

# WNN Learning Algorithm Based on Unscented Kalman Particle Filter

Niu Junfeng<sup>1,a</sup>, Wu Yarong<sup>2,b</sup>, Li Huaping<sup>3,c</sup>

<sup>1</sup>XiJing College, Xi'an, Shaanxi, 710123, China;

<sup>2</sup>Air Traffic Control and Navigation College, Air Force Engineering University, Xi'an, Shaanxi, 710051, China;

<sup>3</sup>PLA, Air Force, Xi'an Flight Academy, Xi'an, Shaanxi, 710306

<sup>a</sup>guofeng313@163.com; <sup>b</sup>2127gwm@163.com; <sup>c</sup>fyw2121@sohu.com

**Keywords:** Wavelet Neural Network, Kalman Filter, Unscented Transform, Particle Filter.

**Abstract.** In order to improve the nonlinear modeling capability of Wavelet Neural Network (WNN), a learning algorithm of WNN based on modified Unscented Kalman Particle Filter (UPF) is proposed. In the algorithm, first a minimal skew strategy is introduced to reduce the number of Sigma sampling points of Unscented Transform (UT), improving Unscented Kalman Filter (UKF), and then the improved UKF is used to select the importance density function of Particle Filter (PF), forming a new UPF (SUPF), finally, SUPF is taken as learning algorithm of WNN for training and test. The simulation results indicate that, for the learning problem of WNN, the model precision of UPF based on new sampling strategy is approximately close to that of simple UPF, but the former has faster training rate and higher learning efficiency, which validate its feasibility and effectiveness.

## 1. Introduction

WNN is a new neural network based on wavelet analysis theory. It can overcome many shortcomings of traditional neural network with rigorous theory and strong mapping ability. However, it is very important to choose the appropriate learning algorithm to improve the potential performance of WNN, which is also a hot issue in the field. Besides traditional BP learning algorithm [1][2], Particle Filter (PF) algorithm can also be used for WNN learning. PF completely breaks the theoretical framework of Kalman filter without limitation of process noise and observation noise of the system, and is applicable to the non-linear stochastic system with non-Gaussian background represented by any general state space model [3]. But its performance depends on the importance density function selected. The study shows that, Unscented Kalman Filter (UKF) algorithm can be used to generate the importance density function, which can obtain the better filter effect. However, in this way, the computational complexity is too large [4]. The best way to reduce the computational complexity is to reduce the number of Sigma sample points of Unscented Transformation (UT) in UKF algorithm as much as possible while ensuring the similarity of filter accuracy.

Based on the above analysis, in the process of using UKF algorithm to generate the importance density function of PF algorithm, an improved Sigma point sampling strategy is introduced in order to reduce the computational complexity for improving the learning efficiency of WNN. It also is verified by an example.

## 2. Unscented Kalman Filter (UKF)

UT can implicitly correct the deviation of the second-order term without calculating Jacobian and Hessian matrices. In UT, the Sigma point sampling strategy is different, its accuracy and calculation are also different. Usually, increasing the number of Sigma points can improve the

accuracy of the algorithm, at the same time, and also increase the amount of calculation. It has been shown that, UT using Minimal Skew Sigma sampling strategy (SUT), can significantly reduce computational complexity [6]. Its Sigma point sampling steps is as follows:

1. Select  $0 \leq W_0 \leq 1$
2. Select the weight value of Sigma point

$$W_i = (1 - W_0) / (n + 1), i = 1, 2, \dots, n + 1 \quad (1)$$

3. Initialize the sequence

$$\chi_0^1 = [0]; \chi_1^1 = \left[ -\frac{1}{\sqrt{2W_1}} \right]; \chi_2^1 = \left[ \frac{1}{\sqrt{2W_1}} \right] \quad (2)$$

4. For the state vector dimension  $j = 2, \dots, n$ , the iteration formula is

$$\chi_i^j = \begin{cases} \begin{bmatrix} \chi_0^{j-1} \\ 0 \end{bmatrix} & i = 0 \\ \begin{bmatrix} \chi_i^{j-1} \\ 1 \\ \sqrt{j(j+1)W_1} \end{bmatrix} & i = 1, \dots, j \\ \begin{bmatrix} 0_{j-1} \\ j \\ \sqrt{j(j+1)W_1} \end{bmatrix} & i = j + 1 \end{cases} \quad (3)$$

5. Add the mean and variance information to the generated Sigma points

$$\chi_i = \bar{x} + \sqrt{P_x} \chi_i^j \quad (4)$$

The set of above Sigma points can be recorded as  $\chi_{redu}$ . It is easy to see that, in SUT, the weight value of all Sigma points is  $W_i = (1 - W_0) / (n + 1)$  except for the 0th Sigma point. Compared with UT with Sigma point of symmetric distribution, the number of Sigma points is reduced from  $(2n + 1)$  to  $(n + 2)$  (including the center point), which results in the reduction of calculation.

SUT is introduced in UKF to conduct the recursion and update of the state and covariance of nonlinear model, namely SUKF. The whole calculation process does not need to understand the details of nonlinear function, and its estimation accuracy to mean and covariance is roughly equivalent to that of UKF based on UT, and the calculation is significantly reduced [7].

### 3. SUPF Algorithm

PF is a Bayesian filter based on Monte Carlo method. Its theoretical basis is the law of large numbers and the central limit theorem. It uses a series of weighted random particles in the state space to approximate the posterior probability density function of system state. The precision can be approximated to the optimal estimation. It is applicable to any nonlinear system which can be described by the state space model and has great advantage in state and parameter estimation of nonlinear and non-Gaussian systems. The key of PF is the selection of the importance density function.

The closer the importance density function is to the posterior probability density function, the better the filtering performance is, the closer the accuracy is to the optimum, the more robust the value is, however, how to select the importance density function is a difficult problem. When the basic PF conducts the importance sampling, the state transition probability density function is used,

but the particle is heavily dependent on the state transition model. If the model selected is not accurate, it may lead to larger sampling bias. Especially, if the likelihood function is too concentrated, or appears at the tail of the transition probability density function, the deviation is more obvious. For this reason, it is necessary to explore the importance density function that is closer to the posterior probability density function. In reference [8], EKF algorithm is used to calculate the importance function of PF algorithm, that is, EPF algorithm [9], which can be approximated by the first order Taylor series near the particles at previous time. In [10], UKF algorithm is used to generate the importance function of PF algorithm (UPF algorithm), which has a larger overlapping portion with support set of the real state transition probability density function, and the filtering accuracy is higher. The state mean and variance can be approximated to the third order, but it needs UT and the calculation is too large. It can be seen from the foregoing that SUKF algorithm is an improvement of UKF algorithm, and the accuracy of the both is quite equivalent, but in the former, the computation is significantly reduced. Therefore, SUKF algorithm is used to generate the importance density function (SUPF algorithm), which can obtain similar PF performance of UKF algorithm in theory and significantly improve the computational efficiency. Based on this, the step of SUPF algorithm is as follows:

1. Extract  $N$  particles at the time  $k=0$  from the known initial distribution, that is  $\{x_0^i\}_{i=1}^N \sim p(x_0)$ , and initial weight of each particle  $w_0^i=1/N$ ,  $i=1,2,\dots,N$ .

2. Importance sampling,  $k=1,2,\dots,T$

(1) For each sampling point  $x_{k-1}^i, i=1,2,\dots,N$ , use SUKF to obtain  $\hat{x}_k^i, P_k^i$ , and suggest the distribution function  $q(x_k^i | x_{0:k-1}^i, y_{1:k}) = N(\hat{x}_k^i, P_k^i)$ , where  $N(\hat{x}_k^i, P_k^i)$  is the normal distribution of mean  $\hat{x}_k^i$  and variance  $P_k^i$ . Conduct the sampling from the distribution

$$\{\hat{x}_k^i\}_{i=1}^N \sim q(x_k^i | x_{0:k-1}^i, y_{1:k}) = N(\hat{x}_k^i, P_k^i) \quad (5)$$

(2) Calculate the weight of particle

$$w_k^i = w_{k-1}^i \frac{p(y_k | x_k^i) p(x_k^i | x_{k-1}^i)}{q(x_k^i | x_{0:k-1}^i, y_{1:k})} \quad (6)$$

(3) Normalize the weight of particle

$$\tilde{w}_k^i = w_k^i / \sum_{i=1}^N w_k^i \quad (7)$$

3. Resample. The weight of each particle is evaluated, the particles with larger weight value are copied, the particles with smaller weight value are reduced. That is, the old particle set  $\{\hat{x}_k^i, \tilde{w}_k^i, i=1,2,\dots,N\}$  is replaced by the new particle set  $\{x_k^j, w_k^j, j=1,2,\dots,N\}$ , and the same weight  $1/N$  is assigned to each particle. Need to meet the condition  $P\{x_k^j = \hat{x}_k^i\} = \tilde{w}_k^i$  when replacing.

4. Output the results

$$\hat{x}_k = \sum_{i=1}^N w_k^i x_k^i \quad (8)$$

$$P_k = \sum_{i=1}^N w_k^i (x_k^i - \hat{x}_k)(x_k^i - \hat{x}_k)^T \quad (9)$$

5.  $k = k + 1$ , and go to step 2.

#### 4. WNN Algorithm Based on SUPF

In WNN structure in Fig. 1, the input vector is  $X=[x_1, x_2, \dots, x_T]^T$ , the output vector is

$Y=[y_1, y_2, \dots, y_L]^T$ . The training of WNN is to conduct the optimal estimation of the parameter  $\theta=[w_{ji}, w_{ij}, a_j, b_j]^T$ , so as to obtain the optimal network parameters. Therefore,  $\theta$  can be regarded as the state variable of the network system, the output can be regarded as the observation equation of the network. Then the state space model of WNN can be expressed as

$$\theta_k = \theta_{k-1} + U_k \quad (10)$$

$$Y_k = H(\theta_k, X_k) + V_k \quad (11)$$

where  $X_k$  is the observation input at time  $k$ .  $Y_k$  is the observation output at the time  $k$ .  $\theta_k$  is the system state vector at time  $k$ .  $H(\theta_k, X_k)$  is the parameterized nonlinear function.  $U_k$  is the process noise that is Gaussian white noise with the mean 0 and the variance  $Q_k$ .  $V_k$  is the measurement noise that is Gaussian white noise with the mean 0 and the variance  $R_k$ .

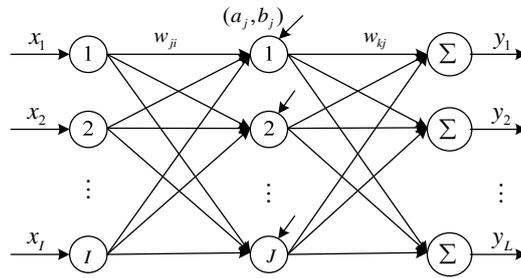


Fig. 1. Structure of WNN

SUPF algorithm is a global optimization strategy. Using SUPF algorithm to train the WNN model, that is to use SUPF algorithm to optimize the network parameters of WNN for optimal or suboptimal solution. Specifically, for the formula (10) and (11), SUPF algorithm can be used to conduct the optimal estimation to the parameter  $\theta_k$ . The realization flow of SUPF- WNN is shown in Fig. 2.

## 5. Experiment Simulation

For given SISO nonlinear system

$$y(k) = \frac{16u(k-1) + 8y(k-1)}{3 + 4u(k-1)^2 + 4y(k-1)^2} + \frac{2}{10}u(k-1) + \frac{2}{10}y(k-1) \quad (12)$$

where  $u(k)$  is the input signal that is the random value obeying uniform distribution in  $[-1, 1]$ . Firstly, 300 data is calculated by the formula (12) while  $y(0) = 0$ , and then according to the pattern with inputs  $[u(k-1), u(k-2), y(k-1), y(k-2), y(k-3)]$  and output  $y(k)$ , 200 samples are obtained, where the first 100 samples are used for training and the remainder samples for test. Morlet wavelet can be selected as the activation function in hidden layer node of WNN, namely  $\psi(t) = \cos(1.75t)e^{-t^2/2}$ . The number of hidden layer nodes can be taken as 20, the initial value of the parameters is the random number in  $[0, 1]$ , the process noise variance  $Q_k$  can be taken as 1, and the measured noise variance  $R_k$  can be taken as 0.1. To verify the proposed algorithm, BP, EKF, UKF, SPF, EPF, UPF and SUPF algorithm were respectively used to train WNN, and the test results are compared. The number of particles involved in various PF algorithms can be taken as 50. The experiment runs 50 times independently, and the maximum number of training times of WNN is 200.

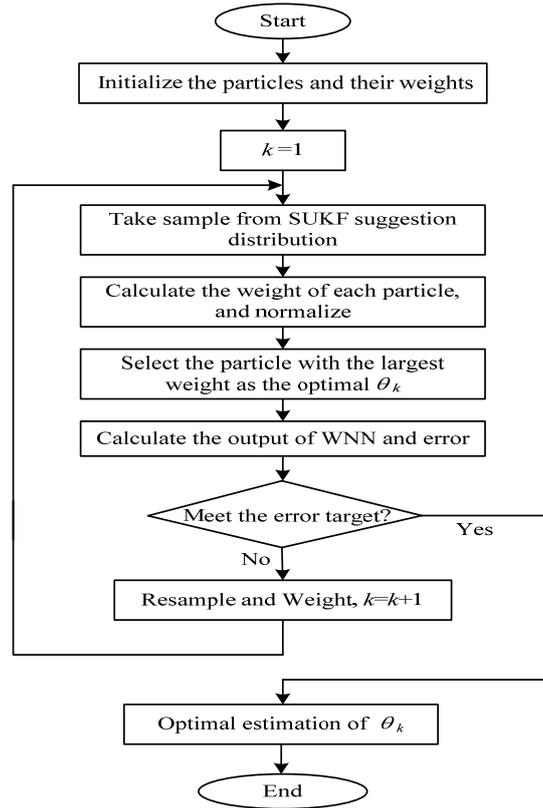


Fig. 2. Realization flow of SUPF-WNN

The comparison on test performance and average training time of WNN models based on 7 learning algorithms is shown in Table 1. As the number of particles is 50, 100, 150, 200, 250, respectively, the comparison test MSE and training time of WNN based on 4 PF algorithms is shown in Fig. 3 and Fig. 4. Fig. 5 shows convergence curves of SUPF-WNN algorithms.

It can be seen from the simulation results that: compared with BP algorithm, the other 6 filtering algorithms all improve the performance of WNN with varying degrees. For EKF algorithm, since the Gaussian hypothesis and the linearization process introduce too much error, the improvement of WNN based on EKF is not better than that based on UKF algorithm. And UPF algorithm is used to train WNN, which has a better effect compared with SPF and EPU algorithm, the main reasons: First, UPF algorithm can take full advantage of the latest observed information; Second, the importance density function is generated by UKF algorithm, which is closer to the posterior probability density function, the accuracy can reach the third level of Taylor series. It must be pointed out that the computational complexity of UPF algorithm is proportional to the number of Sigma points in UT transformation, which makes UPF-WNN require more learning time. The test accuracy is equivalent to UPF-WNN, but the latter can reach under the condition of less Sigma points, and the learning time is significantly reduced. The main reason is that, in SUPF algorithm, while using traditional UKF to generate the importance density function, UT with Sigma points obeying symmetrical distribution is replaced by SUT. It not only inherits good performance of UKF, but also reduces the amount of calculation greatly.

Table 1 Performance Comparison of WNN Based on Various Learning Algorithms

Algorithm	Test MSE	Training time (s)
BP-WNN	1.6605	7.9
EKF-WNN	0.2672	9.8
UKF-WNN	0.1258	12.4
SPF-WNN	0.0859	7.6
EPF-WNN	0.0601	8.4
UPF-WNN	0.0226	17.9
SUPF-WNN	0.0245	10.1

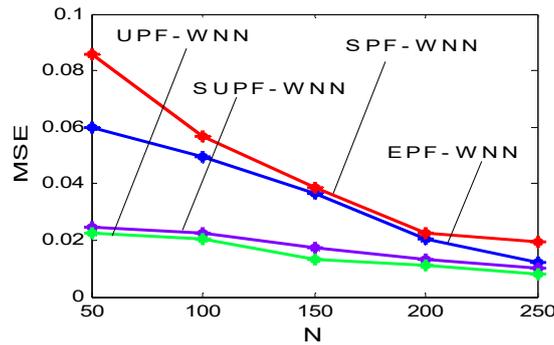


Fig. 3. Comparison on test MSE of WNN based on 4 PF algorithms

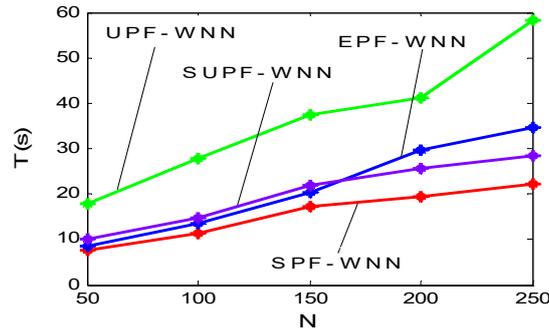


Fig. 4. Comparison on training time of WNN based on 4 PF algorithms

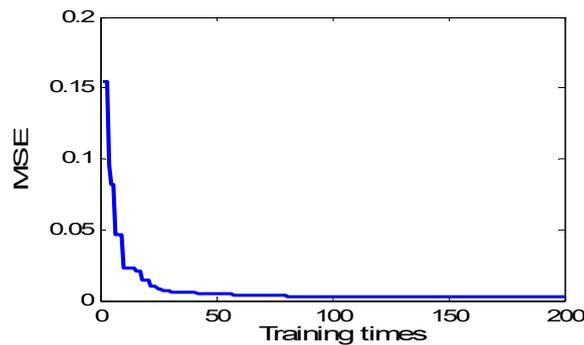


Fig. 5. Convergence curves of SUPF-WNN training

## 6. Conclusion

For the learning problem of WNN, a PF learning algorithm based on improved UKF is proposed. In the algorithm, UT with minimum skew Sigma point sampling strategy is introduced into UKF algorithm to replace UT with traditional symmetric distribution computationally. Then, improved UKF algorithm is adopted to select the importance density function of PF algorithm. Finally WNN parameters are optimized by proposed method to complete WNN training. The simulation results show that PF learning algorithm can improve the performance of WNN, shorten the training time.

At the same time, it also provides a powerful method for modeling and predicting problems of nonlinear systems.

## References

- [1] Q. H. Zhang, A. Benveniste, "Wavelet network," IEEE Transactions on Neural Networks, vol. 3, No. 6, 1992, pp. 889-898.
- [2] Q. H. Zhang, "Using wavelet network in nonparametric estimation," IEEE Transactions on Neural Networks, vol. 8, No. 2, 1997, pp. 227-236.
- [3] Y. P. Chen, L. X. Wang, S. T. Huang, "Neural network learning algorithm based on particle filter," Engineer Journal of Wuhan University, vol. 39, No. 6, 2006, pp. 86-88.
- [4] S. J. Julier, J. K. Uhlmann, H. F. Durrant-whyte, "A new approach for the nonlinear transformation of means and covariances in filters and estimators," IEEE Transactions on Automatic Control, vol. 45, No. 3, 2000, pp. 477-482.
- [5] S. J. Julier, J. K. Uhlmann, "Unscented filtering and nonlinear estimation," Proceedings of the IEEE, vol. 92, No. 3, 2004, pp. 401-422.
- [6] W. N. Jiang, H. Y. Zhou, X. J. Duan, X. G. Pan, "UKF with scaled minimal skew simplex and its application in satellite orbit determination," Aerospace Shanghai, No. 5, 2008, pp. 12-15.
- [7] L. Lv, K. Paul, X. S. Gan, H. Qu, H. T. Zhao, "RBF neural network algorithm based on UKF with a scaled minimal skew simplex sigma point sampling strategy," Fire Control & Command Control, vol. 39, No. 12, 2014, pp. 80-83.
- [8] N. J. Gordon, D. J. Salmond, A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation," IEE Proceedings-F, vol. 140, No. 2, 1993, pp. 107-113.
- [9] F. S. Wang, Q. Guo, "Neural network training based on the extended Kalman particle filter," Computer Engineering and Science, vol. 32, No. 5, 2010, pp. 48-50.
- [10] G. Q. Yu, Z. Liu, X. Liu, "Maneuvering target tracking based on unscented particle filter aided by neural network," Ship Electronic Engineering, vol. 29, No. 12, 2009, pp. 49-51.