sensors

MDPI

Article

A Parametric Design Method for Optimal Quick Diagnostic Software

Xiao-jian Yi [1,2,3] and Peng Hou [1,*]

1 School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China; yixiaojian@amss.ac.cn
2 Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China
3 Department of Overall Technology, China North Vehicle Research Institute, Beijing 100072, China
* Correspondence: 3120160129@bit.edu.cn

Received: 17 December 2018; Accepted: 11 February 2019; Published: 21 February 2019

Abstract: Fault diagnostic software is required to respond to faults as early as possible in time-critical applications. However, the existing methods based on early diagnosis are not adequate. First, there is no common standard to quantify the response time of a fault diagnostic software to the fault. Second, none of these methods take into account how the objective to improve the response time may affect the accuracy of the designed fault diagnostic software. In this work, a measure of the response time is provided, which was formulated using the time complexity of the algorithm and the signal acquisition time. Model optimization was built into the designed method. Its objective was to minimize the response time. The constraint of the method is to guarantee diagnostic accuracy to no less than the required accuracy. An improved feature selection method was used to solve the optimization modeling. After that, the design parameter of the optimal quick diagnostic software was obtained. Finally, the parametric design method was evaluated with two sets of experiments based on real-world bearing vibration data. The results demonstrated that optimal quick diagnostic software with a pre-defined accuracy could be obtained through the parametric design method.

Keywords: early diagnosis; time-critical application; early classification; feature selection

1. Introduction

In time-critical applications, fault diagnostic software is required to respond to a fault as early as possible. Based on the timely diagnostic result, operators can prevent the aggravation of consequences over time, such as the risks associated with crew member exposure to toxic or dangerous chemicals [1], the wrong operation of medical robots [2], error in numerical control processing technology [3], and fracture and crack arrest in ship structures [4].

To achieve this aim, many methods have been applied to enable diagnostic software to respond to faults as early as possible. Those methods are divided into two categories. The first is based on signal processing technologies such as wavelet transform [5,6], empirical mode decomposition (EMD) [7], spectral kurtosis (SK) [8], envelope spectrum [9,10], and others [11–13]. These methods assume that the fault feature changes from weak to strong with the development of the fault. By extracting the weak feature of faults in the frequency domain or the time–frequency domain, a diagnostic result can be obtained in the early stage of fault development, and the objective of receiving a response as early as possible is achieved. Due to the weak nature of early fault features, weak fault features are often submerged in the noisy background, which increases the difficulty of feature recognition. To address this challenge, those methods apply noise-reduction methods before the recognition of weak features [14]. Although useful weak features are extracted, the feature information is also weakened in the process of noise reduction [15], which means that the performance of the fault diagnostic software is degraded. To avoid this situation, the stochastic resonance (SR) method was applied [16–19]. However, there are still some difficulties in practical applications. Firstly, the difficulty of feature recognition is not reduced by the noise-reduction method and stochastic resonance (SR) method. Secondly, even if the noise is reduced successfully, it is hard to know what the corresponding weak fault features are in the early stage of faults without enough knowledge of the fault mechanism. Additionally, the aforementioned methods cannot provide an a priori estimate on the diagnostic accuracy. Fortunately, through the combination of statistical analysis and signal processing technologies, statistical spectral analysis has been recently employed to improve this situation [20,21].

Another effective way is to increase the response rate of the diagnostic software by shortening the collection time of signals. Essentially, this kind of method speeds up the process of fault diagnosis in the software. In these methods, time series classification in machine learning is applied to shorten the collection time of signals [22–25]. As such, the fault diagnostic software is actually a classifier, which is trained to classify on as short as possible a prefix of the specified signal size. When the trained classifier is deployed in service, there is no need to wait for the whole specified signal size to be collected. Therefore, the diagnostic software can respond to the fault quickly. Compared with the approach based on signal processing technologies, these methods focus on the differences between the normal signal and the fault signal that emerge in the time domain. They assume that part of the signal required by the diagnostic software is redundant or useless. In addition, they need no knowledge of fault mechanism for extracting fault features. However, the application of these methods is still limited due to the insufficient data available for training the classifier.

As stated previously, although many approaches have been proposed to obtain a response as early as possible, to the best of our knowledge, there is no widely available quantified measure to assess whether the response time is improved by such methods. These approaches assume that the response time obtained by the software can be objectively evaluated without taking into account the effect caused by the hardware. Furthermore, it is assumed that the response time can be quantitatively assessed, which means different approaches can be evaluated against a common standard, and a reference can be provided in software design. On the other hand, a design to respond to faults as early as possible may be achieved at the expense of the accuracy of the fault diagnostic software. As mentioned in [26], a fault diagnostic system designed to respond as early as possible is very sensitive to noise. This seriously weakens the fault tolerance of the fault diagnostic system. It follows that the requirement for fault diagnostic software to respond as early as possible may lead to low accuracy. Therefore, there is a trade-off between the objective response time and the accuracy. In fact, it is pointless to require fault diagnostic software to respond as early as possible without considering the accuracy of its response. On the other hand, if the fault diagnostic software is only required to diagnose accurately without consideration for the response speed, the diagnostic result is more likely to be a "death certificate" [27]. These challenges motivated us to present a measure of the response time and to provide a novel design method based on a machine learning method to design an optimal diagnostic software based on this measure.

The contribution of this paper is two-fold:

(1) In order to carry out the parametric design of an optimal quick diagnostic software for time-critical applications, a measure of the response time is proposed. By combining the time complexity of the algorithm with the calculation formula of the signal acquisition time, we can quantitatively and rationally evaluate whether the response time is improved.

(2) We present a parametric design method for the optimal quick diagnostic software. This method adopts time series classification in machine learning as the diagnostic algorithm. The objective is to minimize the measure of the response time, and the constraint is to guarantee that the accuracy of the diagnostic software is no less than a pre-defined accuracy. An improved wrapper method is used to solve the optimization model in order to acquire the design parameters.

The remainder of the current paper is organized as follows. Section 1 elaborates the measure of the response time, which is adopted in our proposed design method. Section 2 details the steps in the parametric design method. Section 3 illustrates the application of the parametric design method through two experiments. In Section 4, the experimental results are detailed along with discussions. Finally, Section 5 concludes this paper.

2. The Measure of the Response Time

The response time is actually the time lag Δt between the time when the fault occurs and the time when the fault diagnostic software responds. It is mainly composed of the signal acquisition time and the execution time of the diagnostic algorithm in the diagnostic software. The signal acquisition time t_A is actually the waiting time for the diagnostic software to acquire a specified signal size (the transmission delay is small enough to be negligible). The specified size for the input signal is related to the methods used in the diagnostic algorithm. In this way, enough signal information can be acquired by the diagnostic algorithm, rendering it able to execute fault diagnosis. Theoretically, the waiting time can be represented by $t_\mathrm{A} = \frac{n}{f}$. The sampling number n quantifies the signal size. In most cases, the sampling frequency f is given ahead of most design parameters, thus the sampling number n is the only factor that determines t_A.

The time of the diagnostic algorithm execution t_B can be estimated by the time complexity. This is generally expressed as a function of the size of the input [28]. Using the time complexity, the execution time t_B can be represented as $t_\mathrm{B} = O(m)$, where m is the size of the algorithm input (the size of the input signal). Equivalently, the size of the algorithm input can be replaced by the sampling number n for the fault diagnostic algorithm, thus the expression can be represented by $t_\mathrm{B} = O[\mathrm{C}(n)]$, where C is related to the type of diagnostic algorithm. In general, for the different types of algorithms, $\mathrm{C}(n)$ typically includes n, $n * logn$, n^α, and 2^n. Therefore, the execution time of the diagnostic algorithm t_B is determined by the sampling number n and the algorithm type C.

When a diagnostic software is in service, the time lag Δt can be represented as $\Delta t = t_1 - t_0 \approx a(t_\mathrm{A} + t_\mathrm{B}) + bt_\mathrm{A}$, where $a \in \mathrm{N}^*$, $b \in [0, 1)$. t_0 is the time when the fault occurs, and t_1 is the response time of the fault diagnostic software. In our approach, a denotes the number of loops. A loop is a diagnostic procedure which is composed of the signal acquisition and the diagnostic algorithm execution. Due to the low robustness of the diagnostic software, the software should respond to the fault when there is severe interference or some sort of fault in the fault diagnostic software. If this situation takes place, the quantity a may not be 1 (as is described in Figure 1C,D). Generally, the diagnostic software in time-critical applications is assumed to have enough robustness, therefore the quantity a is set to be 1 in our approach (as is described in Figure 1A,B). The quantity b describes the point when the fault occurs in t_A. Its contribution is to guarantee that the assumption that the diagnostic software in time-critical applications has enough robustness is valid. When $b \in (0, 1)$, the fault information in the current collected signal may not be sufficient for a diagnostic procedure to form a response. If the software responds in the current diagnostic procedure, it means that the robustness of the software is too low, which is inconsistent with the assumption that the diagnostic software in time-critical applications has enough robustness. For this reason, the fault diagnostic system should respond in the next diagnostic procedure (as is shown in Figure 1B). Based on the analysis above, the time lag Δt can be estimated by $\Delta t \approx (1 + b)t_\mathrm{A} + t_\mathrm{B} = (1 + b)\frac{n}{f} + O[\mathrm{C}(n)]$, where $b \in (0, 1)$.

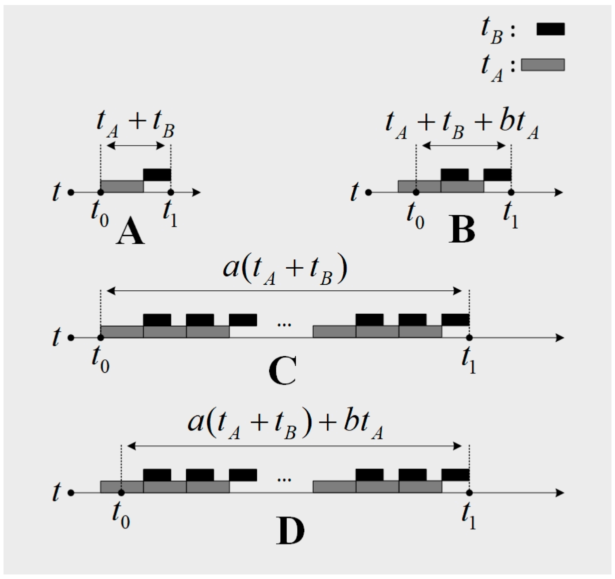


Figure 1. The constitution of the time lag Δt. (**A**,**B**) illustrate the situation that the diagnostic software has enough robustness. (**C**,**D**) illustrate the situation that the diagnostic software has low robustness.

3. Materials and Methods

The method adopted in this study involves the following steps: Building the optimization design model, narrowing the solution space through prior knowledge, and solving the model.

3.1. Building the Optimization Design Model

In the parametric design, a fault diagnostic software is often defined as a tuple $(\mathrm{C}, n, f, \sigma)$, where σ is the required accuracy of the designed fault diagnostic software. The required accuracy σ is often given in the design document. The diagnostic algorithm C is restricted to the classification method in machine learning. The specific type of the classification method should be determined based on the dataset and the premise of the classification method. The determination of the sampling rate f belongs to the work of the sensor selection before the parametric design. Therefore, the goal of designing the optimal quick diagnostic software can be defined as:

$$\underset{n \in \mathbb{N}}{\operatorname{argmin}} \Delta t = (1 + b) \times \frac{n}{f} + O[\mathrm{C}(n)];\ b \in (0, 1) \tag{1}$$

In the context of machine learning, the diagnostic algorithm C is a classifier which maps the signal (or time series) $\mathrm{s} \in \mathcal{R}^n$ to the system state $\ell \in \mathcal{R}$. It is learned by a learning algorithm A from a dataset **D**. The dataset **D** consists of many signal samples $\mathrm{s} \in \mathcal{R}^n$, and the system state ℓ of each sample is known. According to this definition, the diagnostic algorithm C should be represented as $\mathrm{C}^{(n)}$. Subsequently, the classifier is used to classify the new signal sample in order to diagnose faults.

To guarantee the accuracy of the designed diagnostic software to be no less than the required accuracy σ with the consideration of the effect caused by the design objective, the constraint is built into Equation (3). The accuracy of the designed diagnostic software $\mathrm{acc}(\mathrm{C}^{(n)})$ is estimated by the strategy of model selection in machine learning. At present, there are mainly three methods to estimate the accuracy: Holdout, cross validation, and bootstrapping [29]. The design method adopts the holdout approach in this paper. This method sets aside part of the dataset **D** for learning, and this part is called the training set. The remaining part of the dataset **D** is called the test set. It is used to provide an unbiased evaluation of the accuracy $\mathrm{acc}(\mathrm{C}^{(n)})$, as expressed in Equation (2). N in Equation (2) is the number of signal samples in the test set. In general, the holdout method randomly splits the dataset **D** into the training set and the test set according to a specified proportion, and the common proportions of the training set and the test set are 8:2 or 7:3.

$$\mathrm{acc}(\mathrm{C}^{(n)}) = \frac{1}{N}\sum_{n=1}^{N}[[\mathrm{C}^{(n)}(s) = \ell]] \tag{2}$$

$$\text{s.t. } \mathrm{acc}(\mathrm{C}^{(n)}) \geq \sigma \tag{3}$$

Combining Equations (1) and (3), the problem in the design of the optimal quick diagnostic system can be formulated as Equation (4). It minimizes the response time Δt and ensures that the accuracy of the designed diagnostic software $\mathrm{acc}(\mathrm{C}^{\mathrm{n}})$ is no less than σ through finding the optimal parameter n. In Equation (4), the parameters b and f are positive numbers, and the time complexity $O(\cdot)$ is a monotonically increasing function no matter which type of algorithm is employed. Therefore, the objective function is also a monotonically increasing function, and it is equivalent to find the minimal n. Based on this, the optimization model can be equivalently expressed as Equation (5).

$$\begin{gathered}\underset{n\in\mathbb{N}}{\mathrm{argmin}}\Delta t = (1+b)\times\tfrac{n}{f} + O[\mathrm{C}^{(n)}(n)];\ b\in(0,1)\\ \text{s.t. } \mathrm{acc}(\mathrm{C}^{(n)}) \geq \sigma\end{gathered} \tag{4}$$

$$\begin{gathered}\underset{\mathrm{n}\in\mathbb{N}^*}{\arg\min}\, n\\ \text{s.t. } \mathrm{acc}(\mathrm{C}^{(n)}) \geq \sigma\end{gathered} \tag{5}$$

3.2. Narrowing the Solution Space Using a Priori Knowledge

The set of natural numbers $\mathbb{N}$ is a huge range for solving Equation (5). To improve the efficiency of the procedure, the solution space is narrowed with the help of a priori knowledge about the sampling number n. The sampling number n determines the size of the collected signal. The larger the size of the collected signal, the more information that is available for the diagnostic algorithm. At present, the sampling numbers are 1024, 2048, and 4096 for the signal processing method, which are successfully used for fault diagnosis. Consequently, methods in machine learning also apply these sampling numbers to diagnose and achieve high accuracy [30–32]. Obviously, signals of these sizes provide enough information to acquire a satisfactory level of accuracy. Therefore, all these research works actually provide a rational upper bound m for the sampling number, if the required accuracy σ is lower than the accuracy related to the upper bound m. On this basis, an optimal parameter n was determined to be in a new smaller range of sampling numbers. Thus, the objective changed from designing a global optimal quick diagnostic system into designing a local one without any design specification violations, as expressed by Equation (6).

$$\begin{gathered}\underset{\mathrm{n}\in[0,m]}{\arg\ \min}\ n\\ \text{s.t. } \mathrm{acc}(\mathrm{C}^{(n)}) \geq \sigma\end{gathered} \tag{6}$$

3.3. Solving the Model

Theoretically, a signal of the length m is defined as $s = \{(t_i,\ x_i), i = 1,\cdots,m\}$. The timestamp t_i is actually a relative time point, and it can generally be ignored. Equation (6) aims to determine a diagnostic method $\mathrm{C}^{(n)} : R^n \to R^1$ which applies the shortest possible prefix of the signal s, denoted by $s[1,n] = \{(t_i,\ x_i)\,|\mathrm{i} = 1,\cdots,n; n < m\}$, to reach the required accuracy σ. To achieve the goal, an improved wrapper method is proposed. This method is based on the wrapper method, one of the three feature selection methods (filter [33,34], wrapper [33,35], and embedding [33,36]). The wrapper method has three basic steps [33,37,38], which are detailed as follows.

(1) A generation procedure to generate the candidate subset of features: The timestamp t_i is called the feature of a signal, and the value x_i is called the feature value. This step generates a subset of features $\mathrm{G} \subset \{t_i | i = 1, \cdots, m\}$. The number of elements in G is denoted by k, which is less than m. Then, beside the feature values related to the subset of features G, all the other feature values of each sample in the dataset **D** are removed to generate a new dataset $\mathbf{D}^*$.

(2) An evaluation procedure to evaluate the subset: The classifier $\mathrm{C}^{(k)}$ is learned based on the new dataset $\mathbf{D}^*$. The accuracy of the classifier $\mathrm{C}^{(k)}$ is estimated, which is denoted by $\mathrm{acc}(\mathrm{C}^{(k)})$.

(3) A judgment procedure to judge the stopping criterion: Steps 1 and 2 are repeated until the maximal $\mathrm{acc}(\mathrm{C}^{(k)})$ is acquired.

In order to solve the optimization problem in this paper, the wrapper method is improved, which is called Algorithm 1. Upon increasing of the sampling number from 1 to n, the subset of features is generated through the extraction of the first n elements in the feature set $\{t_i | i = 1, \cdots, m\}$ in the generation procedure. In the evaluation procedure, the holdout method is applied to evaluate the accuracy of the classifier $\mathrm{C}^{(n)}$. The stopping criterion is that the accuracy $\mathrm{acc}(\mathrm{C}^{(n)})$ must be no less than the required accuracy σ. When the criterion is met, the optimal sampling number n is obtained. The tuple $(\mathrm{C}, n, f, \sigma)$ is determined, thus the design is completed. As mentioned previously, the precondition that the required design accuracy σ be lower than the accuracy related to the upper bound m should be satisfied. If not, the solution may not be obtained.

Algorithm 1 Algorithm of the optimal solution

```
Step 1.  Determine the dataset D, the classifier $h$, the learning algorithm A, the required accuracy σ, the
proportion τ in the holdout method.
Step 2.  Initialize n = 0, $\mathrm{acc}(\mathrm{C}^{(0)}) = 0$, $\mathbf{D}^* = \varnothing$;
Step 3.  Execute the iteration below:
             While $\mathrm{acc}(\mathrm{C}^{(n)}) < \sigma$:
                 If $n < m$:
                 $n = n + 1$;
                 For every sample $s$ in D:
                     take $s[1, n]$ into $\mathbf{D}^*$
                 End
                 Learn the classifier $\mathrm{C}^{(n)}$ using the learning algorithm A on the dataset $\mathbf{D}^*$;
                 Estimate $\mathrm{acc}(\mathrm{C}^{(n)})$ using the holdout method;
                 $\mathbf{D}^* = \varnothing$
                 Else:
             End while
Step 4.  Output the optimal sampling number $n$
```

4. Experiment

4.1. Experimental Dataset

This paper adopts a dataset of bearing vibration signals from the Case Western Reserve University (CWRU) bearing test data center, and the conditions of data acquisition are given in [39]. The signal samples with single point faults in the inner race way were chosen for the experiments. The fault diameters in the signal samples included 0.007 in, 0.014 in, 0.021 in, and 0.028 in. The sampling frequency f was 12,000 samples/s and the length of each signal sample was at least 480,000.

According to previous research on bearing fault diagnosis based on the CWRU dataset, the signal length of 4096 was used for fault diagnosis [40]. Therefore, the upper bound m was 4096. According to prior knowledge about the signal length, each original signal sample was split into many segments; each segment was a new signal sample with a length of 4096. All the new signal samples composed the dataset **D**, and each sample in dataset **D** had the same label as the original signal from which it was split. The label information is shown in Table 1.

Table 1. The information of labels in dataset **D**.

Label 1	Description
0	This signal label indicates that that there is no fault.
1	This signal label indicates that there is a fault with a diameter of 0.007 in
2	This signal label indicates that there is a fault with a diameter of 0.014 in
3	This signal label indicates that there is a fault with a diameter of 0.021 in
4	This signal label indicates that there is a fault with a diameter of 0.028 in

4.2. Classifier

The Naïve Bayes classifier was selected as the diagnostic method C because the dataset **D** met the model requirement, i.e., the conditional independence assumptions [41]. In machine learning, the Naïve Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with independence assumptions between the features. With the results of the Pearson correlation test at every two timestamps (as is shown in Figure 2), the proportion of the coefficients between -0.3 and 0.3 reached up to 99.75%. The level of coefficients between -0.3 and 0.3 indicated that there was approximately no correlation between every two features. This suggested that the dataset **D** met the assumptions of the Naïve Bayes classifier.

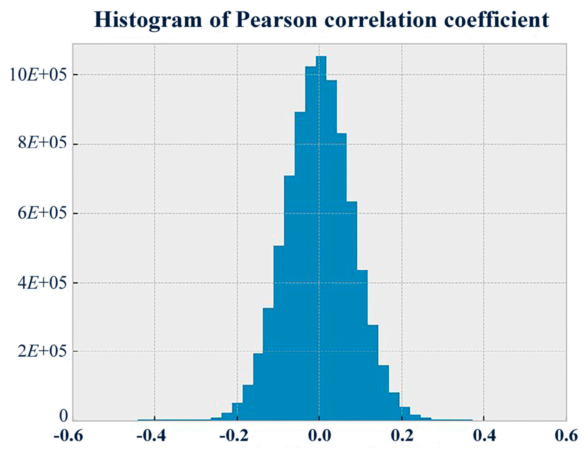


Figure 2. Histogram of Pearson correlation coefficients.

4.3. Experimental Method

Two experiments, denoted as experimental groups A and B, were conducted. In group A, eight diagnostic software were designed based on the proposed method, and the required accuracies σ were 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, and 0.95, respectively. Algorithm 1 was executed 200 times, taking into consideration the effect caused by the uncertainty of the training set on accuracy. In group B, the new datasets, shown in Table 2, were composed of signals from dataset **D**. Based on each new dataset, four diagnostic systems were designed under the required accuracies σ of 0.92, 0.94, 0.96, and 0.98. The algorithm was also executed 200 times due to the same consideration.

Table 2. The new datasets in group B.

Dataset	Description
$\mathbf{D}_{01}$	It includes the signal samples with a fault diameter of 0.007 in and normal signals.
$\mathbf{D}_{02}$	It includes the signal samples with a fault diameter of 0.014 in and normal signals.
$\mathbf{D}_{03}$	It includes the signal samples with a fault diameter of 0.021 in and normal signals.
$\mathbf{D}_{04}$	It includes the signal samples with a fault diameter of 0.028 in and normal signals.

5. Results and Discussion

In order to present the results intuitively, the values of n derived from the 200 executions of the algorithm were statistically analyzed. Four statistics, including the mean, standard deviation, maximum, and minimum, were chosen to quantify the distributions of the sampling number n, and the histograms were also produced.

5.1. Results in Group A

In group A, the 200 executions of the algorithm with different accuracy requirements all obtained the optimal sampling number n. Therefore, the parametric design method proved to be feasible to design the optimal quick diagnostic software. The histograms of all eight distributions of sampling number n are presented in Figure 3 using a style of overlapping densities. Obviously, the sampling number n increased with the increasing required accuracy σ. This was consistent with the conclusion that the fault diagnostic system can become more accurate as more signal information becomes available. However, as shown in Table 3, the standard deviation increased from 2.88 to 245.74. In statistics, a high standard deviation indicates that the data points are spread out over a wider range of values. For example, this phenomenon is observed in the histogram with the required accuracy of 0.95. Although the maximal sampling number n can be used to design the diagnostic software conservatively, it is impossible to ignore the accuracy requirement σ causes the designed diagnostic software to become less robust under conditions of noise and uncertainty. In fact, the designed fault diagnostic software needs more signal information due to the increasing accuracy requirement. However, as more signal information becomes available, the noise also increases. Yet, according to the same proportion τ, the training data is generated randomly through the holdout method. This results in the significant uncertainty of the training signal samples. As is shown in Figure 3 and Table 3, with the same increment in the required accuracy σ, the increasing rate of the standard deviation was much higher. This indicated that the fault diagnostic system, designed to reach high accuracy, was very sensitive to noise and uncertainty. Thus, it follows that a high required accuracy reduces the robustness of the designed software.

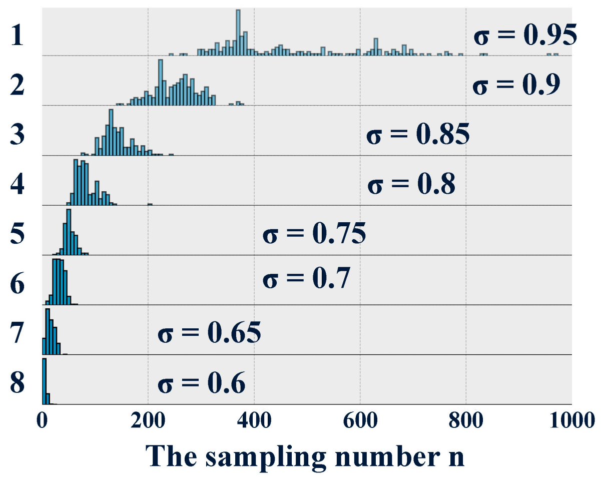


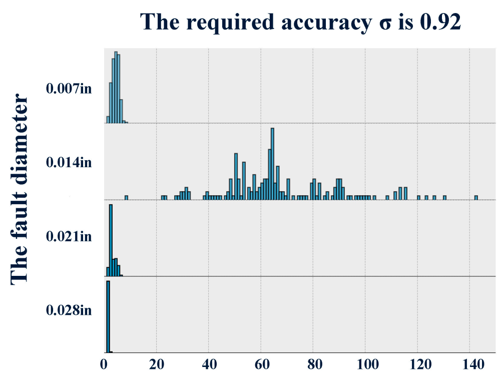
Figure 3. The overlapping densities plot of group A.

Table 3. The statistics for the designed diagnostic system in group A.

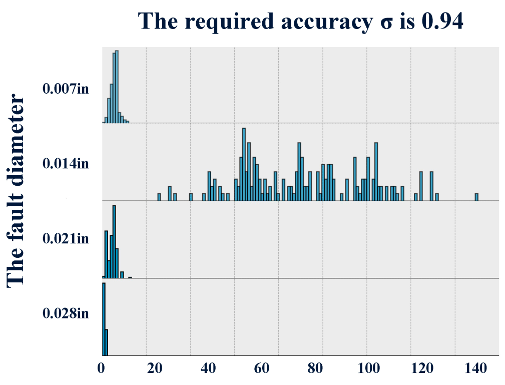
Designed Diagnostic System	Required Accuracy σ (%)	Mean	Standard Deviation	Maximum	Minimum
1	0.6	5	2.88	20	2
2	0.65	15.25	7.53	41	3
3	0.7	32.28	9.15	61	8
4	0.75	51.97	9.79	83	26
5	0.8	82.95	20.77	206	50
6	0.85	142.03	27.72	240	79
7	0.9	252.04	43.01	378	144
8	0.95	506.79	245.74	902	240

5.2. Results in Group B

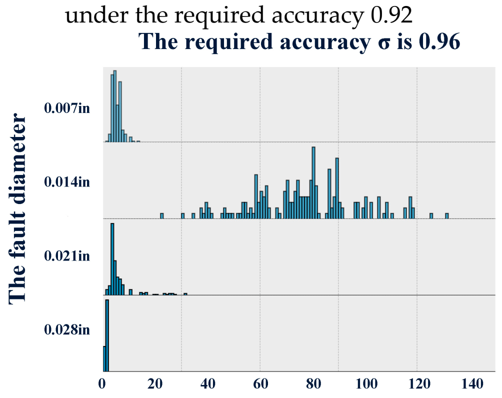
For group B, the overlapping densities plot is presented in Figure 4 by combining the designed diagnostic software with the same required accuracy σ in different datasets. The statistics are shown in Tables 4–7. It can be observed that the means decreased with the increasing fault diameter on the whole. However, it cannot be concluded that the more serious the fault is, the quicker the fault diagnostic software responds. This conclusion was not consistent with the understanding that bearings with a larger fault diameter are often easy to diagnose quickly. Equivalently, as shown in Figure 4, the maximum of the sampling number n in the design based on the dataset $\mathbf{D}_{01}$ was less than the minimum sampling number n in the design based on the dataset $\mathbf{D}_{02}$. This also means that the response of the diagnostic software to a serious fault is slower than the response to a slight fault. However, it can still be observed that the standard deviation increased with the change in the required accuracy. At the same time, we cannot give a rational explanation for the significant large standard deviation in the design based on the signal samples with a fault diameter of 0.014 in.



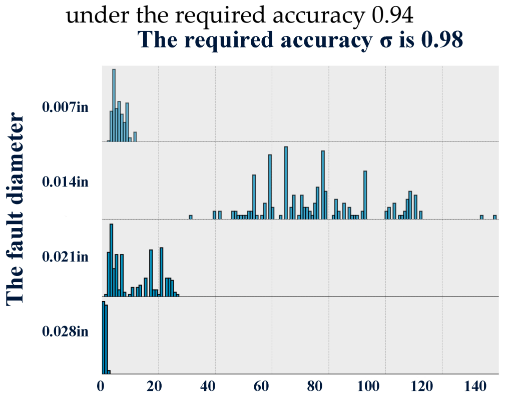
(**a**) The overlapping densities of group B under the required accuracy 0.92



(**b**) The overlapping densities of group B under the required accuracy 0.94



(**c**) The overlapping densities of group B under the required accuracy 0.96



(**d**) The overlapping densities of group B under the required accuracy 0.98

Figure 4. The overlapping densities plot of group B.

Table 4. The standard deviation of the sampling number in group B.

Required Accuracy σ (%)	Fault Diameter			
	0.007 in	0.014 in	0.021 in	0.028 in
0.92	1.33	22.27	1.13	0.12
0.94	1.71	28.61	1.77	0.44
0.96	2.94	33.03	9.47	0.48
0.98	5.40	49.06	20.10	0.76

Table 5. The mean of the sampling number in group B.

Required Accuracy σ (%)	Fault Diameter			
	0.007 in	0.014 in	0.021 in	0.028 in
0.92	3.87	66.84	2.64	1.015
0.94	5.3	91.42	4.305	1.265
0.96	8.495	127.585	10.075	1.77
0.98	14.805	190.625	28.43	2.64

Table 6. The maximum of the sampling number in group B.

Required Accuracy σ (%)	Fault Diameter			
	0.007 in	0.014 in	0.021 in	0.028 in
0.92	8	142	6	2
0.94	11	170	12	2
0.96	22	219	53	3
0.98	30	346	66	6

Table 7. The minimum of the sampling number in group B.

Required Accuracy σ (%)	Fault Diameter			
	0.007 in	0.014 in	0.021 in	0.028 in
0.92	1	8	1	1
0.94	1	26	1	1
0.96	3	37	2	1
0.98	5	78	3	2

5.3. Comparison with Related Works

In order to evaluate the proposed design method objectively, it was compared with other methods. The methods used for comparison should provide the estimation of the accuracy, and the sampling number and sampling frequency should be known. Based on this, four methods were chosen, as listed in the first column of Table 8. The comparison was based on the dataset **D** with the sampling frequency of 12,000 samples/second. The accuracy was estimated based on the dataset **D** by our calculations, which are listed in the fifth column of Table 8. The sampling number that enabled the generation of a diagnostic result is listed in the third column of Table 8, outside the parentheses. It is worth noting, however, that the listed accuracy may not reflect the real performance of those methods, although the listed accuracy is the best of those obtained after many executions of the proposed algorithm. Based on those accuracies, four diagnostic software were designed. The upper bound of the sampling number was also set to be the same as the sampling number used in the compared methods. In each software design, Algorithm 1 was executed 200 times, taking into consideration the effect caused by uncertainty. The mean of the sampling number n is listed in the third column of Table 8, inside the parentheses. Because the execution time of the diagnostic algorithm t_B is determined by the sampling number n and the algorithm type C, the time complexity of each compared method is also provided in Table 8, in the fourth column outside the parentheses (the relationship with the time complexity of our approach is recorded inside the parentheses). According to the comparison of the sampling number and the time complexity, it is reasonable to believe that the designed fault diagnostic software has a smaller time lag Δt than those that were designed by the compared methods under the same accuracy and sampling frequency. Therefore, our approach exhibited superiority in the trade-off between accuracy and the response time.

Table 8. Comparison with related works.

Method	f	n	Time Complexity	σ
W. Du et al. [42]	12 kHz	2048 (>411)	O(nlogn) (> O(n))	88.3%
X. Jin et al. [43]	12 kHz	12,000 (>601)	O(n^3) (> O(n))	93.1%
M. Amar et al. [20]	12 kHz	4104 (>1188)	O(n) (= O(n))	96.2%
X. Zhang et al. [44]	12 kHz	2400 (>1248)	O(n!) (> O(n))	97.3%

6. Conclusions

This study proposed a parametric design method for an optimal quick diagnostic system. Using this method, an optimal quick diagnostic system can be obtained, which has the smallest response time and a pre-defined accuracy. By means of experimental verification, the feasibility of the parametric design method was validated. Contrary to other existing methods, the proposed approach fulfills the following three characteristics:

First, the measure of the response time was proposed in this method, which provides a common standard to quantitatively assess the improvement of the response time.

Secondly, using the parametric design method proposed in this paper, the designed fault diagnostic software exhibited an obvious improvement in the response time to faults as compared with the response times of traditional methods with the same vibration data.

Thirdly, the parametric design method exhibited superiority in the trade-off between the accuracy and the response time.

Additionally, the experimental results revealed that the higher the required accuracy is, the less robust the designed diagnostic software is to noise and uncertainty. At the same time, the experimental results demonstrate that serious faults are guaranteed to be diagnosed more quickly by the diagnostic software; although, the more serious the fault is, the more easily the feature will be recognized.

Author Contributions: X.Y. contributed to the conception of the study, conducted the data analysis, and wrote the manuscript, X.Y. and P.H. prepare the manuscript and revised the manuscript.

Funding: This work is supported by National Natural Science Foundation (71801196), National Major Project from Shanghai Nuclear Engineering Research & Design Institute (2017ZX06002006), Open Research Program from CAS Key Laboratory of Solar Activity in National Astronomical Observatories (KLSA201803), China Postdoctoral Science Foundatuion (2018M631606), and Pre-Research Project (41404060103).

Acknowledgments: The authors is grateful to two anonymous reviewers for their useful comments and suggestions to the original version of the paper.

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

References

1. Hatami, N.; Chira, C. Classifiers with a reject option for early time-series classification. In Proceedings of the IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL), Singapore, 16–19 April 2013; pp. 9–16.
2. Muradore, R.; Fiorini, P. A PLS-Based Statistical Approach for Fault Detection and Isolation of Robotic Manipulators. *IEEE Trans. Ind. Electron.* **2012**, *59*, 3167–3175. [CrossRef]
3. Lee, J.; Ghaffari, M.; Elmeligy, S. Self-maintenance and engineering immune systems: Towards smarter machines and manufacturing systems. *Annu. Rev. Control* **2011**, *35*, 111–122. [CrossRef]

4. Leng, Y.G.; Tian, X.Y. Application of a first-order linear system's stochastic resonance in fault diagnosis of rotor shaft. *J. Vib. Shock* **2014**, *33*, 1–5.
5. Ou, L.; Li, D.; Zeng, X. Ship Propulsion Fault Diagnosis System Design Based on Remote Network. In Proceedings of the Fifth International Conference on Measuring Technology and Mechatronics Automation, Hong Kong, China, 16–17 January 2013; pp. 993–995.
6. Chen, J.; Li, Z.; Pan, J.; Chen, G.; Zi, Y.; Yuan, J.; Chen, B.; He, Z. Wavelet transform based on inner product in fault diagnosis of rotating machinery: A review. *Mech. Syst. Signal Process.* **2016**, *70–71*, 1–35. [CrossRef]
7. Yu, G.; Li, C.; Kamarthi, S. Machine fault diagnosis using a cluster-based wavelet feature extraction and probabilistic neural networks. *Int. J. Adv. Manuf. Technol.* **2009**, *42*, 145–151. [CrossRef]
8. Jia, F.; Lei, Y.; Shan, H.; Lin, J. Early Fault Diagnosis of Bearings Using an Improved Spectral Kurtosis by Maximum Correlated Kurtosis Deconvolution. *Sensors* **2015**, *15*, 29363–29377. [CrossRef] [PubMed]
9. Jiang, R.; Chen, J.; Dong, G.; Liu, T.; Xiao, W. The weak fault diagnosis and condition monitoring of rolling element bearing using minimum entropy deconvolution and envelop spectrum. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2013**, *227*, 1116–1129. [CrossRef]
10. Ming, Y.; Chen, J.; Dong, G. Weak fault feature extraction of rolling bearing based on cyclic Wiener filter and envelope spectrum. *Mech. Syst. Signal Process.* **2011**, *25*, 1773–1785. [CrossRef]
11. Yu, J.; Lv, J. Weak Fault Feature Extraction of Rolling Bearings Using Local Mean Decomposition-Based Multilayer Hybrid Denoising. *IEEE Trans. Instrum. Meas.* **2017**, *99*, 1–12. [CrossRef]
12. Yong, L.; You-Rong, L.I.; Wang, Z.G. Research on a extraction method for weak fault signal and its application. *J. Vib. Eng.* **2007**, *20*, 24–28.
13. Xu, Y.G.; Ma, H.L.; Fu, S.; Zhang, J.Y. Theory and applications of weak signal non-linear detection method for incipient fault diagnosis of mechanical equipments. *J. Vib. Eng.* **2011**, *24*, 529–538.
14. Dong, G.M.; Chen, J. Noise resistant time frequency analysis and application in fault diagnosis of rolling element bearings. *Mech. Syst. Signal Process.* **2012**, *33*, 212–236. [CrossRef]
15. He, Q.; Wang, J.; Liu, Y.; Dai, D.; Kong, F. Multiscale noise tuning of stochastic resonance for enhanced fault diagnosis in rotating machines. *Mech. Syst. Signal Process.* **2012**, *28*, 443–457. [CrossRef]
16. Guo, W.; Zhou, Z.; Chen, C. Cascaded and parallel stochastic resonance for weak signal detection and its simulation study. In Proceedings of the Prognostics and System Health Management Conference (PHM-Chengdu), Chengdu, China, 19–21 October 2016; pp. 1–6.
17. Liu, X.; Liu, H.; Yang, J.; Litak, G.; Cheng, G.; Han, S. Improving the bearing fault diagnosis efficiency by the adaptive stochastic resonance in a new nonlinear system. *Mech. Syst. Signal Process.* **2017**, *96*, 58–76. [CrossRef]
18. Hu, N.Q.; Chen, M.; Wen, X. The Application of Stochastic Resonance Theory for Early Detecting Rub-impact of Rotor System. *Mech. Syst. Signal Process.* **2003**, *17*, 883–895. [CrossRef]
19. Shi, P.; Ding, X.; Han, D. Study on multi-frequency weak signal detection method based on stochastic resonance tuning by multi-scale noise. *Measurement* **2014**, *47*, 540–546. [CrossRef]
20. Amar, M.; Gondal, I.; Wilson, C. Vibration Spectrum Imaging: A Novel Bearing Fault Classification Approach. *IEEE Trans. Ind. Electron.* **2015**, *62*, 494–502. [CrossRef]
21. Ciabattoni, L.; Ferracuti, F.; Freddi, A.; Monteriu, A. Statistical Spectral Analysis for Fault Diagnosis of Rotating Machines. *IEEE Trans. Ind. Electron.* **2018**, *65*, 4301–4310. [CrossRef]
22. Bregón, A.; Simón, M.A.; Rodríguez, J.J.; Alonso, C.; Pulido, B.; Moro, I. Early Fault Classification in Dynamic Systems Using Case-Based Reasoning. In *Current Topics in Artificial Intelligence, Proceedings of the 11th Conference of the Spanish Association for Artificial Intelligence, CAEPIA 2005, Santiago de Compostela, Spain, 16–18 November 2005*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2005; Volume 4177, pp. 211–220.
23. Ando, S.; Suzuki, E. Minimizing response time in time series classification. *Knowl. Inf. Syst.* **2016**, *46*, 449–476. [CrossRef]
24. Shi, W.W.; Yan, H.S.; Ma, K.P. A new method of early fault diagnosis based on machine learning. In Proceedings of the 2005 International Conference on Machine Learning and Cybernetics, Guangzhou, China, 18–21 August 2005; pp. 3271–3276.
25. Wan, J.; Yu, Y.; Wu, Y.; Feng, R.; Yu, N. Hierarchical leak detection and localization method in natural gas pipeline monitoring sensor networks. *Sensors* **2012**, *12*, 189–214. [CrossRef]

26. Willsky, A.S. A survey of design methods for failure detection in dynamic systems. *Automatica* **1976**, *12*, 601–611. [CrossRef]
27. Wang, G. Basic Research on Machinery Fault Diagnosis—What is the Prescription. *J. Mech. Eng.* **2013**, *49*, 63–72. [CrossRef]
28. Sipser, M. *Introduction to the Theory of Computation: Preliminary Edition*; PWS Pub. Co.: Boston, MA, USA, 1996.
29. Kim, J.-H. Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Comput. Stat. Data Anal.* **2009**, *53*, 3735–3745. [CrossRef]
30. Sreenath, P.G.; Praveen Kumare, G.; Pravin, S.; Vikram, K.N.; Saimurugan, M. Automobile Gearbox Fault Diagnosis Using Naive Bayes and Decision Tree Algorithm. *Appl. Mech. Mater.* **2015**, *813/814*, 943–948. [CrossRef]
31. Jack, L.B.; Nandi, A.K. Fault Detection Using Support Vector Machines and Artificial Neural Networks, Augmented by Genetic Algorithms. *Mech. Syst. Signal Process.* **2002**, *16*, 373–390. [CrossRef]
32. Samanta, B.; Al-Balushi, K.R.; Al-Araimi, S.A. Bearing fault detection using artificial neural networks and genetic algorithm. *Eng. Appl. Artif. Intell.* **2003**, *16*, 657–665. [CrossRef]
33. Dash, M.; Liu, H. Feature selection for classification. *Intell. Data Anal.* **1997**, *1*, 131–156. [CrossRef]
34. Kira, K.; Rendell, L.A. The feature selection problem: Traditional methods and a new algorithm. In Proceedings of the Tenth National Conference on Artificial Intelligence, San Jose, CA, USA, 12–16 July 1992; pp. 129–134.
35. Liu, H.; Setiono, R. Feature selection and classification—A probabilistic wrapper approach. In Proceedings of the 9th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Fukuoka, Japan, 4–7 June 1996; pp. 419–424.
36. Weston, J.; Tipping, M. Use of the zero norm with linear models and kernel methods. *J. Mach. Learn. Res.* **2003**, *3*, 1439–1461.
37. Doak, J. An Evaluation of Feature Selection Methods and Their Application to Computer Security. Available online: https://escholarship.org/uc/item/2jf918dh (accessed on 13 February 2019).
38. Siedlecki, W.; Sklansky, J. On Automatic Feature Selection. In *Handbook of Pattern Recognition and Computer Vision*; World Scientific Publishing Co., Inc.: River Edge, NJ, USA, 1988; pp. 63–87.
39. Smith, W.A.; Randall, R.B. Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study. *Mech. Syst. Signal Process.* **2015**, *64–65*, 100–131. [CrossRef]
40. Yi, X.J.; Chen, Y.F.; Hou, P. Fault Diagnosis of Rolling Element Bearing Using Naïve Bayes Classifier. *J. Vibroeng.* **2017**, *19*, 64–69.
41. Murty, M.N.; Devi, V.S. *Pattern Recognition: An Algorithmic Approach*; Springer: London, UK, 2011.
42. Du, W.; Tao, J.; Li, Y.; Liu, C. Wavelet leaders multifractal features based fault diagnosis of rotating mechanism. *Mech. Syst. Signal Process.* **2014**, *43*, 57–75. [CrossRef]
43. Jin, X.; Zhao, M.; Chow, T.W.; Pecht, M. Motor Bearing Fault Diagnosis Using Trace Ratio Linear Discriminant Analysis. *IEEE Trans. Ind. Electron.* **2014**, *61*, 2441–2451. [CrossRef]
44. Zhang, X.; Liang, Y.; Zhou, J. A novel bearing fault diagnosis model integrated permutation entropy, ensemble empirical mode decomposition and optimized SVM. *Measurement* **2015**, *69*, 164–179. [CrossRef]