A hybrid learning algorithm of radial basis function networks for reliability analysis

Dequan Zhang, Ning Zhang, Nan Ye. Jianguang Fang, Xu Han*

***Abstract*—With the wide application of industrial robots in the field of precision machining, reliability analysis of positioning accuracy has been becoming increasingly more important for industrial robots. Since the industrial robot is a complex nonlinear system, the traditional approximate reliability methods often produce unreliable results in analyzing its positioning accuracy. In order to study the reliability of industrial robot positioning accuracy more efficiently and accurately, a radial basis function network is used to construct the mapping relationship between the uncertain parameters and the position coordinates of the end-effector and analyze the positioning accuracy reliability combined with the Monte Carlo simulation method. A novel hybrid learning algorithm of training radial basis function network, which integrates the clustering learning algorithm with the orthogonal least squares learning algorithm, is proposed in this paper. Examples are presented to illustrate the high accuracy and high efficiency of the proposed method.**

***Keywords*—industrial robot, reliability analysis, radial basis function network, positioning accuracy, hybrid learning algorithm.**

I. INTRODUCTION

INDUSTRIAL robots are widely used in automobiles, electronics, new energy batteries and high-end equipment manufacturing to improve the productivity and precision of repetitive machining [1]. However, due to the existence of uncertain parameters such as joint clearance, axial length machining error and installation error, the actual moving position of the end-effector of the industrial robot deviates from the target position, which is called positioning error. Therefore, it is necessary to analyze the reliability of industrial robots positioning accuracy considering the influence of uncertain parameters.

The reliability analysis of industrial robot positioning accuracy belongs to mechanism reliability analysis. The mechanism reliability analysis method is developed from a planar and spatial mechanism to spatial multiple degree-of-freedom robots. In the context of planar mechanism reliability analysis, Yan and Guo [2] proposed a kinematics accuracy analysis method of the flexible planar mechanism with random variables and proved the influence of joint clearance and random link length on mechanism motion through simulation. Wang et al. [3] applied a hybrid dimension reduction method to compute the mean and variance of the mechanism output error which was assumed to be normally distributed. Zhang et al. [4] proposed a time-dependent reliability analysis method with response surface to estimate the time-dependent reliability for nondeterministic structures under stochastic. Yu et al. [5] proposed a novel time-variant reliability analysis method for multiple failure modes and temporal parameters based on the combination of the extreme value moment method and improved maximum entropy method. Jha et al. [6] used a combination of high-dimensional model representation and artificial neural network (ANN) to approximate the implicit limit state function to estimate the failure probability of structures affected by random parameters. In order to improve the accuracy of reliability analysis, Huang et al. [7] used the first-order asymptotic integration method for reliability analysis. Kim et al. [8] used the first-order reliability method to analyze the reliability problem of an open-loop manipulator and also assumed that all parameters are normally distributed. Sun et al. [9] established a surrogate model based on neural networks and proposed a kinematics accuracy analysis method for planar mechanisms with clearance involving random and cognitive uncertainties. In recent years, the reliability analysis of spatial mechanism including spatial multi-degree of freedom robots has gained more and more attention of researchers. Bhatti and Rao [10] defined the joint probability density function to analyze the kinematic reliability of the robot by assuming that the position and direction of the end-effector obey a joint normal distribution. Then, Rao and Bhatti [11] considered the relationship between the tolerance, the configuration and the reliability of the manipulator, and proposed a probabilistic method to study the kinematics and dynamics of the manipulator. Taking a three degree-of-freedom parallel robot as an example, Cui et al. [12] analyzed the influence of error sensitivity on position parameters and structural parameters and calculated the motion reliability of some positions and the dynamic accuracy reliability of trajectories. Chaker [13] proposed a random method to analyze

This work was supported by the National Key R&D Program of China (Grant No. 2017YFB1301300), the National Natural Science Foundation of China (Grant No. 51905146), the Key R&D Plan Program of Hebei Province (Grant No. 19211808D) and the Research Program of Education Bureau of Hebei Province (Grant No. QN2019141).

The authors Dequan Zhang, Ning Zhang, Nan Ye and Xu Han are with State Key Laboratory of Reliability and Intelligence of Electrical Equipment, School of Mechanical Engineering, Hebei University of Technology, Tianjin, 300401, China (e-mail: dequan.zhang@hebut.edu.cn (D. Q. Zhang), zhangning_zone@163.com (N. Zhang), yenan@hebut.edu.cn (N. Ye) and xhan@hebut.edu.cn (X. Han)).

The author Jianguang Fang with School of Civil and Environmental Engineering, University of Technology Sydney, Sydney, NSW 2007, Australia. (email: fangjg@gmail.com).

the influence of manufacturing error and clearance on the pose error of spherical parallel robot. Zhu and Ting [14] proposed a method based on probability density function to study the positional uncertainty of planar and spatial robots affected by joint clearance, where the joint clearance can be arbitrary. Wu and Rao [15] discussed the optimal allocation of joint angular tolerances by modelling tolerances as interval variables. Wu et al. [16] established a framework for reliability analysis of industrial robots and the positional accuracy reliability for single coordinate, the positional accuracy reliability for a single point, the positional accuracy reliability of multiple points and the trajectory accuracy reliability of industrial robots are comprehensively analyzed. Pandey et al. [17] used the method of fractional moments combined with the principle of maximum entropy to discretize the trajectory of industrial robots into a series of discrete points for reliability analysis of positioning accuracy. Wu et al. [18] used the Sobol' method to analyze the sensitivity of the uncertain variables for the positioning accuracy of the industrial robots and applied the point estimation method to fit the probability density function. Zhang and Han [19] transformed the three dependent coordinates of the end-effector of the industrial robot into three independent standard normal variables and applied the saddle point approximation to compute the kinematic reliability. The above research analyzed the reliability of the mechanism from multiple perspectives, but when performing reliability analysis for a complex system the computational time is too long and there is a limit to the distribution type of the uncertain parameters. Therefore, how to improve the efficiency without sacrificing the accuracy when the uncertain parameters are randomly distributed is still an urgent problem to be solved in the mechanism reliability analysis.

Radial basis function network (RBFN) has been widely used in fault diagnosis [20], image recognition [21] and reliability analysis [22]. The RBFN is applied in this study to reliability analysis of industrial robot positioning accuracy. Moody and Darken [23] proposed a training method of radial basis function (RBF) that uses the *k*-means clustering method to obtain the RBF center and used the least squares algorithm [24] to evaluate the weight. Chen et al. [25] proposed an orthogonal least squares learning algorithm to train the RBFN, which improves the network convergence rate and reduces the network scale. Alexandridis et al. [26] proposed a fast noniterative categorical clustering algorithm to training RBFN and a novel learning scheme for categorical data based on RBFN. Wang et al. [27] proposed a fault tolerant algorithm to train an RBFN and select the RBF centers simultaneously. Han et al. [28] designed a self-organizing RBFN with the aid of adaptive particle swarming optimization that can optimize both the network size and the parameters of an RBFN simultaneously. Lai et al. [29] proposed a set of novel radial basis functions with adaptive input and composite trend representation for portfolio selection. Meng et al. [30] trained RBFN through hybrid learning algorithms and applied them in the field of face recognition. Deng et al. [31, 32] used ANN and RBFN technology to model and approximate the implicit function or partial derivative, and proposed a structural reliability analysis method suitable for the implicit function. Dai et al. [33] proposed an RBFN method based on support vector machine and the improved RBF model was used to approximate the limit state function and to estimate the failure probability by combining with the reliability method. Kavousifar et al. [34] proposed a reliability analysis method that combined the RBF and the fuzzy theory to accurately evaluate the reliability of the combined power system. Tan et al. [35] compared the similarities and differences between the RBFN and support vector machine and illustrated the applicability of the two methods by examples. Chau et al. [36] used the RBF to approximate the implicit limit state function and proposed a first-order reliability analysis method based on the response surface. Li et al. [37] maximized the probability density function by adding a new sample point in each iteration, which improved the accuracy of the RBF surrogate model in the important region of the failure boundary. Then, Li et al. [38] decomposed the original reliability optimization problem into a series of deterministic problems based on surrogate models and Monte Carlo simulation by the above method, which reduced the costly black box call in reliability optimization. The above RBFN research has improved training speed and network accuracy. However, there still exist problems such as slow convergence of the iterative process, excessive network size, and insufficient use of sample information. And there is also no illustration of which cluster centers and sample points are more suitable as the center of the RBF.

The traditional approximate reliability analysis methods, such as first-order reliability method [39] and second-order reliability method [40, 41], are relatively efficient, but there are defects with poor precision when solving highly nonlinear problems. Since industrial robots are highly nonlinear systems, these types of methods are not suitable to solve reliability problems for industrial robots. It is not easy to balance the efficiency and accuracy of the reliability analysis for industrial robots, but the widely used RBFN provides an idea to solve such problems. In this paper, a hybrid learning algorithm which combines the clustering learning algorithm with the orthogonal least squares learning algorithm is proposed to train RBFN. The proposed method adds suitable hidden node centers iteratively until the network accuracy requirement is met. On the premise of ensuring the network accuracy, the learning efficiency of the RBFN is improved. The main contributions of this study can be included: 1) The disadvantages of the clustering center learning algorithm and orthogonal least square learning algorithm are avoided and the sample information is fully utilized. 2) A new convergence criterion is proposed for the training process. 3) The effectiveness and engineering practicability of the proposed method are verified by numerical examples. 4) A new idea to solve the problem of reliability of industrial robot positioning accuracy is proposed. The proposed method is first validated through mathematical benchmark problems and then applied to reliability analysis of the industrial robot positioning accuracy. The results show that the proposed method has a good balance between computational efficiency and accuracy and is able to deal with uncertain parameters of arbitrary distribution.

The remainder of the paper is organized as follows: Section

2 mainly introduces the basic principle of a radial basis neural network; Section 3 elaborates the specific implementation process of the proposed method; Section 4 uses the numerical example to illustrate the effectiveness of the proposed method; Section 5 gives the application of the radial basis network in the industrial robots positioning accuracy; Section 6 gives some conclusions.

II. Radial Basis Function Network

As a kind of neural network, the RBFN is characterized by its simple structure, fast convergence and short calculation time. The radial basis network structure is shown in Fig. 1.

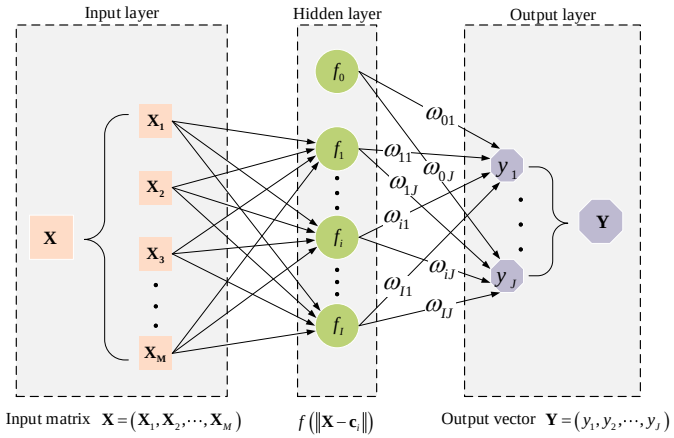


Fig. 1 Radial basis neural network structure.

The radial basis neural network is a three-layer structure: the first layer is an input layer marked as $\mathbf{X}=(\mathbf{X}_1,\mathbf{X}_2,\cdots,\mathbf{X}_M)$, where M is the number of input samples; the second layer is the hidden layer which contains I hidden nodes with output f_i (i =1,2,$\cdots$,I) and a bias f_0, and each hidden node corresponds to a basis function center $\mathbf{c}_i$. The ith hidden node output is $f(\|\mathbf{X}-\mathbf{c}_i\|)$, where $\|\cdot\|$ is the Euclidian distance between two samples, and f denotes the RBF whose type is shown in Table 1; the third layer is the output layer denoted as a J-dimensional vector $\mathbf{Y}$. The transformation from the input layer to the hidden layer is non-linear, directly connected without the need for weights. But the transformation from the hidden layer to the output layer is linear, connected by weights. For an input sample $\mathbf{X}_K$, the output of the jth output neuron of the RBFN can be expressed as

$$\hat{y}_j(\mathbf{X}_K)=\omega_{oj}+\sum_{i=1}^{I}\omega_{ij}f(\|\mathbf{X}_K-\mathbf{c}_i\|)\quad j=1,2,\cdots,J \tag{1}$$

where ω_{ij} is the connection weight of the ith neuron of the hidden layer to the jth neuron of the output layer.

Assuming that the number of input samples is n and the number of central points of the network hidden layer is m, the radial basis network can be formulated as

$$\mathbf{Y}=\mathbf{FW} \tag{2}$$

where

$$F=\begin{bmatrix} f_0 & f(\|\mathbf{X}_1-\mathbf{c}_1\|) & f(\|\mathbf{X}_1-\mathbf{c}_2\|) & \cdots & f(\|\mathbf{X}_1-\mathbf{c}_m\|) \\ f_0 & f(\|\mathbf{X}_2-\mathbf{c}_1\|) & f(\|\mathbf{X}_2-\mathbf{c}_2\|) & \cdots & f(\|\mathbf{X}_2-\mathbf{c}_m\|) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ f_0 & f(\|\mathbf{X}_n-\mathbf{c}_1\|) & f(\|\mathbf{X}_n-\mathbf{c}_2\|) & \cdots & f(\|\mathbf{X}_n-\mathbf{c}_m\|) \end{bmatrix} \tag{3}$$

$$\mathbf{W}=[\omega_0,\omega_1,\cdots,\omega_m]^{\mathrm{T}} \tag{4}$$

$$\mathbf{Y}=[y_1,y_2,\cdots,y_n]^{\mathrm{T}} \tag{5}$$

where $\omega_i\,(i=0,1,\cdots,m)$ is the connection weight of the ith neuron of the hidden layer to the output layer, $\mathbf{X}_i$ is the ith input sample, y_i is the response of the ith input sample.

Since matrix $\mathbf{F}$ is non-singular, Eq. (2) has a unique solution

$$\mathbf{W}=\mathbf{F}^{+}\mathbf{Y} \tag{6}$$

where $\mathbf{F}^{+}$ is the generalized inverse matrix of $\mathbf{F}$. Therefore, the output of the network can be predicted as,

$$\hat{y}(\mathbf{X})=\mathbf{f}(\mathbf{X})^{\mathrm{T}}\mathbf{F}^{+}\mathbf{Y} \tag{7}$$

where the value of $\mathbf{f}(\mathbf{X})$ depends on the Euclidean distance between the predicted point and the center point. For a new sample point $\mathbf{x}$, the predicted value $\hat{y}(\mathbf{X})$ can be obtained by calculating Eq. (7) after the network training is completed.

TABLE I
Different Types of RBF

Type	Function form $f(r)$
Gaussian	$\exp(-\alpha r^2)$
Cubic	$(r^3+\alpha)^3$
Multiquadric	$(r^2+\alpha^2)^{\frac{1}{2}}$
Inverse Multiquadric	$(r^2+\alpha^2)^{-\frac{1}{2}}$
Thin plate spline	$r^2\log(1+\alpha r^2)$

α is the shape parameter in Table 1, $r=\|\cdot\|$ is the Euclidian distance between two points. Different shape parameters can result in different influence fields for each basis function, and thus different constructions and accuracies of the RBFN. For a Gaussian function, the larger the shape parameter the smaller the influence domain of the function and the worse the smoothness of the RBF. Shape parameters are generally determined by empirical formulas or other optimization methods [42].

III. Hybrid Learning Method For RBF Network (RBF-HLA)

There are many learning algorithms for radial basis networks, such as self-organizing selection center learning algorithm [43] and orthogonal least squares (OLS) learning algorithm [25]. The learning algorithm generally achieves a better fitting effect by training the basis function center, the basis function parameter and the weight of the hidden layer to the output layer. But it is usually difficult for a single learning algorithm to meet the needs of high precision. A method choosing some data points as centers usually leads to an unnecessarily large network to achieve a satisfactory performance which often adds computational burden and causes numerical ill-conditioned issues. Self-organizing selection center learning algorithm can adjust RBF centers to achieve a better network performance and is more intelligent than selecting centers arbitrarily. However,

the adjusted center usually does not include the sample points, even if there is a suitable basis function center in the sample point. In addition, Orthogonal least squares learning algorithm can choose the appropriate center among the sample points according to the magnitude of the influence on the result but cannot select the appropriate centers outside the sample set.

The self-organizing selection center learning algorithm usually uses the *k*-means clustering algorithm [44] to cluster the basis functions that can represent the input samples. The specific clustering process is as follows:

Assuming the input sample is $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_n]$, select the initial center $\mathbf{C}^t=[\mathbf{c}_1{}^t,\mathbf{c}_2{}^t,\cdots,\mathbf{c}_m{}^t]$ and calculate the Euclidean distance d_{ij} between the ith input sample and the jth initial center,

$$d_{ij}=\|\mathbf{x}_i-\mathbf{c}_j{}^t\| \quad i=1,2,\cdots,n \quad j=1,2,\cdots,m \tag{8}$$

For an input sample, the corresponding cluster number should be

$$j=\operatorname{argmin}(d_{ij}) \tag{9}$$

where argmin indicates the value of the independent variable when the dependent variable takes the minimum value. According to the above method, all the input samples are classified into the corresponding cluster centers. The samples under the jth cluster can be expressed as $\mathbf{x}_i{}^j$, and the average of all sample points in the jth cluster is obtained as the new sample center,

$$\mathbf{c}_j{}^{t+1}=\frac{\sum_{i=1}^{n_j}\mathbf{x}_i{}^j}{n_j} \tag{10}$$

where $\mathbf{c}_j{}^{t+1}$ is the new cluster center, and n_j is the number of sample points under the cluster center. Iterate as above until the old and new sample centers are identical, i.e. $\mathbf{c}_j{}^t=\mathbf{c}_j{}^{t+1}$. Fig. 2 shows the clustering diagram.

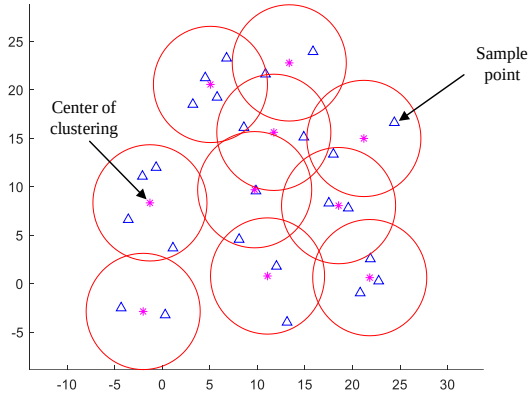


Fig. 2 Diagram of cluster sample centers.

Although the *k*-means clustering algorithm (*k*-means) seems to give reasonable sample centers, the algorithm does not indicate the superiority of using the cluster sample centers as the basis function centers compared to the original sample points as the basis function centers. Therefore, it is still necessary to further study whether the cluster sample centers or the original sample points are more suitable as the basis function centers. In the orthogonal least square method, the center of the RBF can be selected according to the influence degree of the basis vector on the result vector, which can just solve the above problems. Assume that vector $\mathbf{Y}$ can be represented linearly by a set of vectors $\mathbf{A}=[\mathbf{A}_1,\mathbf{A}_2,\cdots,\mathbf{A}_n]$,

$$\mathbf{Y}=\omega_1\mathbf{A}_1+\omega_2\mathbf{A}_2+\cdots+\omega_n\mathbf{A}_n \tag{11}$$

Obviously, each base vector $\mathbf{A}_i$ has a different degree of influence on the result $\mathbf{Y}$, and the degree of influence can be expressed by the projection size of $\mathbf{Y}$ on $\mathbf{A}_i$. The size of the projection depends on the angle between the two vectors. The same change of $\mathbf{A}_i$ can produce a greater effect on $\mathbf{Y}$ when the angle is closer to 0 or π rad. The cosine of the angle between the two vectors can be expressed by

$$\cos\theta_i=\frac{\mathbf{A}_i\cdot\mathbf{Y}}{|\mathbf{A}_i|\cdot|\mathbf{Y}|} \tag{12}$$

Define an influence factor I_i,

$$I_i=(\cos\theta_i)^2=\left(\frac{\mathbf{A}_i\cdot\mathbf{Y}}{|\mathbf{A}_i|\cdot|\mathbf{Y}|}\right)^2 \tag{13}$$

Then, a greater I_i indicates a greater influence of $\mathbf{A}_i$ on the result $\mathbf{Y}$.

Through the above method, the cluster center points and the sample points can be simultaneously used as the basis function centers. As the number of selectable center points increases, the number of basis vectors in the radial basis matrix increases. According to the size of the influence factor I_i, the most suitable basis function centers can be selected to construct the RBFN. In this way, the best approximation can be obtained with the smallest number of basis vectors. The decrease of the number of basis vectors means the decrease of the number of nodes in the hidden layer. In the case of a large amount of data, the number of fewer hidden nodes will undoubtedly improve the convergence speed of the network. The proposed method in this study is called hybrid learning algorithm for RBFN (RBF-HLA). The proposed method avoids the disadvantage that the clustering center learning algorithm cannot select the data centers according to the degree of influence on the results, meanwhile, it also overcomes the shortcoming that the orthogonal least square learning algorithm can only select the data centers in the sample points according to the degree of influence on the results. The efficiency and accuracy of the proposed algorithm will be proven through the application of four numerical examples and an engineering example. In order to meet the requirements of different accuracy, a new convergence criterion that is easy to distinguish is introduced. This criterion indicates that when network training is stopped, the error between the predicted value and the actual value of the network should be selected relative to the actual value.

The steps of the proposed hybrid learning algorithm are as follows:

Step 1. Selecting initial samples $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_n]^{\mathrm{T}}$ by the Latin hypercube sampling method (n is the number of samples) and calculating the response values $\mathbf{Y}=[y_1,y_2,\cdots,y_p]^{\mathrm{T}}$ of the performance function at the sample points. The sampling interval is generally chosen as [μ-$k\sigma$, μ+$k\sigma$], where μ is the mean value of the uncertain variable distribution, σ is the

standard deviation of the sample distribution, and *k* generally takes 1~3.

Step 2. Clustering *m* center points $\mathbf{C}=[\mathbf{c}_1,\mathbf{c}_2,\cdots,\mathbf{c}_m]$ from the initial samples by using the *k*-means clustering method.

Step 3. Combining the cluster center points with the initial sample points to form the candidate base function center points $\mathbf{S}=[\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_m,\mathbf{c}_1,\mathbf{c}_2,\cdots,\mathbf{c}_m]$.

Step 4. Selecting the Gaussian type RBF. Then using all points in **S** as the center of the alternative basis function and calculating the hidden node output matrix $\mathbf{F}=[f\|\mathbf{X}-\mathbf{S}_1\|,f\|\mathbf{X}-\mathbf{S}_2\|,\cdots,f\|\mathbf{X}-\mathbf{S}_i\|,\cdots,f\|\mathbf{X}-\mathbf{S}_{n+m}\|]$, where $\mathbf{S}_i$ represents the *i*th point in **S**.

Step 5. Finding the base vector in the base vector matrix with the largest influence factor on the result **Y**, and using the corresponding data point $\mathbf{S}_i$ as the center of basis function.

Step 6. Using the generalized inverse method to calculate the weight ω_i between the hidden layer and the output layer.

Step 7. Constructing the RBFN according to the weight ω_i and calculating the network error. Then the remaining unselected basis vectors are orthogonalized to facilitate the next selection.

Step 8. Defining a new convergence criterion $|\hat{y}_i-y_i|<|y_i|\times10^{-h}$. In this paper, *h* generally takes 4~7. If the error between each network predictor and actual value at the sample point satisfies the given condition, the network training is completed; otherwise, return to Step 5, and select the base vector with the largest influence factor in the remaining base vector.

Step 9. If the error requirements are not met after all the base vectors have been selected, increase the number of sample points and return to Step 2.

Flow chart of the proposed BRF-HLA is shown in Fig.3.

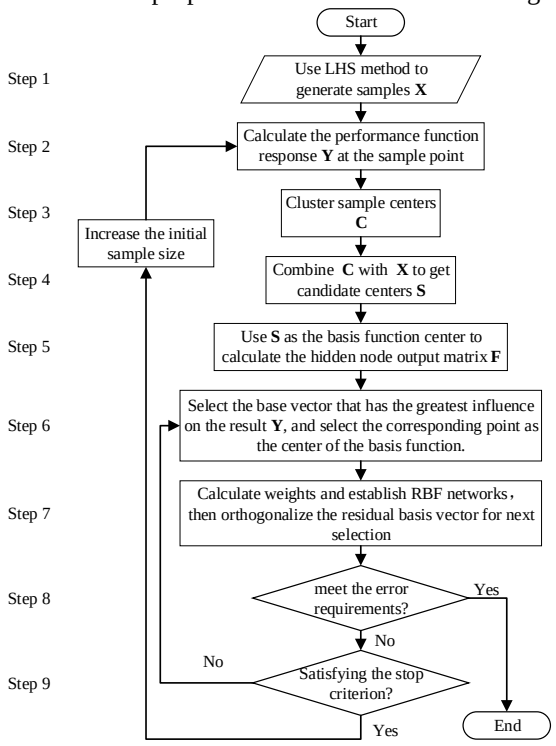


Fig.3 Flow chart of RBF-HLA.

IV. Numerical Examples

A. Example 1

Consider a highly non-liner limit state function (LSF) [45, 46]:

$$g(X)=X_1^3+X_1^2X_2+X_2^3-18 \tag{14}$$

where X_1 and X_2 are two random variables, and the distribution of variables is shown in Table 2.

TABLE II
Statistics of the Random Variables for Example 1

Variables	Means	Standard deviation	Distribution
X_1	10.0	5.0	Normal
X_2	9.9	5.0	Normal

In this example, the RBFN trained by the hybrid learning algorithm is compared with FORM, SORM. The calculation results of RBFN trained by clustering algorithms and orthogonal least squares learning algorithm are also used for comparison. The results obtained by Monte Carlo simulation (MCS) are used as benchmarks. Table 3 shows the results of the comparison. It can be seen that the RBFN trained by the hybrid learning algorithm training has higher accuracy and efficiency than the RBFN trained by the clustering algorithm and orthogonal least squares learning algorithm. ~~and~~ FORM and SORM are not applicable to highly nonlinear problems. In this paper, non-standard normal distribution variables are transformed into standard normal distributions for calculation and drawing by Rosenblatt transformation, and the formula is as follows,

$$U_i=\Phi^{-1}\left[F_{X_i}(X_i)\right],\ i=1,2,\cdots,n \tag{15}$$

where F_{X_i} is the cumulative distribution function of X_i and Φ^{-1} is the inverse of the normal distribution function.

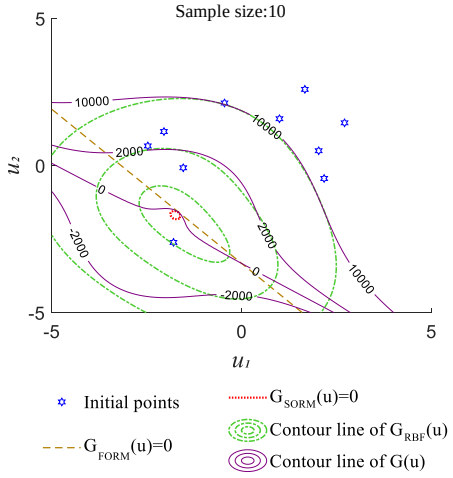
TABLE III
Comparison of the Result in Example 1

Methods	Failure probability	Relative error (%)	Number of LSF evaluations
FORM	1.077×10^{-2}	83.602	48
SORM	4.326×10^{-3}	26.284	53
RBF(*k*-means)	5.600×10^{-3}	3.260	36
RBF(OLS)	5.948×10^{-3}	2.747	27
RBF-HLA	5.782×10^{-3}	0.121	25
MCS	5.789×10^{-3}	0.000	1×10^{6}

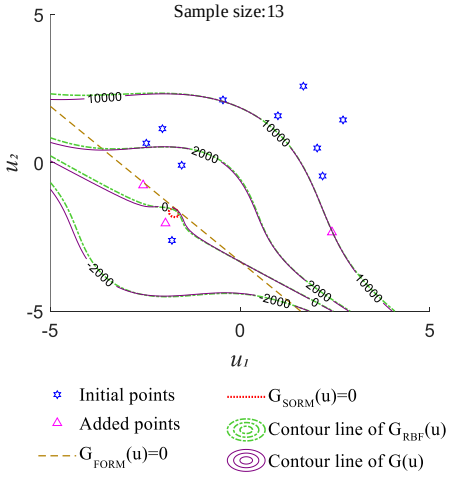
To better illustrate the process of RBF-HLA fitting the limit state function in Example 1, Fig. 4 shows the RBF-HLA fitting effect in the iterative process. In this example, the initial sample points are selected as 10. It can be seen from Fig.4 that as the number of sample points increases, the RBF-HLA gradually approaches the limit state function. When the number of sample points reaches 25, the fitting accuracy requirement is satisfied.

The core of the RBF-HLA method is to find the appropriate hidden layer center. Only by finding a suitable center can we achieve a better fitting effect with the least hidden layer nodes in the network, which helps to reduce the network structure and improve convergence speed. Fig. 5 shows the relationship between the number of hidden layer centers and the training error *err* in the iterative process. Here, the training error *err*

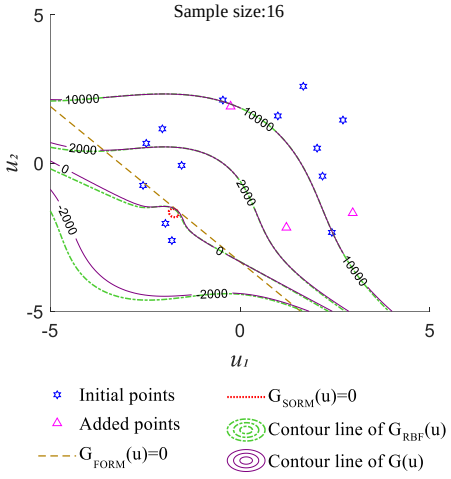
refers to $err = \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$, where n is the number of the samples, $\hat{y}_i$ represents the predicted value corresponding to the ith sample point, and y_i represents the actual response of the ith sample point. The number of sample points used in Fig. 5 is in one-to-one correspondence with Fig. 4. It is not difficult to see that as the number of hidden layer nodes increases, the training error decreases. In the iterative process, the change in failure probability is shown in Fig. 6. When the number of sample points reaches 25, the failure probability results are relatively stable and converge to the MCS results. In Fig. 5(e), although the number of sample points is 22, since there are cluster centers as the basis function center, the number of hidden layer nodes reaches 25 when the training error requirement is satisfied; in Fig 5(f), since the number of sample points is increased to 25, there will be more suitable sample points or cluster centers as the basis function center. At the end of training, therefore, the number of hidden nodes is only 23. The reduction in the number of hidden nodes leads to a streamlined network structure.



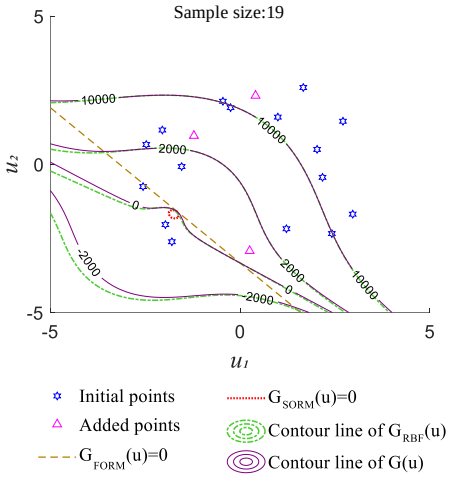
(a)



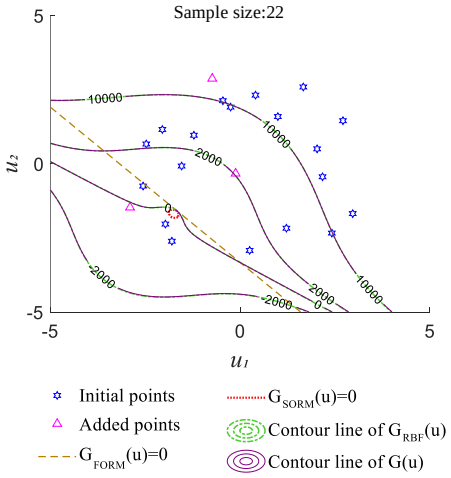
(b)



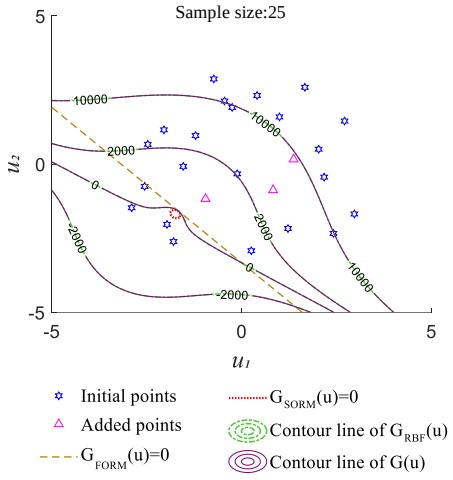
(c)



(d)

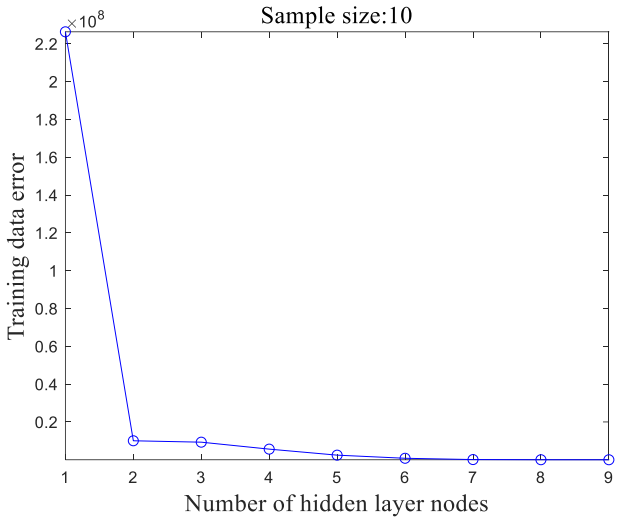


(e)

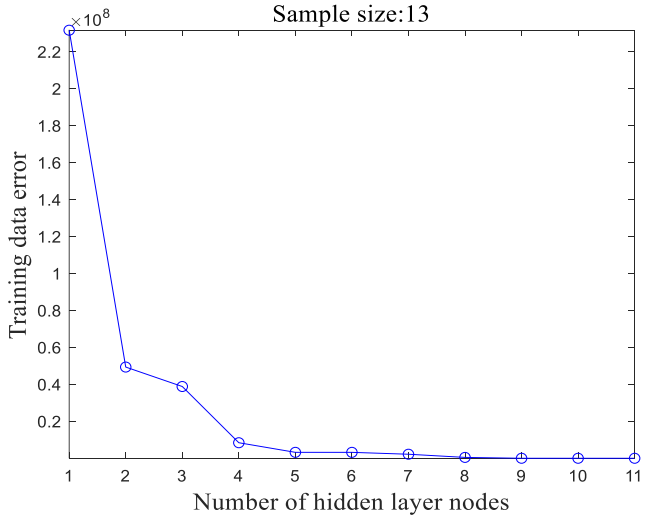


(f)

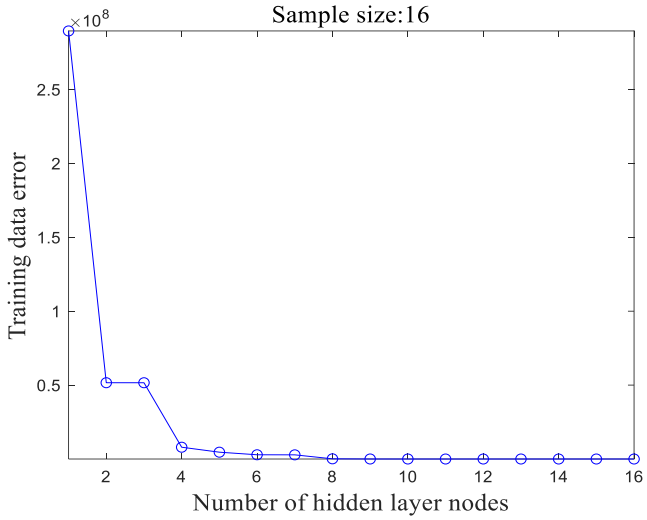
Fig. 4 The fitting process of RBF-HLA in Example 1.



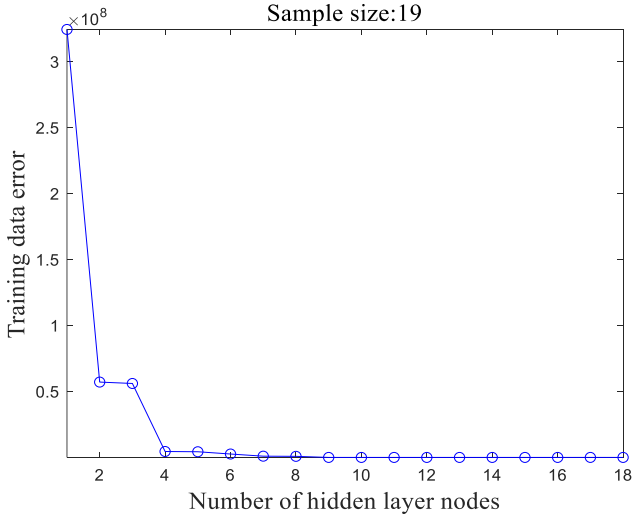
(a)



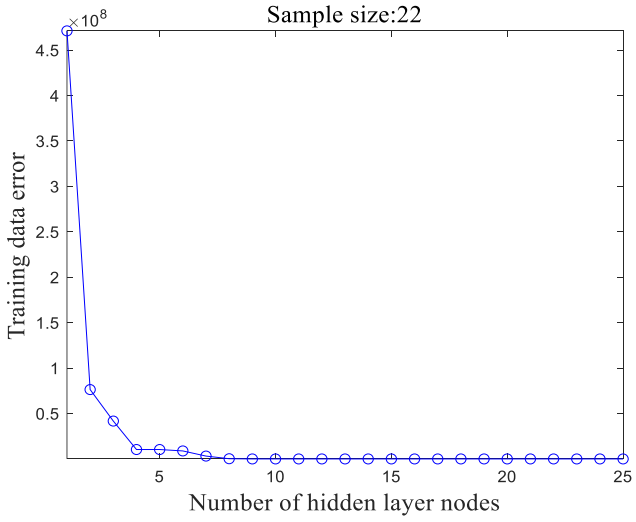
(b)



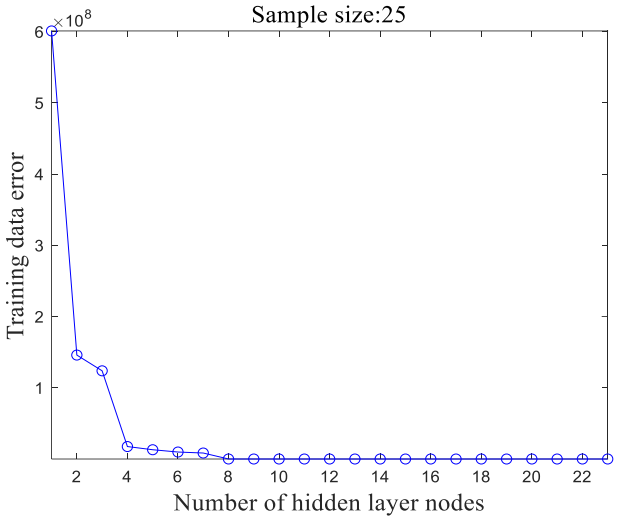
(c)



(d)



(e)



(f)

Fig.5 Relationship between the number of hidden layer nodes and training error in Example 1.

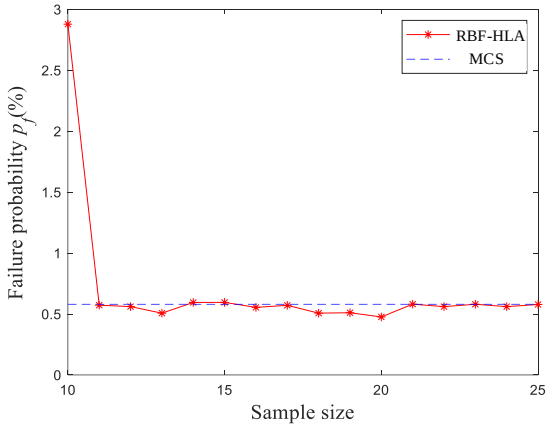


Fig.6 Iterative process of P_f in Example 1.

B. Example 2

To better illustrate the effectiveness of the proposed method, this example increases the variable dimension to three dimensions. The LSF is given by

$$g(X) = X_3 - \sqrt{300X_1^2 + 1.92X_2^2} \tag{16}$$

where X_1, X_2 and X_3 are three random variables and the variable distribution is shown in Table 4. Table 5 gives the comparative results of different reliability methods to solve the problem of this example. It can be seen that the failure probability obtained by the RBF-HLA method is $P_f = 8.61\times10^{-4}$, and the number of LSF evaluations is 19. Compared with other methods including RBF(*k*-means) and RBF(OLS), the efficiency and accuracy of RBF-HLA are greatly improved. The uncertainty variable distribution in this example is no longer a normal distribution and the results show that RBF-HLA is still applicable to problems with uncertain parameters of other distribution types. Table 6 directly lists the coordinates of the initial sample points, the added sample points in the iterative process and the corresponding failure probability.

When the number of sample points reaches 19, the failure probability error obtained by the RBF-HLA fitting limit state function is less than 1%, which meets the requirements. The final number of hidden nodes in the network is 21, and the number of hidden nodes exceeds the number of samples. It can be seen that if only the sample points are used as the hidden layer center, the requirements might not be met, and additional sample points or center points need to be selected as the hidden layer center to meet the fitting requirements. For engineering problems, it often takes a lot of time to add additional sample points. By using the clustering method to obtain the centers, the sample information can be more fully utilized. At the end of the iteration, the relationship between the number of hidden nodes and the training data error is shown in Fig. 7. It is shown that the change in the probability of failure during the process of increasing the sample size in Fig. 8.

TABLE IV
STATISTICS OF THE RANDOM VARIABLES FOR EXAMPLE 2

Variables	Means	Standard deviation	Distribution
X_1	1.0	0.16	Lognormal
X_2	20	2	Gumbel
X_3	48	3	Weibull

TABLE V
COMPARISON OF THE RESULTS IN EXAMPLE 2

Methods	Failure probability	Relative error (%)	Number of LSF evaluations
FORM	7.9765×10^{-4}	7.645	88
SORM	1.8895×10^{-3}	121.253	92
RBF(*k*-means)	9.0200×10^{-4}	5.621	34
RBF(OLS)	8.3200×10^{-4}	2.580	25
RBF-HLA	8.6100×10^{-4}	0.820	19
MCS	8.5400×10^{-4}	0.000	1×10^{6}

TABLE VI
ITERATIVE PROCESS OF P_f IN EXAMPLE 2

Number of iterations	X_1	X_2	X_3	P_f(%)
0	1.1538	19.8952	46.8344	2.78×10^{-2}
	0.9466	19.1220	49.5422	
	0.9050	20.1079	48.8695	
	0.9724	21.0643	43.0932	
	0.7928	21.7930	47.5178	
	0.8795	19.4071	47.3978	
	1.3327	22.1732	51.3824	
1	0.6186	19.8633	46.8657	5.35×10^{-2}
2	1.1539	18.4451	49.9721	3.66×10^{-2}
3	0.8170	20.8878	50.8402	6.11×10^{-2}
4	0.9210	20.0441	49.3987	6.01×10^{-2}
5	1.2271	22.6487	50.1893	8.16×10^{-2}
6	0.8595	22.1437	50.8236	6.34×10^{-2}
7	1.0016	20.8567	49.4548	5.88×10^{-2}
8	1.0271	22.0514	48.95795	6.30×10^{-2}
9	1.0534	17.1836	48.6937	6.36×10^{-2}
10	1.0916	23.0148	51.8008	8.41×10^{-2}
11	0.9620	19.1588	47.7098	8.65×10^{-2}
12	1.1234	18.1454	44.8245	8.61×10^{-2}

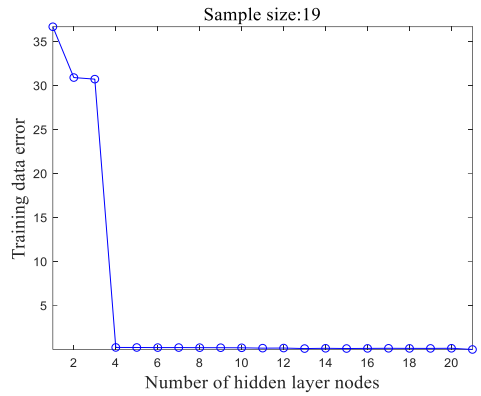


Fig.7 Relationship between the number of hidden layer nodes and training error in Example 2.

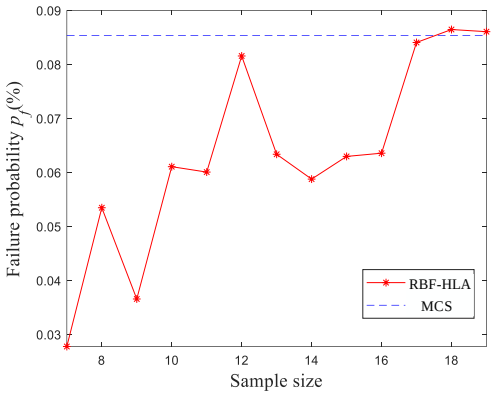


Fig.8 Iterative process of P_f in Example 2.

C. *Example 3*

This example consists of a nonlinear undamped single degree of freedom system with six random variables (Fig. 9) [37, 47, 48]. The LSF is given by

$$g(c_1,c_2,m,t_1,F_1) = 3r - \left|\frac{2F_1}{m\omega_0^2}\sin\left(\frac{\omega_0 t_1}{2}\right)\right| \qquad (17)$$

where $\omega_0 = \sqrt{(c_1+c_2)/m}$. The random variables are given in Table 7.

In this example, the initial sample points are selected as 13 and the number of the sample points at the end of the iteration is 21. The corresponding failure probability is $P_f = 2.851\times10^{-2}$, and the relative error compared with MCS results is only 0.243%. Table 8 gives the comparison results of different reliability methods to solve the problem of this example. The relationship between the number of hidden nodes and the training error is shown in Fig. 10. The number of hidden nodes is 14 at the end of the iteration, and the trend of failure probability in the iterative process is shown in Fig. 11. This example also shows that the RBF-HLA algorithm has strong applicability to high-dimensional problems. The accuracy and efficiency of neural network training based on hybrid learning algorithm is higher than that of single algorithm.

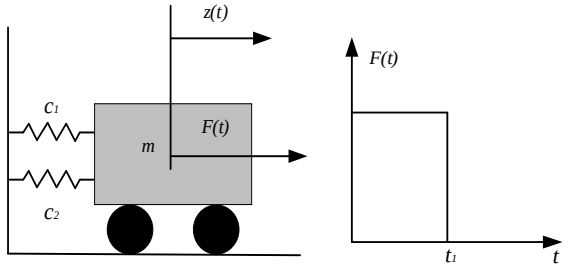


Fig. 9 Nonlinear oscillator.

TABLE VII
DISTRIBUTION OF RANDOM VARIABLES FOR THE NONLINEAR OSCILLATOR

Variables	Means	Standard deviation	Distribution type
m	1	0.05	Normal
c_1	1	0.1	Normal
c_2	0.1	0.01	Normal
r	0.5	0.05	Normal
F_1	1	0.2	Normal
t_1	1	0.2	Normal

TABLE VIII
COMPARISON OF THE RESULT OF A NONLINEAR OSCILLATOR

Methods	Failure probability	Relative error (%)	Number of LSF evaluations
FORM	3.108×10^{-2}	9.283	35
SORM	2.840×10^{-2}	0.141	48
RBF(k-means)	2.782×10^{-2}	2.180	35
RBF(OLS)	2.821×10^{-2}	0.798	21
RBF-HLA	2.851×10^{-2}	0.246	21
MCS	2.844×10^{-2}	0.000	1×10^{6}

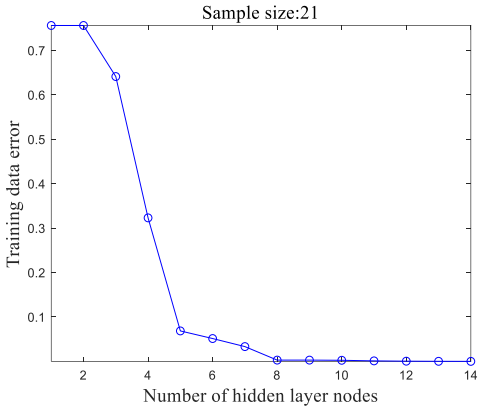


Fig.10 Relationship between the number of hidden layer nodes and training error for nonlinear oscillator.

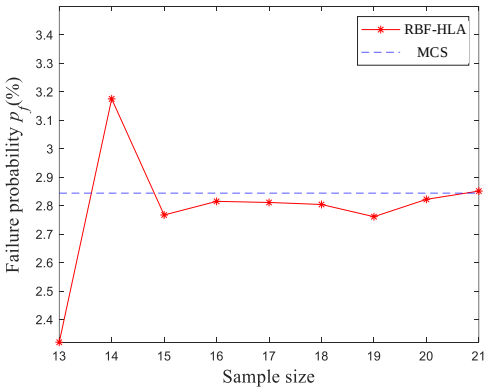


Fig.11 Iterative process of P_f for nonlinear oscillator.

D. *Example 4*

This example discusses the ultra-high-dimensional problem

[49]. The LSF is given by

$$g(x_1,\cdots,x_n)=\left(n+3\sigma\sqrt{n}\right)-\sum_{i=1}^{n}x_i \tag{18}$$

where n can be any value and n=40 in this example. x_i is a log-normally distributed variable with a mean of 1 and a standard deviation σ of 0.2. When dealing with the ultra-high dimensional problem, the failure probability of the RBF-HLA method analysis is the same as that obtained by the MCS, and the LSF is only called 12 times. The results of SS, AK-MCS and AK-SS are taken from reference [49]. It can be seen that RBF-HLA has certain advantages when dealing with high-dimensional problems. The results are shown in Table 9.

TABLE IX
COMPARISON OF THE RESULTS IN EXAMPLE 4

Methods	n	Failure probability	Relative error (%)	Number of LSF evaluations
SS[a]	40	1.813×10^{-3}	7.121	2800
AK[b]-MCS	40	1.852×10^{-3}	5.122	42
AK-SS	40	1.818×10^{-3}	6.864	42
RBF-HLA	40	1.952×10^{-3}	0.000	12
MCS	40	1.952×10^{-3}	0.000	1×10^{6}

[a] Subset Simulation; [b] Active Learning

E. Example 5

1) Introduction to robot kinematics

Industrial robots can replace humans with repetitive tasks efficiently, accurately, such as industrial processing, palletizing and welding. Industrial robots contain multiple links and rotating pairs, and the existence of uncertain parameters such as joint clearance and rod length machining error will make the actual position of the end-effector of the industrial robot deviate from the target position. When this offset exceeds a certain range, the positioning accuracy of the industrial robot is considered to be a failure. The kinematics reliability of an industrial robot refers to the probability that the end-effector reaches the specified moving path or trajectory within the error tolerance. Likewise, the positioning accuracy reliability discussed herein refers to the probability that an industrial robot end-effector reaches a specified position within the error tolerance. The reliability of positioning accuracy of industrial robots will seriously affect the quality of their work. Therefore, it is crucial to analyze the reliability of industrial robots positioning accuracy. This paper takes a six-axis industrial robot as an example to analyze the reliability of industrial robot positioning accuracy. Fig. 12 shows the three-dimensional model of a six-axis industrial robot. Fig. 13 shows the schematic diagram of the six-axis industrial robot mechanism.

Fig. 12 Three-dimensional model of a six-axis industrial robot.

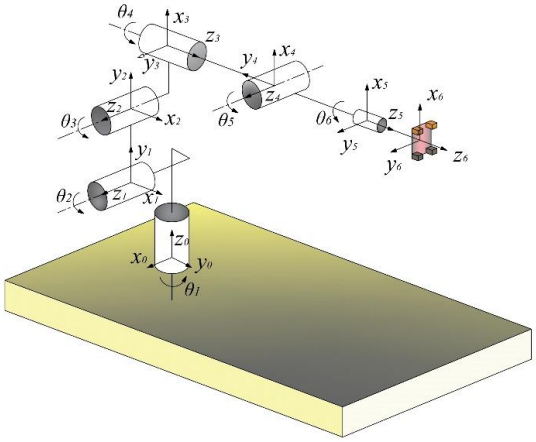


Fig. 13 The mechanism diagram of the six-axis industrial robot.

To carry out the reliability analysis of industrial robot positioning accuracy, a kinematics model of industrial robots must be established first. The Denavit-Hartenberg (D-H) matrix [50] describes the kinematic equations of the industrial robot including position and orientation. The specific form of the D-H matrix is as follows,

$$\mathbf{T}_{i-1}^{i}=\begin{bmatrix}\cos(\theta_i) & -\cos(\psi_i)\sin(\theta_i) & \sin(\psi_i)\cos(\theta_i) & a_i\cos(\theta_i)\\ \sin(\theta_i) & \cos(\psi_i)\cos(\theta_i) & -\sin(\psi_i)\cos(\theta_i) & a_i\sin(\theta_i)\\ 0 & \sin(\psi_i) & \cos(\psi_i) & d_i\\ 0 & 0 & 0 & 1\end{bmatrix} \tag{19}$$

where a_i, d_i, ψ_i and θ_i are used to define the coordinates of the ith link, as shown in Fig. 14. The position of the end-effector can be calculated according to Eq. (19),

$$\mathbf{T}_{end}=\mathbf{T}_0^1\times\mathbf{T}_1^2\times\cdots\times\mathbf{T}_{l-1}^{l}=\begin{bmatrix}\mathbf{n} & \mathbf{s} & \mathbf{a} & \mathbf{p}\\ 0 & 0 & 0 & 1\end{bmatrix} \tag{20}$$

where **n**, **s** and **a** represent the direction cosines of the fixed coordinate system established on the end-effector relative to the fixed coordinate system at the base, and **p** represents the positional coordinates of the end-effector in the fixed coordinate system.

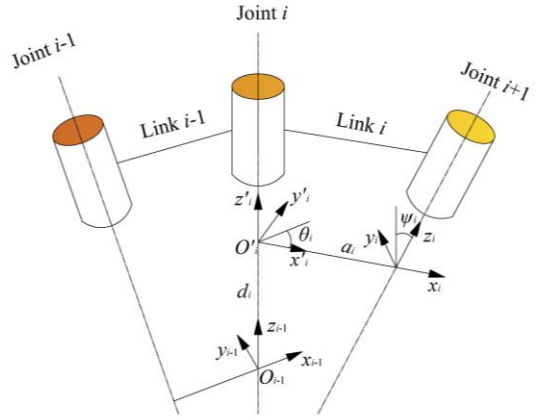


Fig. 14 Denavit-Hartenberg kinematic parameter.

2) Positioning accuracy reliability analysis

Assume that the *i*th target position of the industrial robot end-effector is (x_{oi}, y_{oi}, z_{oi}), and the actual position of the end-effector is (x_{ri}, y_{ri}, z_{ri}). Due to the existence of uncertain parameters such as joint clearance and joint torsion angle, the industrial robot positioning error can be defined as follows.

①single coordinate direction positioning error:

$$E_{xi} = |x_{oi} - x_{ri}| \tag{21}$$

$$E_{yi} = |y_{oi} - y_{ri}| \tag{22}$$

$$E_{zi} = |z_{oi} - z_{ri}| \tag{23}$$

where E_{xi}, E_{yi}, and E_{zi} represent the positioning errors in the *x*, *y* and *z* coordinate directions, respectively.

② single point positioning error:

$$E_{pi} = \sqrt{(x_{ri} - x_{oi})^2 + (y_{ri} - y_{oi})^2 + (z_{ri} - z_{oi})^2} \tag{24}$$

where E_{pi} denotes the single point positioning error which indicates the distance between the actual position and the nominal position. The positioning accuracy of the industrial robots is reliable when the positioning error meets the requirements. The performance function of the end-effector positioning is defined as

$$g(x,y,z) = e - E_i \tag{25}$$

where E_i is a single coordinate or single point positioning error and e is the failure threshold. When $g(x,y,z) < 0$, the positioning accuracy of industrial robots is failed. When $g(x,y,z) > 0$, the positioning accuracy of industrial robots is reliable.

According to the above performance function, the reliability analysis of industrial robot positioning accuracy is carried out. In this paper, the actual position of the end-effector affected by the uncertain variable is obtained by Matlab robot toolbox [51]. According to the three-dimensional model of the industrial robot shown in Fig. 12, the D-H parameters of the industrial robot are set as shown in Table 10, and the distribution of each uncertain variable is shown in Table 11.

TABLE X
PARAMETERS OF D-H MATRIX FOR AN ELBOW MANIPULATOR

Joint No.	a_i(mm)	d_i(mm)	α_i (°)	θ_i(°)
1	200	345	90	-20
2	400	0	0	90
3	175	0	90	30
4	0	750	-90	0
5	0	0	90	-45
6	0	200	0	0

TABLE XI
PROBABILITY DISTRIBUTIONS OF RANDOM VARIABLES

Variables	Means	Standard deviation	Distribution type
a_1(mm)	200	0.200	Normal
a_2(mm)	400	0.400	Normal
a_3(mm)	175	0.175	Normal
d_1(mm)	345	0.345	Normal
d_4(mm)	750	0.750	Normal
d_6(mm)	200	0.200	Normal
ε_1 (°)	0	0.1	Normal
ε_5 (°)	0	0.1	Normal
λ_1 (°)	0	0.1	Normal

In Table 11, ε_1 and ε_5 are random variables representing the uncertainty of joint angles θ_1 and θ_5. This uncertainty is derived from the effects of joint space. The detailed derivation process can be found in the literature [17]. λ_1 is a random variable representing the uncertainty of the joint torsion angle of the joint 1. The uncertainty comes from the influence of joint installation error or gravity.

Figure 15 shows the simulation model obtained after input of uncertain variables. According to different uncertain variable inputs, a series of actual positions of the industrial robot end-effector can be obtained from simulation. In this paper, the number of simulations is set as 2×10^5, and the total simulation time is 31,255 seconds, and 2×10^5 sets of uncertain parameters and their corresponding end-effector coordinate positions are obtained as shown in Fig. 16. The sphere represents the positioning failure threshold of the industrial robot.

Under different failure thresholds, the reliability of the positioning accuracy of the industrial robot is different, and the failure threshold is determined according to the actual working conditions of the industrial robot. Due to the low efficiency of the simulation, the radial basis network is used to solve the problem of reliability analysis of positioning accuracy. By selecting a small number of uncertain parameters generated in the simulation and their corresponding end-effector coordinate positions, the network fitting is carried out, and the sample points are added until the network accuracy meets the requirements. Finally, when the number of sample points reaches 14, the convergence condition for the RBFN is satisfied, and the iteration stops. The predicted coordinate position of the end-effector of the industrial robot is as shown in Fig. 17.

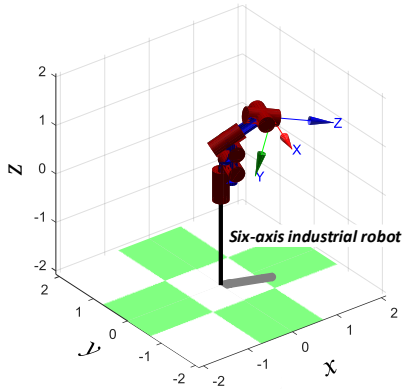


Fig. 15 Simulation model of industrial robot.

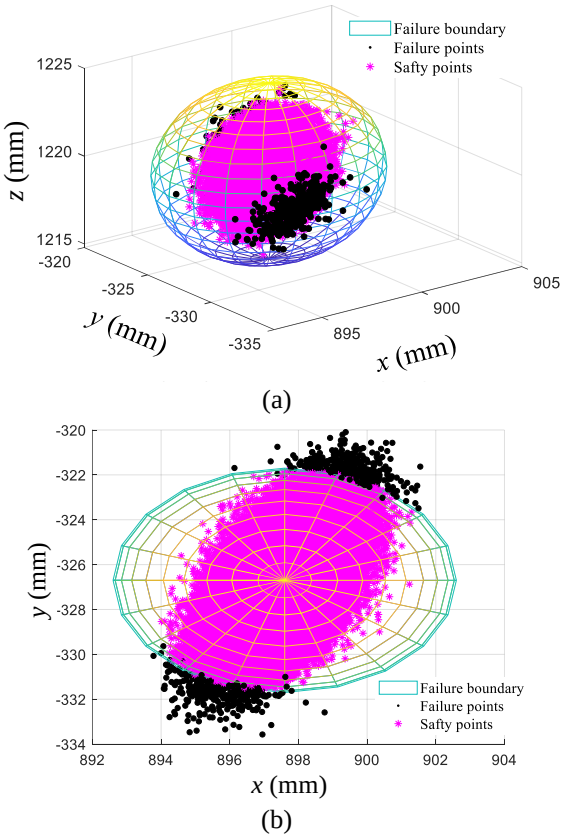
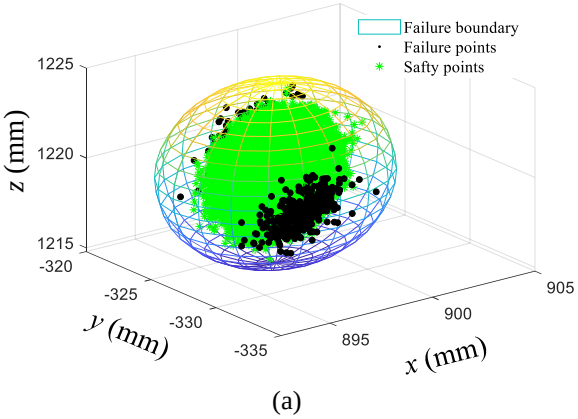


Fig. 16 End position coordinates obtained by Matlab simulation.



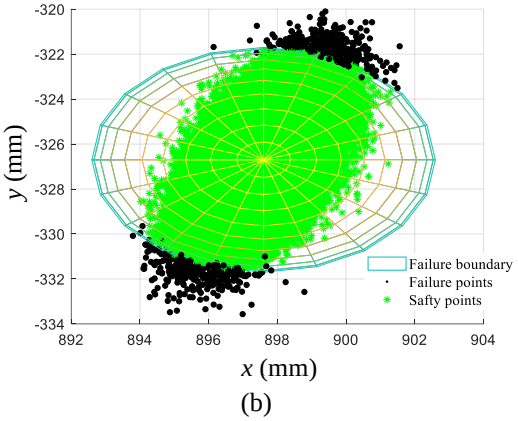
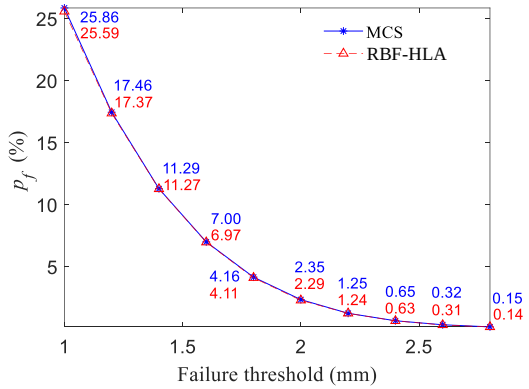
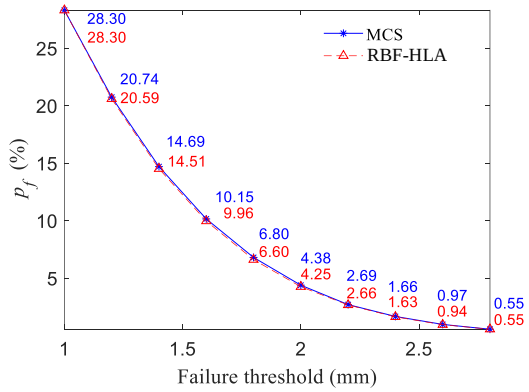


Fig.17 End position coordinates predicted by the RBF-HLA.

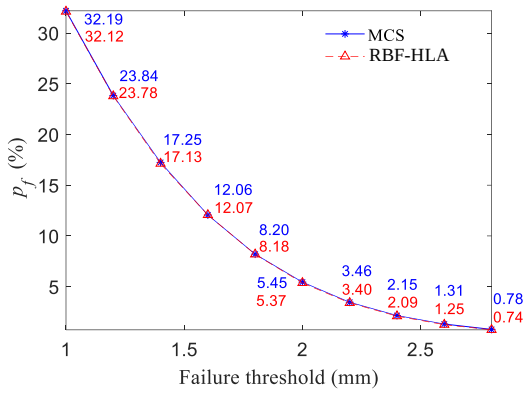
Fig. 16(a) and Fig. 17(a) are three-dimensional views of the end position points respectively and Fig. 16(b) and Fig.17(b) are projection views on the *x*-*y* plane respectively. It can be seen that the predicted end position points are consistent with the overall shape of the simulated end position points, which is roughly an ellipsoid. In order to verify the accuracy of the RBF prediction results, the failure probability of the end points obtained by the two methods under different failure thresholds is calculated, and the single-coordinate and single-point failure probability are compared. The results are shown in Fig. 18.



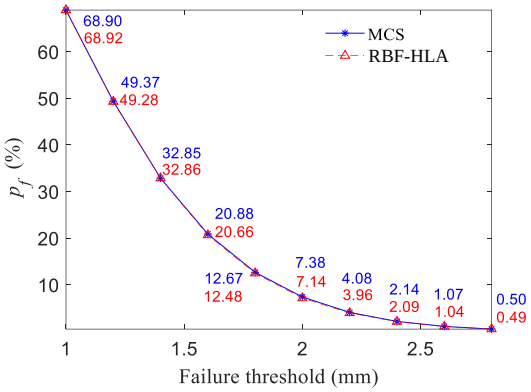
(a) Failure probability in the *x*-direction.



(b) Failure probability in the *y*-direction.



(c) Failure probability in the *z*-direction.



(d) Failure probability of single points.

Fig. 18 Failure probability under different failure thresholds.

Fig. 18(a), (b) and (c) show the failure probability in the *x*, *y* and *z* coordinate directions at different failure thresholds, respectively, and Fig. 18(d) shows the failure probability of a single point with different failure thresholds. It can be seen that under the same failure threshold, the failure probability of single point and single coordinate obtained by RBF prediction is very close to the result obtained by MCS simulation. It takes a long time to analyze the positioning reliability of industrial robots through simulation, but the proposed method only needs running the simulation for 14 times. The RBFN can be built based on simulation results to predict the position of the remaining points, which greatly improves the computational efficiency without losing accuracy.

V. Conclusion

In this paper, a hybrid learning algorithm for training radial basis neural network is proposed by integrating two learning algorithms. The proposed method is able to not only select the centers of the basis function according to its importance but also cluster more reasonable centers. Through four mathematical examples, the method is proven effective to deal with the high-dimension problems with uncertain parameters of different distribution types. Then the proposed method is applied to reliability analysis of positioning accuracy for industrial robots. The results show that under the same failure threshold, the failure probability obtained from the proposed method is very close to that from the Monte Carlo simulation. More importantly, the proposed method only needs 14 sample points, indicating the computational efficiency is considerably improved while the accuracy is maintained. In future research, the accuracy of the RBFN near the most probable point will be improved to reduce the evaluations of LSF. When the cluster centers and the sample points are selected as the center of the RBF, the vector in radial basis matrix may have a linear correlation which will affect the performance of the network. Thus, the linear correlation in basis vectors should be excluded in the future research.

Acknowledgment

The authors would like to greatly appreciate the financial support provided by the National Key R&D Program of China (Grant No. 2017YFB1301300), the National Natural Science Foundation of China (Grant No. 51905146), the Key R&D Plan Program of Hebei Province (Grant No. 19211808D) and the Research Program of Education Bureau of Hebei Province (Grant No. QN2019141).

References

[1] M. Hägele, K. Nilsson, and J. N. Pires, "Industrial robotics". Springer handbook of robotics, pp. 963-986, 2008.
[2] S. Yan and P. Guo, "Kinematic accuracy analysis of flexible mechanisms with uncertain link lengths and joint clearances". P. I. Mech. Eng. C-J. M. E.C., vol. 225, no. 8, pp. 1973-1983, 2011.
[3] J. Wang, J. Zhang, and X. Du, "Hybrid dimension reduction for mechanism reliability analysis with random joint clearances". Mech. Mach. Theory, vol. 46, no. 10, pp. 1396-1410, 2011.
[4] D. Zhang, X. Han, C. Jiang, J. Liu, and Q. Li, "Time-dependent reliability analysis through response surface method". J. Mech. Design, vol. 139, no. 4, 2017.
[5] S. Yu, Z. Wang, and D. Meng, "Time-variant reliability assessment for multiple failure modes and temporal parameters". Struct. Multidiscip. O., vol. 58, no. 4, pp. 1705-1717, 2018.
[6] B. N. Jha and H.-s. Li, "Structural reliability analysis using a hybrid HDMR-ANN method". Journal of Central South University, vol. 24, no. 11, pp. 2532-2541, 2017.
[7] Z. Huang, T. Yang, and F. Li, "A decoupling algorithm with first-order asymptotic integration for reliability-based design optimization". Advances in Mechanical Engineering, vol. 10, no. 9, p. 1687814018793336, 2018.
[8] J. Kim, W.-J. Song, and B.-S. Kang, "Stochastic approach to kinematic reliability of open-loop mechanism with dimensional tolerance". Appl. Math. Model., vol. 34, no. 5, pp. 1225-1237, 2010.
[9] D. Sun and G. Chen, "Kinematic accuracy analysis of planar mechanisms with clearance involving random and epistemic uncertainty". Eur. J. Mech. ASolid., vol. 58, pp. 256-261, 2016.
[10] P. Bhatti and S. Rao, "Reliability analysis of robot manipulators". J. Mech.,Trans., and Automation, vol. 110, no. 2, pp. 175-181, 1988.
[11] S. Rao and P. Bhatti, "Probabilistic approach to manipulator kinematics and dynamics". Reliab. Eng. Syst. Safe., vol. 72, no. 1, pp. 47-58, 2001.
[12] G. Cui, H. Zhang, D. Zhang, and F. Xu, "Analysis of the kinematic accuracy reliability of a 3-DOF parallel robot manipulator". Int. J. Adv. Robot. Syst., vol. 12, no. 2, p. 15, 2015.
[13] A. Chaker, A. Mlika, M. Laribi, L. Romdhane, and S. Zeghloul, "Clearance and manufacturing errors' effects on the accuracy of the 3-RCC Spherical Parallel Manipulator". Eur. J. Mech. A-Solid., vol. 37, pp. 86-95, 2013.
[14] J. Zhu and K.-L. Ting, "Uncertainty analysis of planar and spatial robots with joint clearances". Mech. Mach. Theory., vol. 35, no. 9, pp. 1239-1256, 2000.
[15] W. Wu and S. S. Rao, "Uncertainty analysis and allocation of joint tolerances in robot manipulators based on interval analysis". Reliab. Eng. Syst. Safe., vol. 92, no. 1, pp. 54-64, 2007.
[16] J. Wu, D. Zhang, J. Liu, X. Jia, and X. Han, "A computational framework of kinematic accuracy reliability analysis for industrial robots". APPL. MATH. MODEL, 2020.
[17] M. D. Pandey and X. Zhang, "System reliability analysis of the robotic manipulator with random joint clearances". Mech. Mach. Theory., vol. 58, pp. 137-152, 2012.

[18] J. Wu, D. Zhang, J. Liu, and X. Han, "A Moment Approach to Positioning Accuracy Reliability Analysis for Industrial Robots". IEEE T. Reliab., 2019.
[19] D. Zhang and X. Han, "Kinematic reliability analysis of robotic manipulator". J. Mech. Design, vol. 142, no. 4, 2020.
[20] S. Wu and T. W. Chow, "Induction machine fault detection using SOM-based RBF neural networks". IEEE T. Ind. Electron., vol. 51, no. 1, pp. 183-194, 2004.
[21] T. Kondo and J. Ueno, "Medical image recognition of abdominal multiorgans by RBF GMDH-type neural network". Int. J. Innov. Comput. I., vol. 5, no. 1, pp. 225-240,2009..
[22] K. Xu, M. Xie, L. C. Tang, and S. Ho, "Application of neural networks in forecasting engine systems reliability". Appl. Soft. Comput., vol. 2, no. 4, pp. 255-268, 2003.
[23] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units". Neural. Comput., vol. 1, no. 2, pp. 281-294, 1989.
[24] Y. Engel, S. Mannor, and R. Meir, "The kernel recursive least-squares algorithm". IEEE T. Signal Proces., vol. 52, no. 8, pp. 2275-2285, 2004.
[25] S. Chen, C. F. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks". IEEE T NEURAL NETWOR, vol. 2, no. 2, pp. 302-309, 1991.
[26] A. Alexandridis, E. Chondrodima, N. Giannopoulos, and H. Sarimveis, "A fast and efficient method for training categorical radial basis function networks". IEEE T. Neur. Net. Lear., vol. 28, no. 11, pp. 2831-2836, 2016.
[27] H. Wang, R. Feng, Z.-F. Han, and C.-S. Leung, "ADMM-based algorithm for training fault tolerant RBF networks and selecting centers". IEEE T. Neur. Net. Lear., vol. 29, no. 8, pp. 3870-3878, 2017.
[28] H.-G. Han, W. Lu, Y. Hou, and J.-F. Qiao, "An adaptive-PSO-based self-organizing RBF neural network". IEEE T. Neur. Net. Lear., vol. 29, no. 1, pp. 104-117, 2016.
[29] Z.-R. Lai, D.-Q. Dai, C.-X. Ren, and K.-K. Huang, "Radial basis functions with adaptive input and composite trend representation for portfolio selection". IEEE T. Neur. Net. Lear., vol. 29, no. 12, pp. 6214-6226, 2018.
[30] M. J. Er, S. Wu, J. Lu, and H. L. Toh, "Face recognition with radial basis function (RBF) neural networks". IEEE T NEURAL NETWOR., vol. 13, no. 3, pp. 697-710, 2002.
[31] J. Deng, D. Gu, X. Li, and Z. Q. Yue, "Structural reliability analysis for implicit performance functions using artificial neural network". Struct. Saf., vol. 27, no. 1, pp. 25-48, 2005.
[32] J. Deng, "Structural reliability analysis for implicit performance function using radial basis function network". Int. J. Solids Struct., vol. 43, no. 11-12, pp. 3255-3291, 2006.
[33] H. Dai, W. Zhao, W. Wang, and Z. Cao, "An improved radial basis function network for structural reliability analysis". J. Mech. Sci. Technol., vol. 25, no. 9, p. 2151, 2011.
[34] A. Kavousifard and H. Samet, "Consideration effect of uncertainty in power system reliability indices using radial basis function network and fuzzy logic theory". Neurocomputing, vol. 74, no. 17, pp. 3420-3427, 2011.
[35] X.-h. Tan, W.-h. Bi, X.-l. Hou, and W. Wang, "Reliability analysis using radial basis function networks and support vector machines". Comput. Geotech., vol. 38, no. 2, pp. 178-186, 2011.
[36] M. Chau, X. Han, Y. Bai, and C. Jiang, "A structural reliability analysis method based on radial basis function". Comput. Mater. Con., vol. 27, no. 2, p. 128, 2012.
[37] X. Li, C. Gong, L. Gu, W. Gao, Z. Jing, and H. Su, "A sequential surrogate method for reliability analysis based on radial basis function". Struct. Saf., vol. 73, pp. 42-53, 2018.
[38] X. Li, C. Gong, L. Gu, Z. Jing, H. Fang, and R. Gao, "A reliability-based optimization method using sequential surrogate model and Monte Carlo simulation". Struct. Multidiscip. O., pp. 1-22, 2018.
[39] X. Du, "Unified uncertainty analysis by the first order reliability method". J. Mech. Design, vol. 130, no. 9, p. 091401, 2008.
[40] J. Zhang and X. Du, "A second-order reliability method with first-order efficiency". J. Mech. Design, vol. 132, no. 10, p. 101006, 2010.
[41] Y.-G. Zhao and T. Ono, "A general procedure for first/second-order reliabilitymethod (FORM/SORM)". Struct. Saf., vol. 21, no. 2, pp. 95-112, 1999.
[42] G. E. Fasshauer and J. G. Zhang, "On choosing “optimal” shape parameters for RBF approximation". Numer. Algorithms, vol. 45, no. 1-4, pp. 345-368, 2007.
[43] H. Moradkhani, K.-l. Hsu, H.V. Gupta and S. Sorooshian, "Improved streamflow forecasting using self-organizing radial basis function artificial neural networks". J. Hydrol., vol. 295, no. 1-4, pp. 246-262,2004.
[44] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation". IEEE T. Pattern Anal., no. 7, pp. 881-892, 2002.
[45] J. Cheng, Q. Li, and R.-c. Xiao, "A new artificial neural network-based response surface method for structural reliability analysis". Probabilist. Eng. Mech., vol. 23, no. 1, pp. 51-63, 2008.
[46] I. Kaymaz, "Application of kriging method to structural reliability problems". Struct. Saf., vol. 27, no. 2, pp. 133-151, 2005.
[47] M. R. Rajashekhar and B. R. Ellingwood, "A new look at the response surface approach for reliability analysis". Struct. Saf., vol. 12, no. 3, pp. 205-220, 1993.
[48] L. Schueremans and D. Van Gemert, "Benefit of splines and neural networks in simulation based structural reliability analysis". Struct. Saf., vol. 27, no. 3, pp. 246-261, 2005.
[49] X. Huang, J. Chen, and H. Zhu, "Assessing small failure probabilities by AK–SS: an active learning method combining Kriging and subset simulation". Struct. Saf., vol. 59, pp. 86-95, 2016.
[50] C. Rocha, C. Tonetto, and A. Dias, "A comparison between the Denavit–Hartenberg and the screw-based methods used in kinematic modeling of robot manipulators". Robot. Cim-Int Manuf., vol. 27, no. 4, pp. 723-728, 2011.
[51] P. I. Corke, "A robotics toolbox for MATLAB". IEEE Robot. Autom. Mag., vol. 3, no. 1, pp. 24-32, 1996.