the curve graph. NKF performs better slightly than the others. The trend is evident in the error evaluation criteria. MAE represents the mean level of errors. MAE of NKF declines 45.72% of KF and 21.41% of IAKF. RMSE shows the fluctuation degree of errors. RMSE of NKF decreases 45.23% of KF and 20.91% of IAKF.

For the second set, the filtering results show the distinguishable trends. The results of KF fluctuate sharply and become diverging in the latter period. IAKF and NKF can trace the signals more closely, and NKF is more effective in the intuitionistic graph. MAE of NKF has been reduced by 69.16% of KF, and 18.80% of IAKF. The decreasing percentage of RMSE reaches 67.77% and 21.13% for NKF to KF and IAKF. The error reduction of the second set is larger than the first set.

4.2. Practical Experiment and Result

Except for the simulation, a practical experiment is also conducted to verify the proposed method. A trajectory of the wheeled robot (shown in Figure 8) is measured on the playground, and the presupposed trajectory is presented in Figure 9. The robot started from the top right corner and ended at the same point. A low-cost GPS receiver is used to obtain the location information, including the longitude and latitude. The relative coordinates are transformed from the longitude and latitude:

$$d = 111.12 \cdot \cos \frac{1}{\sin \phi_{t-1} \sin \phi_t + \cos \phi_{t-1} \cos \phi_t (\lambda_t - \lambda_{t-1})} \quad (21)$$

where d is the displacement, ϕ is the latitude and λ is the longitude. The displacement can be decomposed into the coordinates on a plane. The measurements and true trajectory in the relative coordinates are shown in Figure 10.

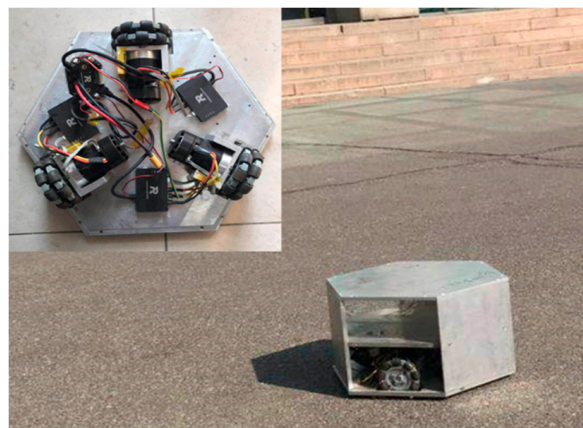


Figure 8. Wheeled robot used to measure the trajectory, and the robot is developed and assembled by laboratory of system engineering in Beijing Institute of Technology, Beijing, China.



Figure 9. Presupposed trajectory in the practical experiment.

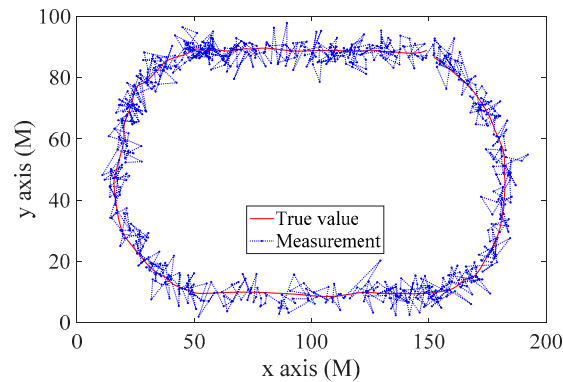
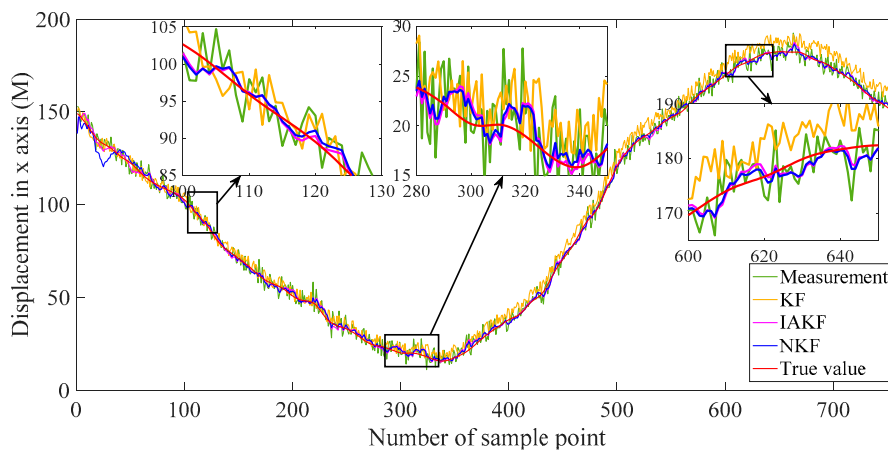


Figure 10. Relative coordinates transformed from the practical trajectory measurements.

In this part, the data of the whole trajectory is filtered firstly. Then a segment of the trajectory in another measurement is tested again.

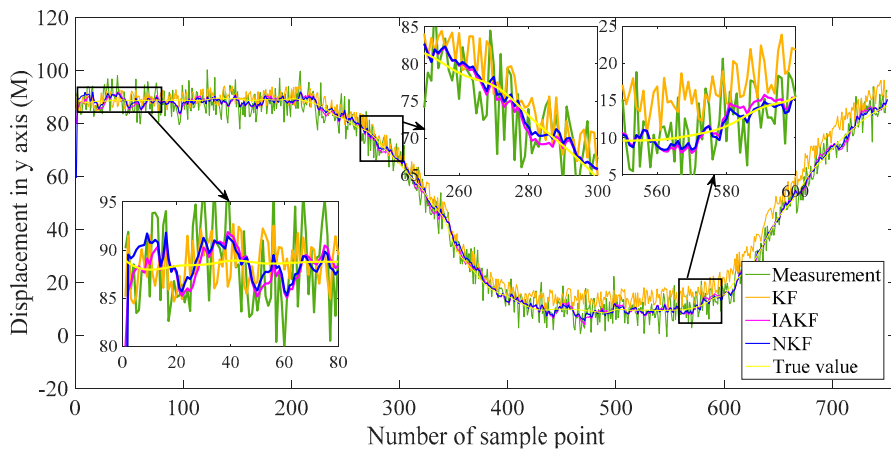
4.2.1. Result of the Whole Trajectory

Similar to the simulation, the traditional Kalman filter and improvement of AKF in [38] are set as the contrast methods. The filtering results are shown in Figure 11, including the distance in the x-axis, y-axis, and x-y plane. The absolute errors are shown in Figure 12, and the evaluation indexes are in Table 2.

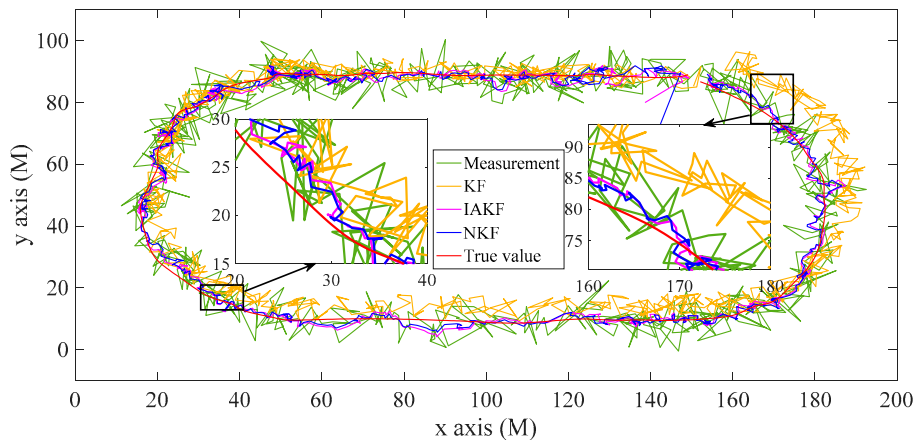


(a) Displacement in x-axis

Figure 11. Cont.

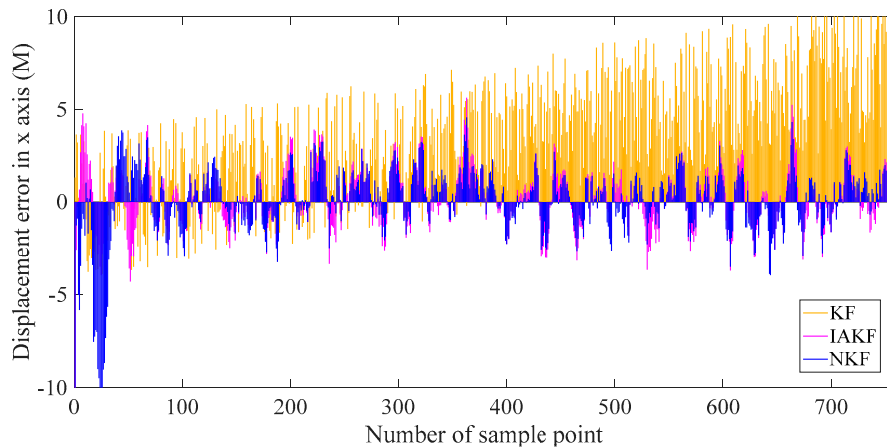


(b) Displacement in y-axis



(c) Whole trajectory in the x-y plane

Figure 11. Filtering results of the whole trajectory.



(a) Absolute error in x-axis

Figure 12. Cont.

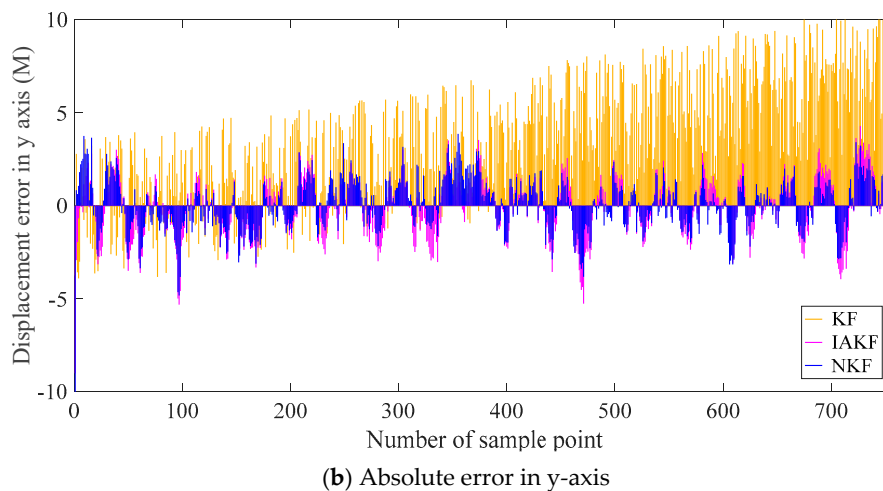


Figure 12. Absolute errors of displacement in x and y axes.

The results in Figure 11 show intuitively the small differences in the filtering results, and it is because the difference is in the order of magnitudes less than 1 m. The differences are more evident in the absolute errors in Figure 12. The general trends of the filtering results with the three methods are similar to the pattern of the second set signal in the simulation. It is because that the noises in the real sensors mainly are the color noises instead of the white noises. The basic Kalman filter performs badly in the practical system. Its results are unsteady and diverging along with the time. NKF perform better than others in various time periods besides the beginning. The neuro unit based on NARX reaches stable performance after the drastic fluctuation, which usually occurs at the beginning. Therefore, NKF is superior to the Kalman filter on the whole without the initial period. The performance can be analyzed quantitatively with the error criteria in Table 2.

Table 2. Evaluation of filtering errors (whole trajectory).

		KF	IAKF	NKF
x axis	MAE	3.8730	1.3117	1.3048
	RMSE	4.6732	1.9079	1.6594
y axis	MAE	3.7327	1.3184	1.1651
	RMSE	4.5560	1.7578	1.6430

The general trend is consistent in the x -axis and y -axis for the three indexes. There is a conspicuous promotion in MAE for NKF and the RMSE of NKF declines about 6% with the Kalman filter and 8% with IAKF.

4.2.2. Result of Segment in the Trajectory

To test the proposed filter with more data, a segment of the whole trajectory is selected from one of the multiple measurements, which is not in the same measurement with the whole trajectory above.

The data from 180 s to 450 s are selected, including the displacements in x -axis and y -axis. The contrast methods are the same as the experiment above. The filtering results are shown in Figure 13, and the evaluation indexes of errors are listed in Table 3.

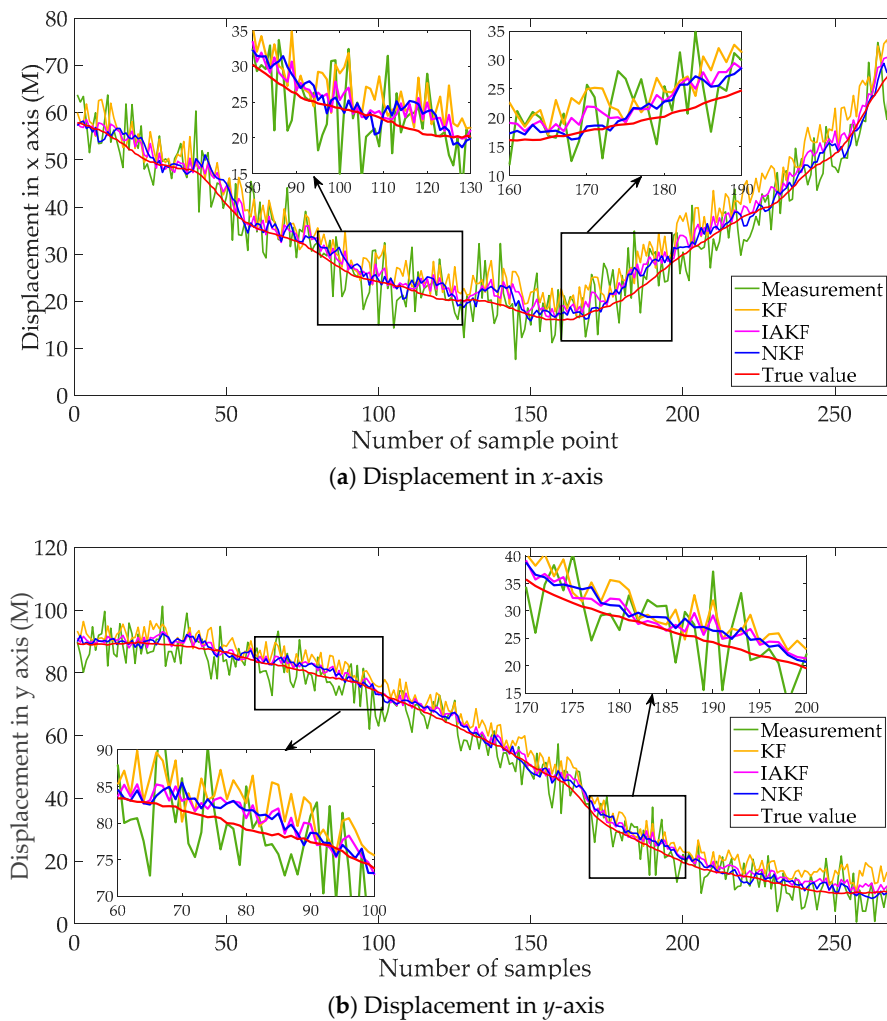


Figure 13. Filtering results of the selected segment of the trajectory.

Table 3. Evaluation of filtering errors (segment of the trajectory).

		KF	IAKF	NKF
x axis	MAE	4.0157	2.0905	1.7159
	RMSE	4.7610	2.5017	2.0879
y axis	MAE	3.8769	1.7897	1.5230
	RMSE	4.5707	2.1330	1.8024

It can be found from Figure 13 that the general trend is similar to the results of the whole trajectory in Figure 11. The results of NKF and IAKF are intuitively approximately the same. From careful and detailed identification in the magnifying subfigure, the proposed NKF can filter the noises closer to the true value than other methods. In the results of the x -axis and y -axis, the basic KF performs badly with the obvious deviation.

The filtering performance can be evaluated more accurately with Table 3. For results in the x -axis, MAE of NKF is 32.73% of KF, 82.01% of IAKF. RMSE of NKF is similar to IAKF, but relatively lower than KF. The trend of results in the y -axis is consistent with the ones in x -axis. Compared with the error indicators in Table 2, the errors of the segment are larger than the whole trajectory, which may be due to the fewer data to train the neural networks.

5. Discussion

In this paper, a novel Kalman filter is designed by introducing neural computing. Simulations and experiments are carried out, and the results are presented and described briefly. Following the results, the methods are discussed in this section.

From the filtering results of simulation signals and practical measurements, it can be proved that the proposed filter can eliminate the noises to the anticipated degree. It performs distinctly better than the traditional Kalman filter, especially for complex noises. Besides, the proposed filter can achieve the latest improvements of AKF. The core thought of the proposed filter is to obtain more knowledge from the existing limited data during the filtering procedure. The process variables in the filtering are reutilized with the neural units, while the reutilization in AKF [36–38] is conducted with statistical methods. Therefore, the proposed filter can be regarded as a new exploration of parameter adjusting, which is similar to the essential thought of AKF.

In the proposed filter, the neuro unit is built based on the nonlinear autoregressive model. The neuro unit specializes in the nonlinear time-series feature extraction with a small-scale structure. Although many more networks have been proposed, it should be conservative in the selection of networks. The complex network may destroy the efficient and straightforward features of the Kalman filter. Besides, the complex network may be not suitable for the terminal applications without the high-performance processor.

Except for the intuitive estimation results, the computational complexity can be analyzed for the proposed and contrast method. According to the basic evaluation method of computational complexity, the complexity of KF, IAKF, and NKF is $O(n^2)$, $O(n^2)$ and $O(n^3)$, respectively, where n is the number of state variables. The complexity of NKF increases because dual matrix multiplication is introduced by neuro units. The operation time is also recorded in the experiment, shown in Table 4.

Table 4. Operation time of different methods in simulation and experiment (time unit: s).

	Simulation		Practical Experiment (Whole Trajectory)	
	Signal x_1	Signal x_2	x Axis	y Axis
KF	1.23	1.45	2.15	2.09
IAKF	1.37	1.73	2.32	2.43
NKF	1.27	1.79	2.41	2.24

For computing time, the methods distinct slightly, although the complexity of NKF is higher. However, an important fact that cannot be ignored is the training of the neuro units. The time above is the test procedure, while NKF needs prior training. The training time is between 3 s and 7 s, according to the preset convergence conditions. In this paper, the training requires historical data in the offline mode. The filtering is conducted after the training, which reduces the real-time performance. It is the challenge how to realize online learning along with the filtering process, which can be studied in the future.

The neurons in the proposed method work well from the experiment results. Although a good filtering performance has been obtained, the inherent mechanism of the proposed method is actually not completely clear. Hence the theoretical analysis should be conducted, and the effect of the neural network on the filter should be deduced in the view of numerical analysis in the future.

In the proposed method, ANN inspired us to optimize the intermediate factors and calculating process in the Kalman filter with the black-box thought. It mainly solves the problem of modeling and parameter adjusting of the traditional filter. It can be a useful tool in the target tracking, trajectory estimation, and pedestrian navigation, especially in the situations of inexperienced modeling of complex systems and the parameter settings.

6. Conclusions

For the intelligent terminals and objects in the internet of things, it has been the vital task to sense the environment and self-status accurately. An improved Kalman filter is proposed with neural computing for accurate sensing. Kalman filter provides a favorable framework in which the system model can be replaced according to the concrete applications. The neuro unit based on NARX is a powerful tool to examine nonlinear and time-series relations. The proposed filter focuses on the data change features and tries to lower the impact of model analysis. In future work, the stability and rapidity of neural computing should be studied deeply. The neuron-based Kalman filter can develop more fully with smarter and faster online neural computing. Moreover, the theoretical derivation should be carried out to support the neuro-based filter. The proposed method can combine other identification approaches [63–66] to study the modeling and filtering problems of other dynamic time series and stochastic systems with colored noises [67–70], and can be applied to other fields [71–74], such as signal modeling and control systems [75–79] studied in other literature [80].

Author Contributions: Conceptualization, Y.-t.B. and X.-b.J.; methodology, Y.-t.B.; writing—original draft preparation, Y.-t.B.; project administration, X.-y.W.; funding acquisition, Z.-y.Z.; supervision, B.-h.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Key Research and Development Program of China No. 2017YFC1600605, National Natural Science Foundation of China No. 61903008, 61673002, Young Teacher Research Foundation Project of BTBU No. QNJJ2020-26.

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

References

1. Mohd-Yasin, F.; Nagel, D.J.; Korman, C.E. Noise in MEMS. *Meas. Sci. Technol.* **2009**, *21*, 012001. [[CrossRef](#)]
2. Shiau, J.K.; Huang, C.X.; Chang, M.Y. Noise characteristics of MEMS gyro's null drift and temperature compensation. *J. Appl. Sci. Eng.* **2012**, *15*, 239–246.
3. Jiang, Z.; Ni, M.; Lu, Q.; Liu, Z.; Zhao, Y. Wavelet filter: Pure-intensity spatial filters that implement wavelet transforms. *Appl. Opt.* **1996**, *35*, 5758–5760. [[CrossRef](#)] [[PubMed](#)]
4. Yu, P.; Li, Y.; Lin, H.; Wu, N. Seismic random noise removal by delay-compensation time-frequency peak filtering. *J. Geophys. Eng.* **2017**, *14*, 691–697. [[CrossRef](#)]
5. Boudraa, A.O.; Cexus, J.C.; Benramdane, S.; Beghdadi, A. Noise filtering using empirical mode decomposition. In Proceedings of the 9th International Symposium on Signal Processing and Its Applications, Sharjah, UAE, 12–15 February 2007; pp. 1–4.
6. Harvey, A.C. *Forecasting, Structural Time Series Models and the Kalman Filter*; Cambridge University Press: Cambridge, UK, 1990.
7. Wang, Y.J.; Ding, F.; Wu, M.H. Recursive parameter estimation algorithm for multivariate output-error systems. *J. Frankl. Inst.* **2018**, *355*, 5163–5181. [[CrossRef](#)]
8. Ding, F.; Zhang, X.; Xu, L. The innovation algorithms for multivariable state-space models. *Int. J. Adapt. Control Signal Process.* **2019**, *33*, 1601–1608. [[CrossRef](#)]
9. Pan, J.; Jiang, X.; Wan, X.K.; Ding, W. A filtering based multi-innovation extended stochastic gradient algorithm for multivariable control systems. *Int. J. Control. Syst.* **2017**, *15*, 1189–1197. [[CrossRef](#)]
10. Ding, F. Two-stage least squares based iterative estimation algorithm for CARARMA system modeling. *Appl. Math. Model.* **2013**, *37*, 4798–4808. [[CrossRef](#)]
11. Ding, F. Decomposition based fast least squares algorithm for output error systems. *Signal Process.* **2013**, *93*, 1235–1242. [[CrossRef](#)]
12. Li, M.H.; Liu, X.M.; Ding, F. The filtering-based maximum likelihood iterative estimation algorithms for a special class of nonlinear systems with autoregressive moving average noise using the hierarchical identification principle. *Int. J. Adapt. Control Signal Process.* **2019**, *33*, 1189–1211. [[CrossRef](#)]
13. Liu, L.J.; Ding, F.; Xu, L.; Pan, J.; Alsaedi, A.; Hayat, T. Maximum likelihood recursive identification for the multivariate equation-error autoregressive moving average systems using the data filtering. *IEEE Access* **2019**, *7*, 41154–41163. [[CrossRef](#)]

14. Zhang, X.; Ding, F.; Xu, L.; Yang, E. State filtering-based least squares parameter estimation for bilinear systems using the hierarchical identification principle. *IET Control Theory Appl.* **2018**, *12*, 1704–1713. [[CrossRef](#)]
15. Gu, Y.; Ding, F.; Li, J.H. States based iterative parameter estimation for a state space model with multi-state delays using decomposition. *Signal Process.* **2015**, *106*, 294–300. [[CrossRef](#)]
16. Liu, Y.J.; Ding, F.; Shi, Y. An efficient hierarchical identification method for general dual-rate sampled-data systems. *Automatica* **2014**, *50*, 962–970. [[CrossRef](#)]
17. Mehra, R.K. On the identification of variances and adaptive Kalman filtering. *IEEE Trans. Autom. Control.* **1970**, *15*, 175–184. [[CrossRef](#)]
18. Mohamed, A.H.; Schwarz, K.P. Adaptive Kalman filtering for INS/GPS. *J. Geod.* **1999**, *73*, 193–203. [[CrossRef](#)]
19. Rutan, S.C. Adaptive Kalman filtering. *Anal. Chem.* **1991**, *63*, 687–689. [[CrossRef](#)]
20. Julier, S.J.; Uhlmann, J.K. New extension of the Kalman filter to nonlinear systems. In Proceedings of the Signal Processing, Sensor Fusion, and Target Recognition VI, Orlando, FL, USA, 28 July 1997; Volume 3068, pp. 182–193.
21. Xiong, K.; Zhang, H.; Chan, C. Performance evaluation of UKF-based nonlinear filtering. *Automatica* **2006**, *42*, 261–270. [[CrossRef](#)]
22. Chen, B.; Liu, X.; Zhao, H.; Principe, J.C. Maximum correntropy Kalman filter. *Automatica* **2017**, *76*, 70–77. [[CrossRef](#)]
23. Zhang, X.; Xu, L.; Ding, F.; Hayat, T. Combined state and parameter estimation for a bilinear state space system with moving average noise. *J. Frankl. Inst.* **2018**, *355*, 3079–3103. [[CrossRef](#)]
24. Zhang, X.; Ding, F.; Xu, L.; Yang, E. Highly computationally efficient state filter based on the delta operator. *Int. J. Adapt. Control Signal Process.* **2019**, *33*, 875–889. [[CrossRef](#)]
25. Liu, M.; Tian, Z.; Qi, H.; Zhang, C.; Liu, X. Cooperative fusion model based on Kalman-BP neural network for suspended sediment concentration measurement. *J. Basic Sci. Eng.* **2016**, *5*, 970–977. (In Chinese)
26. Leandro, V.M.; Boada, B.L.; Boada, M.J.L.; Gauchía, A.; Díaz, A. A sensor fusion method based on an integrated neural network and Kalman filter for vehicle roll angle estimation. *Sensors* **2016**, *16*, 1400.
27. Leandro, V.M.; Boada, B.L.; Boada, M.J.L.; Gauchía, A.; Díaz, A. Sensor Fusion based on an integrated neural network and probability density function (PDF) dual Kalman filter for on-line estimation of vehicle parameters and states. *Sensors* **2017**, *17*, 987.
28. Sinopoli, B.; Schenato, L.; Franceschetti, M.; Poolla, K.; Jordan, M.I.; Sastry, S.S. Kalman filtering with intermittent observations. *IEEE Trans. Autom. Control* **2004**, *49*, 1453–1464. [[CrossRef](#)]
29. Li, S.E.; Li, G.; Yu, J.; Cheng, B.; Wang, J.; Li, K. Kalman filter-based tracking of moving objects using linear ultrasonic sensor array for road vehicles. *Mech. Syst. Signal Process.* **2018**, *98*, 173–189. [[CrossRef](#)]
30. Khan, M.W.; Salman, N.; Ali, A.; Khan, A.M.; Kemp, A.H. A comparative study of target tracking with Kalman filter, extended Kalman filter and particle filter using received signal strength measurements. In Proceedings of the IEEE International Conference on Emerging Technologies, Peshawar, Pakistan, 19–20 December 2015; pp. 1–6.
31. Chang, L.; Li, K.; Hu, B. Huber's M-estimation-based process uncertainty robust filter for integrated INS/GPS. *IEEE Sens. J.* **2015**, *15*, 3367–3374. [[CrossRef](#)]
32. Durantin, G.; Scannella, S.; Gateau, T.; Delorme, A.; Dehais, F. Processing functional near infrared spectroscopy signal with a Kalman filter to assess working memory during simulated flight. *Front. Hum. Neurosci.* **2016**, *9*, 707. [[CrossRef](#)]
33. Mou, Z.; Sui, L. Improvement of UKF algorithm and robustness study. In Proceedings of the 2009 IEEE International Workshop on Intelligent Systems and Applications, Wuhan, China, 23–24 May 2009; pp. 1–4.
34. Huang, Y.; Zhang, Y.; Li, N.; Chambers, J. Robust Student's t based nonlinear filter and smoother. *IEEE Trans. Aerosp. Electron. Syst.* **2016**, *52*, 2586–2596. [[CrossRef](#)]
35. Zhang, X.; Ding, F.; Yang, E. State estimation for bilinear systems through minimizing the covariance matrix of the state estimation errors. *Int. J. Adapt. Control Signal Process.* **2019**, *33*, 1157–1173. [[CrossRef](#)]
36. Zhou, Q.; Zhang, H.; Wang, Y. A redundant measurement adaptive Kalman filter algorithm. *Acta Aeronaut. Astronaut. Sin.* **2015**, *36*, 1596–1605.
37. Qian, X.; Yong, Y. Fast, accurate, and robust frequency offset estimation based on modified adaptive Kalman filter in coherent optical communication system. *Opt. Eng.* **2017**, *56*, 096109.

38. Yi, S.; Jin, X.; Su, T.; Tang, Z.; Wang, F.; Xiang, N.; Kong, J. Online denoising based on the second-order adaptive statistics model. *Sensors* **2017**, *17*, 1668. [[CrossRef](#)] [[PubMed](#)]
39. Ding, F.; Pan, J.; Alsaedi, A.; Hayat, T. Gradient-based iterative parameter estimation algorithms for dynamical systems from observation data. *Mathematics* **2019**, *7*, 428. [[CrossRef](#)]
40. Ding, F.; Lv, L.; Pan, J.; Wan, X.; Jin, X. Two-stage gradient-based iterative estimation methods for controlled autoregressive systems using the measurement data. *Int. J. Control Autom. Syst.* **2020**, *18*, 1–11. [[CrossRef](#)]
41. Xu, L.; Ding, F. Iterative parameter estimation for signal models based on measured data. *Circuits Syst. Signal Process.* **2018**, *37*, 3046–3069. [[CrossRef](#)]
42. Ding, J.; Chen, J.; Lin, J.X.; Wan, L.J. Particle filtering based parameter estimation for systems with output-error type model structures. *J. Frankl. Inst.* **2019**, *356*, 5521–5540. [[CrossRef](#)]
43. Ding, J.; Chen, J.Z.; Lin, J.X.; Jiang, G.P. Particle filtering-based recursive identification for controlled auto-regressive systems with quantised output. *IET Control Theory Appl.* **2019**, *13*, 2181–2187. [[CrossRef](#)]
44. Ding, F.; Xu, L.; Meng, D.D.; Jin, X.; Alsaedi, A.; Hayate, T. Gradient estimation algorithms for the parameter identification of bilinear systems using the auxiliary model. *J. Comput. Appl. Math.* **2020**, *369*, 112575. [[CrossRef](#)]
45. Cui, T.; Ding, F.; Jin, X.B.; Alsaedi, A.; Hayat, T. Joint multi-innovation recursive extended least squares parameter and state estimation for a class of state-space systems. *Int. J. Control Autom. Syst.* **2020**, *18*, 1–13. [[CrossRef](#)]
46. Ding, F. Coupled-least-squares identification for multivariable systems. *IET Control Theory Appl.* **2013**, *7*, 68–79. [[CrossRef](#)]
47. Xu, L.; Xiong, W.L.; Alsaedi, A.; Hayat, T. Hierarchical parameter estimation for the frequency response based on the dynamical window data. *Int. J. Control Autom. Syst.* **2018**, *16*, 1756–1764. [[CrossRef](#)]
48. Ding, F. Hierarchical multi-innovation stochastic gradient algorithm for Hammerstein nonlinear system modeling. *Appl. Math. Model.* **2013**, *37*, 1694–1704. [[CrossRef](#)]
49. Hu, Y.; Li, L. The application of Kalman filtering-BP neural network in autonomous positioning of end-effector. *J. Beijing Univ. Posts Telecommun.* **2016**, *39*, 110–115. (In Chinese)
50. Liu, J.; Cheng, K.; Zeng, J. A novel multi-sensors fusion framework based on Kalman Filter and neural network for AFS application. *Trans. Inst. Meas. Control* **2015**, *37*, 1049–1059. [[CrossRef](#)]
51. Cui, L.; Gao, S.; Jia, H.; Chu, H.; Jiang, R. Application of neural network aided Kalman filtering to SINS/GPS. *Opt. Precis. Eng.* **2014**, *22*, 1304–1311. (In Chinese)
52. Shang, Y.; Zhang, C.; Cui, N.; Zhang, Q. State of charge estimation for lithium-ion batteries based on extended Kalman filter optimized by fuzzy neural network. *Control Theory Appl.* **2016**, *33*, 212–220. (In Chinese)
53. Li, S.; Ma, W.; Liu, J.; Chen, H. A Kalman gain modify algorithm based on BP neural network. In Proceedings of the International Symposium on Communications and Information Technologies, Qingdao, China, 26–28 September 2016; pp. 453–456.
54. Zheng, Y.Y.; Kong, J.L.; Jin, X.B.; Wang, X.Y.; Su, T.L.; Wang, J.L. Probability fusion decision framework of multiple deep neural networks for fine-grained visual classification. *IEEE Access* **2019**, *7*, 122740–122757. [[CrossRef](#)]
55. Pei, E.; Xia, X.; Yang, L.; Jiang, D.; Sahli, H. Deep neural network and switching Kalman filter based continuous affect recognition. In Proceedings of the IEEE International Conference on Multimedia & Expo Workshops, Seattle, WA, USA, 11–15 July 2016; pp. 1–6.
56. Menezes, J.M.P., Jr.; Barreto, G.A. Long-term time series prediction with the NARX network: An empirical evaluation. *Neurocomputing* **2008**, *71*, 3335–3343. [[CrossRef](#)]
57. Goudarzi, S.; Jafari, S.; Moradi, M.H.; Sprott, J.C. NARX prediction of some rare chaotic flows: Recurrent fuzzy functions approach. *Phys. Lett. A* **2016**, *380*, 696–706. [[CrossRef](#)]
58. Ouyang, H. Nonlinear autoregressive neural networks with external inputs for forecasting of typhoon inundation level. *Environ. Monit. Assess.* **2017**, *189*, 376. [[CrossRef](#)] [[PubMed](#)]
59. Bai, Y.; Jin, X.; Wang, X.; Su, T.; Kong, J.; Lu, Y. Compound autoregressive network for prediction of multivariate time series. *Complexity* **2019**, *2019*, 9107167. [[CrossRef](#)]
60. Bai, Y.; Wang, X.; Sun, Q.; Jin, X.B.; Wang, X.K.; Su, T.L.; Kong, J.L. Spatio-temporal prediction for the monitoring-blind area of industrial atmosphere based on the fusion network. *Int. J. Environ. Res. Public Health* **2019**, *16*, 3788. [[CrossRef](#)] [[PubMed](#)]

61. Lourakis, M.I.A. A brief description of the Levenberg-Marquardt algorithm implemented by levmar. *Found. Res. Technol.* **2005**, *4*, 1–6.
62. Wilamowski, B.M.; Yu, H. Improved computation for Levenberg-Marquardt training. *IEEE Trans. Neural Netw.* **2010**, *21*, 930–937. [[CrossRef](#)]
63. Ma, H.; Pan, J.; Ding, F.; Xu, L.; Ding, W. Partially-coupled least squares based iterative parameter estimation for multi-variable output-error-like autoregressive moving average systems. *IET Control Theory Appl.* **2019**, *13*, 3040–3051. [[CrossRef](#)]
64. Liu, S.Y.; Ding, F.; Xu, L.; Hayat, T. Hierarchical principle-based iterative parameter estimation algorithm for dual-frequency signals. *Circuits Syst. Signal Process.* **2019**, *38*, 3251–3268. [[CrossRef](#)]
65. Zheng, Y.Y.; Kong, J.L.; Jin, X.B.; Wang, X.Y.; Su, T.L.; Zuo, M. CropDeep: The crop vision dataset for deep-learning-based classification and detection in precision agriculture. *Sensors* **2019**, *19*, 1058. [[CrossRef](#)]
66. Xu, L. Application of the Newton iteration algorithm to the parameter estimation for dynamical systems. *J. Comput. Appl. Math.* **2015**, *288*, 33–43. [[CrossRef](#)]
67. Xu, L. The damping iterative parameter identification method for dynamical systems based on the sine signal measurement. *Signal Process.* **2016**, *120*, 660–667. [[CrossRef](#)]
68. Ding, F.; Wang, F.F.; Xu, L.; Wu, M.H. Decomposition based least squares iterative identification algorithm for multivariate pseudo-linear ARMA systems using the data filtering. *J. Frankl. Inst.* **2017**, *354*, 1321–1339. [[CrossRef](#)]
69. Ding, F.; Liu, G.; Liu, X.P. Partially coupled stochastic gradient identification methods for non-uniformly sampled systems. *IEEE Trans. Autom. Control.* **2010**, *55*, 1976–1981. [[CrossRef](#)]
70. Ding, J.; Ding, F.; Liu, X.P.; Liu, G. Hierarchical least squares identification for linear SISO systems with dual-rate sampled-data. *IEEE Trans. Autom. Control.* **2011**, *56*, 2677–2683. [[CrossRef](#)]
71. Xu, L.; Chen, L.; Xiong, W.L. Parameter estimation and controller design for dynamic systems from the step responses based on the Newton iteration. *Nonlinear Dyn.* **2015**, *79*, 2155–2163. [[CrossRef](#)]
72. Xu, L. The parameter estimation algorithms based on the dynamical response measurement data. *Adv. Mech. Eng.* **2017**, *9*, 1–12. [[CrossRef](#)]
73. Wang, Y.J.; Ding, F. Novel data filtering based parameter identification for multiple-input multiple-output systems using the auxiliary model. *Automatica* **2016**, *71*, 308–313. [[CrossRef](#)]
74. Ding, F.; Liu, Y.J.; Bao, B. Gradient based and least squares based iterative estimation algorithms for multi-input multi-output systems. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **2012**, *226*, 43–55. [[CrossRef](#)]
75. Xu, L.; Ding, F.; Gu, Y.; Alsaedi, A.; Hayat, T. A multi-innovation state and parameter estimation algorithm for a state space system with d-step state-delay. *Signal Process.* **2017**, *140*, 97–103. [[CrossRef](#)]
76. Ma, H.; Pan, J.; Lv, L.; Xu, G.; Ding, F.; Alsaedi, A.; Hayat, T. Recursive algorithms for multivariable output-error-like ARMA systems. *Mathematics* **2019**, *7*, 558. [[CrossRef](#)]
77. Ma, J.X.; Xiong, W.L.; Chen, J.; Feng, D. Hierarchical identification for multivariate Hammerstein systems by using the modified Kalman filter. *IET Control Theory Appl.* **2017**, *11*, 857–869. [[CrossRef](#)]
78. Ding, F.; Liu, X.G.; Chu, J. Gradient-based and least-squares-based iterative algorithms for Hammerstein systems using the hierarchical identification principle. *IET Control Theory Appl.* **2013**, *7*, 176–184. [[CrossRef](#)]
79. Wan, L.J.; Ding, F. Decomposition- and gradient-based iterative identification algorithms for multivariable systems using the multi-innovation theory. *Circuits Syst. Signal Process.* **2019**, *38*, 2971–2991. [[CrossRef](#)]
80. Jin, X.; Yang, N.; Wang, X.; Bai, Y.; Su, T.; Kong, J. Integrated predictor based on decomposition mechanism for PM2.5 long-term prediction. *Appl. Sci.* **2019**, *9*, 4533. [[CrossRef](#)]

