

## Research Article

# A New Reliable Operating Region Design Method

Yue Chen <sup>1,2</sup>, Jian Shi <sup>1,2</sup> and Xiao-jian Yi <sup>1,3,4</sup>

<sup>1</sup>Academy of Mathematics and Systems Science, Chinese Academy of Sciences, 100190 Beijing, China

<sup>2</sup>School of Mathematical Sciences, University of Chinese Academy of Sciences, 100049 Beijing, China

<sup>3</sup>Department of Overall Technology, China North Vehicle Research Institute (No. 1 Research Academy of China Ordnance), 100072 Beijing, China

<sup>4</sup>School of Mechatronics Engineering, Beijing Institute of Technology, 100081 Beijing, China

Correspondence should be addressed to Jian Shi; [jshi@iss.ac.cn](mailto:jshi@iss.ac.cn) and Xiao-jian Yi; [yixiaojianbit@sina.cn](mailto:yixiaojianbit@sina.cn)

Received 29 October 2019; Accepted 14 January 2020; Published 24 February 2020

Academic Editor: Alberto Campagnolo

Copyright © 2020 Yue Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The reliable operating region, which is the set of all possible points in the design space that satisfy the reliability requirement, is capable to improve the reliability of products in the design stage. However, the reliable operating region has an irregular geometry shape and it is hard to derive an explicit expression; therefore, its practicality is poor. In order to obtain a more convenient approach, this paper proposes a reliable hyperrectangle operating region, which is expressed by permissible intervals for each design parameter and has the advantage that design parameters are decoupled. An iterative algorithm that seeks an axis-parallel reliable hyperrectangle with maximum volume is proposed. Starting from a design point with target performance, the lengths of the sides of the reliable hyperrectangle are iteratively updated. Theoretical analysis shows that the proposed algorithm is convergent. Furthermore, we extend the proposed methodology to deal with design space constraints. Some numerical examples and engineering cases demonstrate that the proposed algorithm can achieve the requirement of reliability efficiently.

## 1. Introduction

There are many product quality indexes, among which performance and reliability are the most important. Performance is a measurement that designers use to evaluate products' operation. Reliability relates to a products' ability to perform its specified performance under the stated conditions [1].

Design, manufacture, use, maintenance, and various other factors determine the reliability of products [2, 3]. In the manufacturing industry, however, the focus is upon the design and manufacture of products. It is relatively easy to evaluate reliability in the later stages of product development, but it is expensive to change and improve products at that time. This makes it important to ensure product reliability in the earlier design stage [4].

The relationship between the design and the performance is shown in Figure 1. In this paper, the vector  $\mathbf{x} = (x_1, \dots, x_m)$  represents the design points or design parameters [5] and the function  $h(\mathbf{x})$  describes the performance, and  $y$

denotes the output performance, and the performance specification [5] describes the performance requirement of products, represented by  $[y^l, y^u]$ . The design space is defined by the lower and upper bounds, represented by  $\mathbf{x}^l = (x_1^l, \dots, x_m^l)$  and  $\mathbf{x}^u = (x_1^u, \dots, x_m^u)$ , respectively. The feasible operating region [5] or feasible space [6] is the set of all possible points in the design space that satisfy the performance specification in the deterministic case. The set of all possible points in the design space that meet the reliability requirement in the probabilistic case is called the reliable operating region [7, 8] or reliable design space [6]. The reliable operating region defined by the probabilistic case ought to be a subset of the feasible operating region, for which no uncertainty is considered [8, 9]. An exact description of the reliable operating region is in general not simple, since it may be a very complex set [6].

Some efforts have been devoted to the reliability design [8–13], and the existing literature is mainly focused on how to find an optimum design point in the reliable operating region [6, 14–16]. Aiming to find the optimum design point while

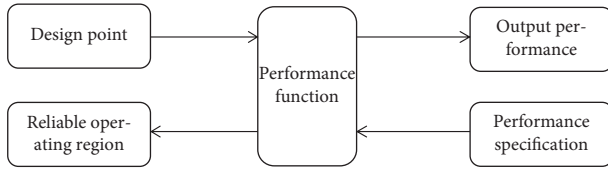


FIGURE 1: The relationship between design and performance.

considering the impacts of uncertainties, a method based on the first-order reliability was proposed in [6], which converted the probabilistic constraints to approximate deterministic constraints without unduly sacrificing accuracy. A deterministic optimization method was presented to identify the optimum design points whose performances are equal to the target values within the reliable operating region in [14].

Apart from the optimum design point, engineers are also interested in the reliable operating region because it provides the permissible ranges for design parameters. However, the reliable operating region is irregularly shaped and inconvenient to use. It is desirable in engineering to build a space which is not only reliable but also easy to use.

We aim to seek a maximum reliable hyperrectangle operating region in which every design point satisfies the reliability requirement. There are two main benefits from this hyperrectangle. First, it determines a relatively large permissible interval for each design parameter, so engineers can use low-precision equipment which helps to reduce the manufacturing cost. Second, it is such that design parameters are decoupled. For a design to be reliable, the choice of a design parameter value within its assigned interval does not depend on the values of the other design parameters as long as they are within their respective intervals.

In some approaches, hyperrectangles are identified with the aid of Monte Carlo sampling technique [17–19]. Cluster analysis in [19] was to compute hyperrectangles on the basis of one stochastic sample of design points. This approach is limited to problems with large design spaces or extremely large sample sizes. The stochastic algorithm proposed in [20] combined query and online learning, which probed a candidate hyperrectangle by stochastic sampling, and then readjusted its boundaries in order to, first, remove designs outside the feasible operating region and, second, explore more design space that had not been probed before. The quality of the results and the efficiency of this stochastic algorithm were studied in detail in [21]. This stochastic approach, however, produced a hyperrectangle that might contain some design points outside the feasible operating region and might not have the best size because Monte Carlo sampling was used to estimate the boundaries of the feasible operating region. The algorithm presented in [5], applied interval arithmetic within an iterative optimization scheme to decide whether the candidate hyperrectangle was contained in the feasible operating region. Interval arithmetic, however, limits the applicability of the algorithm because the accuracy of the results depends on the problem and cannot be assessed for general cases [22].

There are several challenges associated with seeking the maximum hyperrectangle. First, because the reliable

operating region is often irregularly shaped, we cannot solve this optimization problem directly. How to find a reliable hyperrectangle with the maximum volume, which is contained in the reliable operating region, is challenging. Second, the initial point of the reliable hyperrectangle is critical, but how to determine a reasonable initial point is difficult.

To address these challenges, firstly, we propose an iterative algorithm called the reliable hyperrectangle algorithm. This algorithm iteratively updates the lengths of the sides of the reliable hyperrectangle. Moreover, we prove the convergence of this algorithm under some conditions. Furthermore, the design point whose performance is approximately equal to its target value is suitable as the initial point.

## 2. The Reliable Hyperrectangle

As we point out in the introduction, the reliable operating region is determined by the performance function, the performance specification, and the reliability requirement. It is hard to describe the geometry shape of the reliable operating region and difficult to derive an explicit expression [23]. The lack of the explicit expression brings difficulty to the directly application of the reliable operating region in the practical engineering. This motivates us to find a reliable hyperrectangle operating region. Actually, because the rectangle-shaped operating region is characterized by simple structure, convenient operation, easy control, and powerful usability, it has been widely applied in other engineering problems [24]. For example, the hyperrectangle is used for concurrent design of vehicle tires and axles in [25] and is applied to some medical problems in [26].

The axis-parallel reliable hyperrectangle operating region is expressed as intervals for each design parameter. It has the advantage that design intervals for one design parameter are independent of the other design parameter values, that is, design parameters can be decoupled. The reliability requirement is satisfied as long as design parameters lie within their respective design intervals. Moreover, the intervals might be combined with intervals of other disciplines, thus their cross-sections are global reliable hyperrectangle. Additionally, maximum flexibility for the design is provided if the intervals are as large as possible.

The problem for solving the global minimum of a function in a hyperrectangle operating region has been studied [27–30]. Instead, this paper focuses on the inverse problem, that is, finding a maximum hyperrectangle operating region which enables the function to satisfy constraints.

In this section, we consider hyperrectangle design in two different cases. One is the deterministic case in which we assume there are no random errors among the manufacture or the effect of the errors is usually very small and can be neglected. The other is the probabilistic case in which we assume there exist random errors between the nominal performances and the output performances.

The axis-parallel hyperrectangle is a product of intervals, that is,

$$\Omega = I_1 \times I_2 \times \cdots \times I_m, \quad (1)$$

where  $I_i = [x_i^l, x_i^u]$  and  $x_i^l$  and  $x_i^u$  are the lower and upper boundaries of the hyperrectangle. As convenient, we group the lower boundaries,  $x_i^l$ 's into a vector  $\mathbf{x}^l$  and group the upper boundaries  $x_i^u$ 's into a vector  $\mathbf{x}^u$ , that is,  $\mathbf{x}^l = (x_1^l, \dots, x_m^l)$  and  $\mathbf{x}^u = (x_1^u, \dots, x_m^u)$ .

**2.1. Feasible Operating Region (Deterministic Case).** In the deterministic case, the relationship between the design point and the output performance is stated as

$$y = h(\mathbf{x}), \quad (2)$$

that is, the uncertainty is not considered.

We are looking for a maximum hyperrectangle, which has the advantage that whatever design point is selected in this hyperrectangle, it will comply with the performance specification. That is, we seek a maximum hyperrectangle contained in the feasible operating region.

The problem of searching for a maximum hyperrectangle in the feasible operating region can be formulated as follows:

$$\begin{aligned} \max_{\mathbf{x}^l, \mathbf{x}^u} \quad & \prod_{i=1}^m (x_i^u - x_i^l), \\ \text{subject to} \quad & \text{mdl}(\mathbf{x}^l, \mathbf{x}^u) \geq 0, \\ & \text{mdu}(\mathbf{x}^l, \mathbf{x}^u) \geq 0, \\ & \mathbf{x}^l \leq \mathbf{x}^l \leq \mathbf{x}^u \leq \mathbf{x}^u, \end{aligned} \quad (3)$$

where

$$\text{mdl}(\mathbf{x}^l, \mathbf{x}^u) = \min_{\mathbf{x}^l \leq \mathbf{x} \leq \mathbf{x}^u} h(\mathbf{x}) - y^l, \quad (4)$$

$$\text{mdu}(\mathbf{x}^l, \mathbf{x}^u) = \min_{\mathbf{x}^l \leq \mathbf{x} \leq \mathbf{x}^u} y^u - h(\mathbf{x}). \quad (5)$$

If the conditions  $\text{mdl}(\mathbf{x}^l, \mathbf{x}^u) \geq 0$  and  $\text{mdu}(\mathbf{x}^l, \mathbf{x}^u) \geq 0$  are satisfied, then for any  $\mathbf{x} \in [\mathbf{x}^l, \mathbf{x}^u]$ , we have  $y^l \leq h(\mathbf{x}) \leq y^u$ . This implies that each design point in the hyperrectangle maintains the output performance within specified bounds.

**2.2. Reliable Operating Region (Probabilistic Case).** Due to noise factors and manufacture errors, there exist some errors between the output performances and the nominal performances. Let  $\varepsilon$  represent the random error, then

$$y = h(\mathbf{x}) + \varepsilon, \quad (6)$$

that is, uncertainty is considered. Notice that  $\varepsilon$  is a random error which is described by a statistical distribution type predetermined by engineers.

In this probabilistic case, the reliability can be calculated point by point as follows:

$$r(\mathbf{x}) = P(y^l \leq y \leq y^u), \quad (7)$$

where

$$\begin{aligned} P(y^l \leq y \leq y^u) &= P(y^l - h(\mathbf{x}) \leq \varepsilon \leq y^u - h(\mathbf{x})) \\ &= F(y^u - h(\mathbf{x})) - F(y^l - h(\mathbf{x})), \end{aligned} \quad (8)$$

and  $F(\cdot)$  is the cumulative distribution function of  $\varepsilon$ . The design point is considered reliable if its reliability is above a given threshold  $\alpha$ . The reliable operating region is the set of all possible points in the design space whose reliabilities are above this given threshold.

Maximizing an axis-parallel hyperrectangle in which each design point can satisfy the reliability requirement, that is, seeking an axis-parallel maximum hyperrectangle in the reliable operating region, can be stated as the following optimization problem:

$$\begin{aligned} \max_{\mathbf{x}^l, \mathbf{x}^u} \quad & \prod_{i=1}^m (x_i^u - x_i^l), \\ \text{subject to} \quad & \text{mp}(\mathbf{x}^l, \mathbf{x}^u) \geq 0, \\ & \mathbf{x}^l \leq \mathbf{x}^l \leq \mathbf{x}^u \leq \mathbf{x}^u, \end{aligned} \quad (9)$$

where

$$\text{mp}(\mathbf{x}^l, \mathbf{x}^u) = \min_{\mathbf{x}^l \leq \mathbf{x} \leq \mathbf{x}^u} r(\mathbf{x}) - \alpha. \quad (10)$$

If the condition  $\text{mp}(\mathbf{x}^l, \mathbf{x}^u) \geq 0$  is satisfied, then for any  $\mathbf{x} \in [\mathbf{x}^l, \mathbf{x}^u]$ , we have  $r(\mathbf{x}) \geq \alpha$ . This implies there is at least a  $100\alpha\%$  probability that the obtained products are reliable as long as the design point is within the reliable hyperrectangle.

In order to verify whether the design point satisfies the reliability requirement, the Monte Carlo sampling technique is used. We sample some random errors from the predetermined distribution and calculate the coverage probability which is the proportion that the output performances lie within specified bounds. Notice that the output performance is given by equation (6), that is, the sum of the nominal performance and the random error. If the coverage probability is above the given reliability threshold, the reliability of the design point is considered to be validated.

### 3. The Reliable Hyperrectangle Algorithm

The optimization problems (3) and (9) cannot be directly solved, so an iterative method is used. In this section, we propose an iterative reliable hyperrectangle algorithm, which starts from an initial point  $\mathbf{x}_0 = (x_{01}, x_{02}, \dots, x_{0m})$  to search out a maximum hyperrectangle in the reliable operating region. For some  $\epsilon > 0$ , the condition  $\text{mp}(\mathbf{x}_0, \mathbf{x}_0) \geq \epsilon$  is satisfied. We will present the selection strategy of the initial point  $\mathbf{x}_0$  in the remainder of this paper.

This section mainly presents how to use the proposed algorithm to solve problem (9). This algorithm can also be used to solve problem (3).

**3.1. Algorithm Design.** The proposed iterative algorithm is called the reliable hyperrectangle algorithm. The basic idea is to adjust the boundaries of the hyperrectangle one-by-one until they cannot be extended any further. If an index  $i$  is selected, the hyperrectangle is updated by increasing an

appropriate size in the  $i$ th design parameter while other design parameters are fixed. Assuming the index  $i$  is chosen in a cyclic manner from the set  $o = (o_1, o_2, \dots, o_m)$ , which is a permutation of  $1, 2, \dots, m$ .

More specifically, if an index  $i \in o$  is selected, an extension part of the hyperrectangle is expressed by the lower boundary  $l_i^u$  and the upper boundary  $u_i^u$ , where  $l_i^u = (x_1^l, \dots, x_{i-1}^l, x_i^u, x_{i+1}^l, \dots, x_m^l)$  and  $u_i^u = (x_1^u, \dots, x_{i-1}^u, x_i^u + d, x_{i+1}^u, \dots, x_m^u)$ . We first seek an appropriate size  $d$  which ensures the extension part of hyperrectangle is reliable, that is, the condition  $\text{mp}(l_i^u, u_i^u) \geq \epsilon$  is satisfied. Then, we set  $x_i^u = x_i^u + d$ . The upper boundary of the  $i$ th design parameter is updated.

The step size  $d$  is determined by the following method. Given an initial step size  $d_0$ , if the condition  $\text{mp}(l_i^u, u_i^u) \geq \epsilon$  is satisfied, we accept this step. Otherwise, based on the dichotomy method, we find some  $d$  between 0 and  $d_0$  such that  $\text{mp}(l_i^u, u_i^u) = \epsilon$ . Notice that the value of  $\text{mp}(l_i^u, u_i^u)$  is calculated by the interior-point algorithm.

The lower boundary  $x^l$  is also updated in the same manner. For the selected index  $i$ , we use the method similar to that of updating the upper boundary to determine an appropriate step size  $d$  which subjects to  $\text{mp}(l_i^l, u_i^l) \geq \epsilon$ , where  $l_i^l = (x_1^l, \dots, x_{i-1}^l, x_i^l - d, x_{i+1}^l, \dots, x_m^l)$  and  $u_i^l = (x_1^u, \dots, x_{i-1}^u, x_i^l, x_{i+1}^u, \dots, x_m^u)$ . Then, we set  $x_i^l = x_i^l - d$ . The lower boundary of the  $i$ th design parameter is updated.

Thus, we begin with an initial point  $\mathbf{x}_0$  and iteratively update the boundaries of each design parameter. The iteration is terminated, when each design parameter reaches the boundary of the reliable operating region.

Note that we extend the boundaries of the hyperrectangle by a relatively small step size, especially when the performance function  $h(\mathbf{x})$  is multimodal. This setting has the advantage that the values of  $\text{mp}(l_i^u, u_i^u)$  and  $\text{mp}(l_i^l, u_i^l)$  calculated by the interior-point algorithm have a high possibility to be the global minimum of  $r(\mathbf{x}) - \alpha$  on the extended hyperrectangles  $[l_i^u, u_i^u]$  and  $[l_i^l, u_i^l]$ , respectively, and thus ensures the obtained hyperrectangle is indeed contained in the reliable operating region.

The order in which the updates of the design parameters are performed, however, may have an effect on the result. Therefore, we use all possible permutations of  $1, 2, \dots, m$  to give  $m!$  different orders, where  $m!$  is the factorial of the number  $m$ . In the outer loop, we loop over  $m!$  different orders (index  $t$ ). In the inner loop, the boundaries of the hyperrectangle are adjusted one-by-one in order  $o_t$  until they cannot be extended any further. Thus, we obtain  $m!$  hyperrectangles, and the one with the maximum volume is chosen as the resulting hyperrectangle.

A formal description of the reliable hyperrectangle algorithm, including some more details that are important to the following analysis of its convergence, is given in Figure 2.

**3.2. Convergence Analysis.** In this section, we give the convergence analysis of the proposed algorithm under the assumption that the performance function  $h(\mathbf{x})$  is continuous. First, we have the following theorem, and the detailed proof is given in Supplementary Material (available here).

**Theorem 1.**  $\text{mp}(l_i^u, u_i^u)$  is a single variable function of  $d$  and it is continuous for all  $d \geq 0$  under the assumption  $h(\mathbf{x})$  is continuous.

In the reliable hyper-rectangle algorithm, the outer loop loops over all  $m!$  different orders, therefore the convergence of algorithm is proved as long as the inner loop is terminated after finite iterations. In the inner loop, we conclude that the  $i$ th design parameter is removed from the index set  $I^u$  if either of the following two conditions is satisfied. The first one is that it reaches the bounds of the design space. And the second one is that it reaches the boundary of reliability constraints.

We show that either of the two termination conditions is satisfied within finite iterations. From Theorem 1,  $\text{mp}(l_i^u, u_i^u)$  is a single variable function of  $d$ . Moreover, by the definitions of  $l_i^u$  and  $u_i^u$ , the value of  $\text{mp}(l_i^u, u_i^u)$  at  $d = 0$  is always greater than or equal to  $\epsilon$ . If the value of  $\text{mp}(l_i^u, u_i^u)$  at  $d = d_0$  is below  $\epsilon$ , then, due to the continuity of  $\text{mp}(l_i^u, u_i^u)$ , according to Intermediate Value Theorem [31], there exist some points  $d$  between 0 and  $d_0$  such that  $\text{mp}(l_i^u, u_i^u) = \epsilon$ . Thus, the second termination condition is met after finite iterations. Otherwise, the first termination condition is met after  $(x_i^u - x_{0i})/d_0$  iterations.

Similarly, the  $i$ th design parameter is removed from the index set  $I^l$  after finite iterations. Therefore, the convergence of the reliable hyperrectangle algorithm is guaranteed.

**3.3. Optimality Analysis.** The order in which the updates of boundaries of design parameters are performed one-by-one, however, may have an effect on the result. To solve this problem to a certain degree, we use nested loops in the reliable hyperrectangle algorithm. The outer loop loops over all  $m!$  different orders. The inner loop updates the boundaries of the hyperrectangle one-by-one in order  $o_t$  until they cannot be extended any further. Thus, the boundaries of the hyperrectangle obtained and the one with the maximum volume is chosen as the resulting hyper-rectangle. In this sense, the result of this algorithm is locally suboptimal.

**3.4. Extensions.** In this section, we consider the extension of the proposed algorithm. First, we consider the extension of this algorithm to deal with multiple performances. We assume  $p$  performances are defined by

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) + \boldsymbol{\varepsilon}, \quad (11)$$

where  $\mathbf{y} = (y_1, \dots, y_p)$ ,  $\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), \dots, h_p(\mathbf{x}))$ , and  $\boldsymbol{\varepsilon} = (\varepsilon_1, \dots, \varepsilon_p)$ . Given performance specifications  $[\mathbf{y}^l, \mathbf{y}^u]$ , the equation  $r(\mathbf{x})$  in problem (9) can be rewritten as

$$\begin{aligned} r(\mathbf{x}) &= P(\mathbf{y}^l \leq \mathbf{y} \leq \mathbf{y}^u) = P(\mathbf{y}^l - \mathbf{h}(\mathbf{x}) \leq \boldsymbol{\varepsilon} \leq \mathbf{y}^u - \mathbf{h}(\mathbf{x})) \\ &= \int_{\mathbf{y}^l - \mathbf{h}(\mathbf{x}) \leq \boldsymbol{\varepsilon} \leq \mathbf{y}^u - \mathbf{h}(\mathbf{x})} f(\boldsymbol{\varepsilon}) d\boldsymbol{\varepsilon}, \end{aligned} \quad (12)$$

where  $f(\cdot)$  is the joint probability density function of random errors.

Next, we consider the extension of the proposed algorithm to deal with the design space constraints. We assume the design space is constrained by  $q$  inequalities, that is,

---

**Input:** Given  $x_0, x^l, x^{uu}, d_0, d_\epsilon$  and  $\epsilon$   
**Output:** The upper boundary  $x^{uu}$  and the lower boundary  $x^{ll}$

```

1 Initialize  $o_t, t = 1, \dots, m!$ ; // initialize  $m!$  different orders
2 for  $t = 1, 2, \dots, m!$  do // the outer loop: traverse  $m!$  different orders
3    $x^{uu} = x^{ll} = x_0, I^u = o_t, I^l = o_t$ ;
4   while  $I^l \neq \emptyset$  and  $I^u \neq \emptyset$  do
5     for  $i$  in  $o_t$  do
6       /* the inner loop: update the boundaries of design
7         parameters one-by-one */
8       if  $i$  in  $I^u$  then // update the upper boundary
9          $d = d_0$ ;
10        if  $mp(I_i^u, u_i^u) \geq \epsilon$  then
11           $x_i^{uu} = x_i^{uu} + d$ ;
12          if  $x_i^{uu} > x_i^u$  then
13             $x_i^{uu} = x_i^u$ ; Remove  $i$  from  $I^u$ ;
14          end
15        else
16           $d_l = 0; d_u = d_0$ ;
17          while  $d_u - d_l > d_\epsilon$  do
18             $d = (d_u + d_l)/2$ ;
19            if  $mp(I_i^u, u_i^u) > \epsilon$  then
20               $d_l = d$ ;
21            else
22               $d_u = d$ ;
23            end
24          end
25          Remove  $i$  from  $I^u$ ;
26        end
27      end
28      if  $i$  in  $I^l$  then // update the lower boundary
29         $d = d_0$ ;
30        if  $mp(I_i^l, u_i^l) < \epsilon$  then
31           $d_l = 0; d_u = d_0$ ;
32          while  $d_u - d_l > d_\epsilon$  do
33             $d = (d_u + d_l)/2$ ;
34            if  $mp(I_i^l, u_i^l) > \epsilon$  then
35               $d_l = d$ ;
36            else
37               $d_u = d$ ;
38            end
39          end
40          Remove  $i$  from  $I^l$ ;
41        end
42         $x_i^{ll} = x_i^{ll} - d$ ;
43        if  $x_i^{ll} < x_i^l$  then
44           $x_i^{ll} = x_i^l$ ; Remove  $i$  from  $I^l$ ;
45        end
46      end
47    end
48  end
49   $V_t = \prod_{i=1}^m (x_i^{uu} - x_i^{ll}), x_t^{uu} = x^{uu}, x_t^{ll} = x^{ll}$ ;
50   $r = \operatorname{argmax}_{t=1, \dots, m!} V_t, x^{uu} = x_r^{uu}, x^{ll} = x_r^{ll}$ ;

```

---

FIGURE 2: Pseudocode of the reliable hyperrectangle algorithm is adjusted for all dimensions one-by-one in  $m!$  different orders and then  $m!$  hyperrectangles are adjusted.



$g_j(\mathbf{x}) \geq 0, j = 1, \dots, q$ . Under this situation, problem (9) can be formulated as follows:

$$\begin{aligned} \max_{\mathbf{x}^l, \mathbf{x}^u} \quad & \prod_{i=1}^m (x_i^{uu} - x_i^{ll}), \\ \text{subject to} \quad & \text{mpg}(\mathbf{x}^l, \mathbf{x}^u) \geq 0, \\ & \mathbf{x}^l \leq \mathbf{x} \leq \mathbf{x}^u, \end{aligned} \quad (13)$$

where

$$\text{mpg}(\mathbf{x}^l, \mathbf{x}^u) = \min\{\text{mp}(\mathbf{x}^l, \mathbf{x}^u), \text{mg}(\mathbf{x}^l, \mathbf{x}^u)\}, \quad (14)$$

$$\text{mg}(\mathbf{x}^l, \mathbf{x}^u) = \min_{\mathbf{x}^l \leq \mathbf{x} \leq \mathbf{x}^u} \min_{j=1, \dots, q} g_j(\mathbf{x}), \quad (15)$$

and  $\text{mp}(\mathbf{x}^l, \mathbf{x}^u)$  is the same as that in equation (10). We can employ the reliable hyperrectangle algorithm to tackle this problem similarly.

**3.5. Initial Point.** In the practical engineering application, engineering personnel's aim is to gain a maximum reliable hyperrectangle around the design points where the performances are equal to their target values, so it is appropriate to consider this design point as the initial point.

However, there are usually multiple design points where the performances are equal to their target values. Some of those are near the boundary of the reliable operating region, while others are away from the boundary of the reliable operating region. The design point which sits near the boundary is unsuitable as initial point. Here, we use the tolerance design to reduce the size of the reliable operating region and obtain a design point which not only minimizes deviation between the performance and their target values but also sits away from the boundary of the original reliable operating region.

Tolerance design pays attention to how variations in design parameters are transmitted to the constraint functions [32]. In general, the maximum variation of design parameter  $i$  is denoted by  $\Delta x_i$ . The effect of variations on a constraint function can be estimated from a first-order Taylor's series as follows:

$$\Delta g_j = \sum_{i=1}^m \left| \frac{\partial g_j}{\partial x_i} \Delta x_i \right|, \quad (16)$$

where  $\Delta g_j$  represents the constraint variation transmitted to constraint function  $g_j$ . If the value of  $\Delta x_i$  is not given by engineers, then we set  $\Delta x_i = (1/6)(x_i^{uu} - x_i^{ll})$ .

Tolerance design addresses both the feasibility robustness [32] and the reliability robustness [33]. Feasibility robustness can be developed by increasing the value of the constraint function during optimization by the amount of the constraint variation. Reliability robustness can be achieved by improving the threshold of reliability. Therefore, the tolerance design model can be expressed as follows:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \max_{i=1, \dots, p} (h_i(\mathbf{x}) - \gamma_{0i})^2 \\ \text{subject to} \quad & r(\mathbf{x}) \geq (1 + \eta)\alpha, \\ & g_j(\mathbf{x}) - \Delta g_j \geq 0, j = 1, \dots, q, \\ & \text{mp}(\mathbf{x}, \mathbf{x}) \geq \epsilon, \\ & \mathbf{x}^l \leq \mathbf{x} \leq \mathbf{x}^u, \end{aligned} \quad (17)$$

where  $\eta$  is a given positive number. This constrained optimization problem can be solved by the common algorithms, such as interior-point method. The solution is used as the initial point of the reliable hyperrectangle, denoted by  $\mathbf{x}_0$ .

However, in the case where the target value of the performance is not given, it is reasonable to take the center point of the design space as the initial point if the center point is feasible and reliable.

### 3.6. Process of the Reliable Hyperrectangle Algorithm.

According to the abovementioned analysis, the steps of using the reliable hyperrectangle algorithm to design a reliable hyperrectangle can be described briefly as follows:

Step 1: to set the parameters, the lower and upper bounds of the design space,  $\mathbf{x}^l$  and  $\mathbf{x}^u$ , and the initial step size  $d_0$  are set for each problem individually. Additionally, here we set default parameters  $\epsilon = 10^{-4}$  and  $d_e = 10^{-3}$ .

Step 2: to obtain the initial point, by means of the interior-point method to solve model (17), we obtain the initial point  $\mathbf{x}_0$ .

Step 3: to obtain the maximum hyperrectangle, by means of the reliable hyperrectangle algorithm shown in Figure 2, we obtain the desired hyperrectangle.

## 4. Numerical Examples and Comparisons

There is recent literature on the same topic considering uncertainties [5, 20, 21]. Therefore, for comparison purpose, certainties is not considered in this section, that is, we aim to find the maximum hyperrectangle in the feasible operating region. More specifically, two numerical examples in [21] are considered to compare the solution of the proposed algorithm with the solution of the method in [21] (we denote this method by "GHZ method" hereafter).

- (1) In Example 1, the feasible operating region is constrained by a convex polytope in two dimensions
- (2) In Example 2, the feasible operating region is constrained by a tilted hyperplane which divides the  $l$ -dimensional unit cube into two equal volumes

### 4.1. Numerical Example 1: A Convex Polytope as Boundary.

Let  $\mathbf{x}^l = (0, 0)$  and  $\mathbf{x}^u = (1, 1)$  be the lower and upper bounds of the design space, respectively, and

$$\mathbf{h}(\mathbf{x}) = \begin{pmatrix} h_1(\mathbf{x}) \\ h_2(\mathbf{x}) \\ h_3(\mathbf{x}) \\ h_4(\mathbf{x}) \\ h_5(\mathbf{x}) \\ h_6(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \frac{1}{8}x_1 + \frac{1}{4}x_2 \\ \frac{4}{17}x_1 + \frac{2}{17}x_2 \\ -\frac{1}{2}x_1 + \frac{1}{2}x_2 \\ -\frac{1}{2}x_1 - \frac{1}{3}x_2 \\ \frac{1}{3}x_1 - \frac{2}{3}x_2 \\ x_1 - \frac{3}{2}x_2 \end{pmatrix}, \quad (18)$$

$$\mathbf{y}^u = \begin{pmatrix} 1 \\ 1 \\ 1 \\ -1 \\ -1 \\ 1 \end{pmatrix}.$$

The particularity of this problem is that  $h_i(\mathbf{x})$  is monotonous, thus, the set

$$\left\{ (\mathbf{x}^l, \mathbf{x}^{uu}) : \min_{\mathbf{x}^l \leq \mathbf{x} \leq \mathbf{x}^{uu}} \min_{i=1, \dots, 6} (y_i^u - h_i(\mathbf{x})) \geq 0 \right\}, \quad (19)$$

is the same as the set

$$\left\{ (\mathbf{x}^l, \mathbf{x}^{uu}) : \mathbf{y}^u - A(\mathbf{x}^l, \mathbf{x}^{uu})^T \geq 0, \quad \mathbf{x}^l \leq \mathbf{x} \leq \mathbf{x}^{uu} \leq \mathbf{x}^u \right\}, \quad (20)$$

where

$$A = \begin{pmatrix} 0 & 0 & \frac{1}{8} & \frac{1}{4} \\ 0 & 0 & \frac{4}{17} & \frac{2}{17} \\ -\frac{1}{2} & 0 & 0 & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{3} & 0 & 0 \\ -\frac{1}{3} & \frac{2}{3} & 0 & 0 \\ 0 & -\frac{3}{2} & 1 & 0 \end{pmatrix}, \quad (21)$$

and  $T$  denotes the transpose of a matrix.

Therefore, the optimization problem (3) can be transformed into the following constrained optimization problem:

$$\begin{aligned} \max_{\mathbf{x}^l, \mathbf{x}^{uu}} \quad & \prod_{i=1}^2 (x_i^{uu} - x_i^l), \\ \text{subject to} \quad & \mathbf{y}^u - A(\mathbf{x}^l, \mathbf{x}^{uu})^T \geq 0, \\ & \mathbf{x}^l \leq \mathbf{x} \leq \mathbf{x}^{uu} \leq \mathbf{x}^u. \end{aligned} \quad (22)$$

This is a quadratic optimization problem under inequality constraints. The exact solution is found by means of Lagrange multipliers and is shown in the second row of Table 1.

The solution of the GHZ method and the result of this paper are also shown in the third and fourth rows of Table 1, respectively. The column entitled ‘‘Volume’’ contains the volume of hyperrectangle, and the column entitled ‘‘Volume Error’’ contains the relative error between the volume and the exact volume.

According to Table 1, the obtained hyperrectangle by our method is in close proximity to the exact result. It is easy to verify that all points in the obtained hyperrectangle are contained in the feasible operating region. Moreover, the operation time of our method is 14.68 seconds (using MATLAB R2016b software in a windows platform with Intel Core i7-4790 CPU 3.60 GHz, 8 cores, 8 GB RAM). For a small number of design parameters, the operation time of our method is less than that of the GHZ method, which uses the Monte Carlo sample simulating 100 times. The volume of hyperrectangle by our method is larger than that by the GHZ method, and volume error by our method is smaller than that by the GHZ method.

#### 4.2. Numerical Example 2: A Tilted Hyperplane as Boundary.

We consider now an  $l$ -dimensional problem with a tilted hyperplane as boundary. The exact solution is computed and the algorithm’s behavior for different  $l$  is studied.

Let us consider the  $l$ -dimensional unit cube  $[0, 1]^l$  as the design space. The tilted hyperplane is described by the following equation:

$$h(\mathbf{x}) = \sum_{i=1}^l x_i - \frac{l}{2}. \quad (23)$$

The associated optimization problem for the hyperrectangle with the lower boundaries  $x_i^l = 0$  and the upper boundaries  $x_i^{uu}$ ,  $i = 1, 2, \dots, l$ , reads as

$$\begin{aligned} \max_{\mathbf{x}^{uu}} \quad & \prod_{i=1}^l x_i^{uu}, \\ \text{subject to} \quad & \min_{0 \leq x_i^{uu} \leq 1} -h(\mathbf{x}) \geq 0. \end{aligned} \quad (24)$$

Its exact solution is given by  $\mathbf{x}^{uu} = [1/2, \dots, 1/2]$ , and the exact hyperrectangle volume is  $2^{-l}$ . For  $l = 2$  and  $l = 3$ , the solutions of the GHZ method and the results of our method are shown in Tables 2 and 3, respectively.

TABLE 1: Comparison of exact solution of Example 1, GHZ method result, and this paper result.

	$x_1^{ll}$	$x_1^{uu}$	$x_2^{ll}$	$x_2^{uu}$	Volume	Volume error (%)
Exact method	1.180	2.850	1.230	2.580	2.2545	—
GHZ method	1.174	2.735	1.235	2.633	2.1823	3.200
Our method	1.232	2.750	1.168	2.614	2.1950	2.640

TABLE 2: Comparison of exact solution of Example 2 when  $l = 2$ , GHZ method result, and our method result.

$l = 2$	$x_1^{uu}$	$x_2^{uu}$	Volume	Volume error (%)
Exact method	0.500	0.500	0.2500	—
GHZ method	0.479	0.523	0.2505	0.200
Our method	0.500	0.500	0.2500	0.000

TABLE 3: Comparison of exact solution of Example 2 when  $l = 3$ , GHZ method result, and our method result.

	$x_1^{uu}$	$x_2^{uu}$	$x_3^{uu}$	Volume	Volume error (%)
Exact method	0.500	0.500	0.500	0.1250	—
GHZ method	0.515	0.508	0.519	0.1358	8.600
Our method	0.500	0.500	0.500	0.1250	0.000

According to Tables 2 and 3, the result by our method is in close proximity to the exact result, so it indicates that the proposed method is reasonable. Compared to the result by the GHZ method, our method has smaller volume errors. Moreover, all points in the hyperrectangle obtained by our method are contained in the feasible operating region; however, some points in the hyperrectangle obtained by the GHZ method are not included in the feasible operating region.

Overall, the proposed method has a low time complexity and performs well with small volume errors. Numerical examples show that our method has a better performance than the GHZ method and thus is a good approach for seeking out the maximum hyperrectangle.

## 5. Case Studies

In this section, with two engineering examples, the capability of our method to seek maximum hyperrectangle in the reliable operating region is illustrated. The first example indicates how our approach deals with multiple performances. The second example shows how our approach handles with a multimodal function.

Because the design parameters may have different units, we normalize each design parameter before applying our proposed approach. Moreover, engineers set normally the reliability threshold  $\alpha = 0.95$ .

*5.1. Case Example 1: The Planetary Gear.* There are two main performances for the planetary gear: bending fatigue and contact fatigue [34]. The bending fatigue strength  $R_1$  and bending stress  $S_1$  have the following expressions [35]:

$$\begin{aligned} R_1 &= 1.956x_1x_2, \\ S_1 &= 0.0174x_3x_4, \end{aligned} \quad (25)$$

respectively, where  $x_1$  is bending fatigue limit stress of gear tooth (in  $N/mm^2$ ),  $x_2$  is life factor of the gear,  $x_3$  is the tangential force on the inner part of the gear face (in  $N/mm^2$ ), and  $x_4$  is the tooth form factor.

Similarly, the contact fatigue strength  $R_2$  and contact stress  $S_2$  can be expressed as follows [35]:

$$\begin{aligned} R_2 &= 0.8526x_5x_6, \\ S_2 &= 9\sqrt{x_3x_7}, \end{aligned} \quad (26)$$

respectively, where  $x_5$  is the contact fatigue strength limit (in  $N/mm^2$ ),  $x_6$  is the lubricating factor,  $x_7$  is the dynamic factor, and  $x_3$  is the same as above.

The bend fatigue  $y_1$  and contact fatigue  $y_2$  are two performances we consider. Performance functions for bend fatigue and contact fatigue are

$$\begin{aligned} h_1(\mathbf{x}) &= \frac{1.956x_1x_2}{0.0174x_3x_4}, \\ h_2(\mathbf{x}) &= \frac{0.8526x_5x_6}{9\sqrt{x_3x_7}}, \end{aligned} \quad (27)$$

respectively. It is considered reliable, provided the following conditions hold:  $y_1 \geq 1$  and  $y_2 \geq 1$ . That is,  $\mathbf{y}^l = (1, 1)$ . Moreover, based on the safety margin, engineers set the target values of these two performances as  $\mathbf{y}_0 = (y_{01}, y_{02}) = (3, 3)$ . The second row of Table 4 shows the design space in terms of the bounds of each design parameter.

Besides, there are two constraints: the bending strength must be greater than  $1200 N/mm^2$  and the contact strength must be greater than  $1600 N/mm^2$ . Therefore, constraint functions are expressed as

$$\begin{aligned} g_1(\mathbf{x}) &= 1.956x_1x_2 - 1200, \\ g_2(\mathbf{x}) &= 0.8526x_5x_6 - 1600. \end{aligned} \quad (28)$$

The reliable hyperrectangle design is studied under the assumption that random errors exist and follow normal distribution. First, the tolerance model (17) is developed to get the initial point  $\mathbf{x}_0$ . Next, the problem of searching for a reliable hyperrectangle is transformed into problem (13), where

$$\begin{aligned} \text{mg}(\mathbf{x}^{ll}, \mathbf{x}^{uu}) &= \min_{\mathbf{x}^{ll} \leq \mathbf{x} \leq \mathbf{x}^{uu}} \min_{j=1,2} g_j(\mathbf{x}), \\ \text{mp}(\mathbf{x}^{ll}, \mathbf{x}^{uu}) &= \min_{\mathbf{x}^{ll} \leq \mathbf{x} \leq \mathbf{x}^{uu}} r(\mathbf{x}) - \alpha, \\ r(\mathbf{x}) &= P(\mathbf{h}(\mathbf{x}) + \boldsymbol{\varepsilon} \geq \mathbf{1}), \end{aligned} \quad (29)$$

$g_1(\mathbf{x})$  and  $g_2(\mathbf{x})$  are defined by equation (28),  $\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), h_2(\mathbf{x}))$ ,  $\mathbf{1} = (1, 1)$ ,  $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2) \sim N(\mathbf{0}, \boldsymbol{\Sigma})$ , and



TABLE 4: The bounds of design parameters.

	$x_1$ (N/mm <sup>2</sup> )	$x_2$	$x_3$ (N/mm <sup>2</sup> )	$x_4$	$x_5$ (N/mm <sup>2</sup> )	$x_6$	$x_7$
Design space	[606, 1039]	[0.784, 1.107]	[7700, 14300]	[1.253, 3.043]	[1531, 2452]	[0.985, 1.212]	[1.264, 1.764]
Reliable hyperrectangle <sup>1</sup>	[753, 1039]	[0.847, 1.107]	[7700, 10096]	[1.253, 2.811]	[2214, 2452]	[1.088, 1.212]	[1.264, 1.464]

<sup>1</sup>The initial point of the reliable hyperrectangle is (903, 0.896, 9436, 2.543, 2352, 1.122, 1.364), and the volume is 1.641e+06.

$\Sigma = \text{diag}(0.5^2, 0.5^2)$ . The results of the proposed algorithm are shown in the third row of Table 4 in terms of the bounds of each design parameter.

Monte Carlo sampling [36] is used to verify whether the obtained reliable hyperrectangle satisfies the reliability requirement in the presence of random errors. We firstly sample some design points which are randomly and uniformly distributed over the obtained reliable hyperrectangle, then sample 1000 random errors from the normal distribution  $N(0, \Sigma)$  finally compute the coverage probability of each sampled design point. The coverage probability is the proportion that output performances lie within the performance specifications, where the output performances are the sum of the nominal performances and random errors. Due to space restrictions, part of simulation results is shown in Table 5. For each sampled design point, its coverage probability is higher than the reliability threshold (0.95); therefore, its reliability is validated.

**5.2. Case Example 2: Diamond Turning Process.** The surface roughness in the diamond turning process is shown in Figure 3. The waveform left on the finished surface is mainly due to the tool nose geometry and feed [37]. A performance function for the surface roughness with three design parameters is given by

$$h(\mathbf{x}) = \frac{x_1^2}{8x_2} + \frac{x_3}{2} \left(1 + \frac{x_2 x_3}{2}\right), \quad (30)$$

where  $x_1$ ,  $x_2$ , and  $x_3$  denote the feed in microns/revolution ( $\mu\text{m}/\text{rev}$ ), the tool nose radius in microns ( $\mu\text{m}$ ), and the minimum undeformed chip thickness in microns ( $\mu\text{m}$ ), respectively.

The output performance is expressed as

$$y = h(\mathbf{x}) + \varepsilon, \quad (31)$$

where  $\varepsilon$  is the random error and follows the normal distribution with mean zero and unknown variance  $\sigma^2$ . The random error comes from one measurement to the next. The random error tends to be normally distributed when it is the sum of many small and independent random errors because of the central limit theorem [38]. The target value of the output performance is  $3.5 \mu\text{m}$ . The performance specification requires that  $y$  is between 2 and  $5 \mu\text{m}$ . The second row of Table 6 shows the design space in terms of the bounds of each design parameter.

The performance function (30) is developed based on some simplified assumptions and may be not accurate; thus, we use experiment data to improve the performance function. Figure 4 shows the experiment data from the microturning of an aluminium alloy using a diamond tool with nose radius of 800 microns and the minimum undeformed chip thickness of 0.02

TABLE 5: Reliability and coverage probability of the sampled design point.

Design point	Reliability	Coverage
(818, 1.107, 7700, 2.811, 2214, 1.212, 1.264)	0.9992	1.0000
(842, 1.075, 8003, 2.638, 2240, 1.198, 1.281)	0.9988	0.9990
(867, 1.042, 8306, 2.465, 2267, 1.184, 1.297)	0.9981	0.9970
(891, 1.010, 8609, 2.292, 2293, 1.171, 1.314)	0.9973	0.9970
(916, 0.977, 8912, 2.119, 2320, 1.157, 1.331)	0.9962	0.9970
(940, 0.945, 9214, 1.946, 2346, 1.143, 1.347)	0.9948	0.9960
(965, 0.912, 9517, 1.773, 2372, 1.129, 1.364)	0.9930	0.9950
(989, 0.880, 9820, 1.599, 2399, 1.116, 1.381)	0.9907	0.9940
(1015, 0.847, 9950, 1.426, 2426, 1.102, 1.397)	0.9870	0.9880
(1039, 0.815, 10096, 1.253, 2452, 1.088, 1.414)	0.9856	0.9870

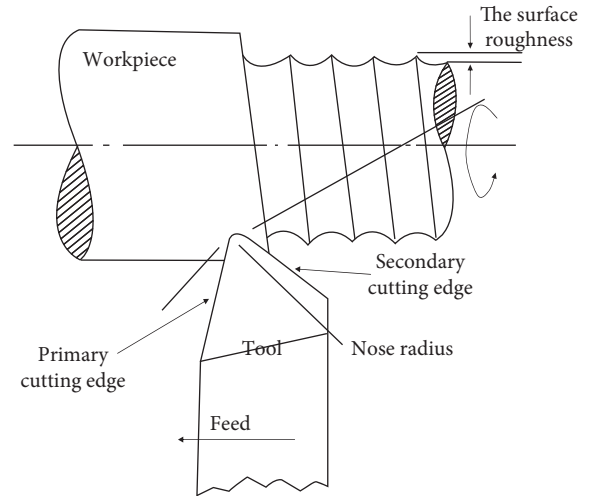


FIGURE 3: Illustration of turning operation showing surface roughness left on the finished surface.

microns [39]. We have 20 measured values of surface roughness at each of the experiment points and therefore can obtain an estimate of  $\sigma^2$ . We obtain  $\sigma^2 = 0.2191$ . The estimate derived from the performance function and the experiment data can be computed as

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n (y_i - h(\mathbf{x}_i))^2 = 9.0563, \quad (32)$$

which is much larger than 0.2191. This shows that the performance function is not good.

We use the experiment data to adjust the performance function. Based on the plot of residuals from the performance function against  $x_1$ , and after some experimentation, we choose  $\mu_1(x_1) = 1$ ,  $\mu_2(x_1) = x_1$ , and  $\mu_3(x_1) = \log(1 + x_1)$ .

TABLE 6: The bounds of design parameters.

	$x_1$ ( $\mu\text{m}/\text{rev}$ )	$x_2$ ( $\mu\text{m}$ )	$x_3$ ( $\mu\text{m}$ )
Design space	[5.0000, 100.0000]	[300.0000, 1000.0000]	[0.0067, 0.0224]
Reliable hyperrectangle <sup>1</sup>	[5.0000, 85.3528]	[681.0000, 1000.0000]	[0.0118, 0.0224]

<sup>1</sup>The initial point of the reliable hyperrectangle is (70.1528, 800.0000, 0.0200), and the volume is 272.9558.

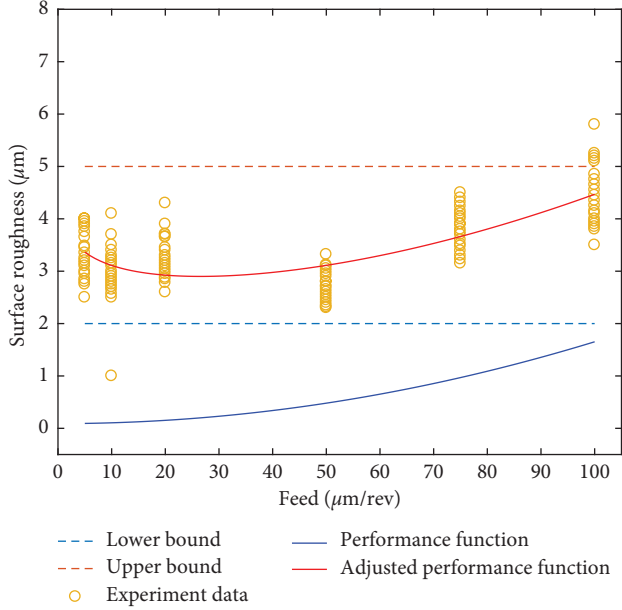


FIGURE 4: Comparison of the performance function and the adjusted performance function.

Thus, the adjusted performance function for the surface roughness is given by

$$y(\mathbf{x}, \boldsymbol{\theta}) = \frac{x_1^2}{8x_2} + \frac{x_3}{2} \left( 1 + \frac{x_2 x_3}{2} \right) + \theta_1 + \theta_2 x_1 + \theta_3 \log(1 + x_1). \quad (33)$$

In [40], an optimum design point was discussed and  $\boldsymbol{\theta}$  was estimated by the Bayesian approach. Our aim is to find a reliable operating region; thus, it is more appropriate to use frequentist procedures which are computationally efficient and simple to implement. Here, the maximum likelihood method is used to estimate the unknown parameters. The likelihood function is expressed as

$$L_n(\boldsymbol{\theta}, \sigma) = \frac{1}{(2\pi)^{60} \sigma^{120}} \exp\left(-\frac{\sum_{i=1}^{120} (y(\mathbf{x}_i, \boldsymbol{\theta}) - y_i)^2}{2\sigma^2}\right). \quad (34)$$

The maximum likelihood estimate of  $\sigma^2$  is  $\hat{\sigma}^2 = 0.2740$ , which is much smaller than that of the performance function (9.0563) and only slightly larger than  $\sigma^2 = 0.2191$ . This implies the adjusted performance function gives a good fit (see Figure 4).

Now we can carry out the proposed algorithm based on the adjusted performance function, where  $\sigma$  is estimated by  $\hat{\sigma}$ . The reliable hyperrectangle is listed in the third row of Table 6 in terms of the bounds of each design parameter.

We use the same method as that in Example 1 to simulate random errors and show part of results in Table 7. We see

TABLE 7: Reliability and coverage probability of the sampled design point.

Design point	Reliability	Coverage
(5.0000, 681.8360, 0.0118)	0.9947	0.9950
(13.9281, 716.2986, 0.0129)	0.9721	0.9735
(22.8562, 751.7613, 0.0141)	0.9586	0.9603
(31.7843, 787.2240, 0.0153)	0.9607	0.9626
(40.7124, 822.6866, 0.0165)	0.9715	0.9727
(49.6405, 858.1493, 0.0176)	0.9834	0.9850
(58.5686, 893.6120, 0.0188)	0.9921	0.9937
(67.4966, 929.0747, 0.0200)	0.9965	0.9969
(76.4247, 964.5373, 0.0212)	0.9970	0.9967
(85.3528, 1000.0000, 0.0224)	0.9922	0.9914

that coverage probabilities are higher than the reliability threshold (0.95). It is obvious that the adjusted performance function (33) is a multimodal function; therefore, the ability of the proposed algorithm in dealing with multimodal functions is confirmed.

Moreover, minimizing the deviation between the performance and its target, that is,  $\min_{\mathbf{x} \in \mathcal{X}^{\text{reliable}}} (y(\mathbf{x}, \hat{\boldsymbol{\theta}}) - y_0)^2$ , is considered as the objective function to find the optimum point inside the proposed reliable design operating region. The obtained optimum point is (69.6664, 800.0171, 0.0173). We can see that the optimum point is close to the initial point.

Besides, we seek the permissible reliable range of  $x_1$  under the condition that the values of  $x_2$  and  $x_3$ , as in the physical experiment shown in Figure 4, are fixed at 800 and 0.0200. The obtained reliable range is [5.0000, 91.6428]. We see that all but the last of those experiment points are included in this range. This result agrees with the experiment because 5 of 20 measured surface roughness of the last experiment point are larger than the upper bound of surface roughness and therefore the last experiment point is unreliable.

## 6. Conclusions

This paper focuses on the design of the reliable hyperrectangle. An iterative algorithm which updates the sides of hyperrectangle one-by-one is proposed to search the reliable hyperrectangle. We have proven the convergence of the proposed algorithm. Moreover, we extend the proposed algorithm to deal with design space constraints. Besides, we have discussed how to determine the initial point. Numerical examples show that our method has a better performance than the GHZ method in [21]. Some engineering cases show that the proposed method can guarantee product reliability. This has validated the effectiveness of our approach.

## Data Availability

The digital data supporting this article are from previously reported studies and datasets, which have been cited.

## Conflicts of Interest

The authors declare no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 71801196, National Major Project from the Shanghai Nuclear Engineering Research and Design Institute under Grant 2017ZX06002006, and China Postdoctoral Science Foundation under Grant 2018M631606.

## Supplementary Materials

Proof of Theorem 1. (*Supplementary Materials*)

## References

- [1] A. Saidane, "Optimal reliability design: fundamentals and applications," *Microelectronics Journal*, vol. 32, no. 10, p. 911, 2001.
- [2] Y. M. Lu, X. F. Wang, and R. Xie, "Derivation of the mathematical approach to the radial pump's meridional channel design based on the controlment of the medial Axis," *Mathematical Problems in Engineering*, vol. 2017, pp. 15–30, 2017.
- [3] J. Jiang, G. F. Ding, J. Zhang, Y. S. Zou, and S. F. Qin, "A systematic optimization design method for complex mechatronic products design and development," *Mathematical Problems in Engineering*, vol. 2018, Article ID 3159637, 14 pages, 2018.
- [4] D. M. Frangopol, "Sensitivity of reliability-based optimum design," *Journal of Structural Engineering*, vol. 111, no. 8, pp. 1703–1721, 1985.
- [5] C. M. Rocco, J. A. Moreno, and N. Carrasquero, "Robust design using a hybrid-cellular-evolutionary and interval-arithmetic approach: a reliability application," *Reliability Engineering & System Safety*, vol. 79, no. 2, pp. 149–159, 2003.
- [6] F. Li, T. Wu, A. Badiru, M. Hu, and S. Soni, "A single-loop deterministic method for reliability-based design optimization," *Engineering Optimization*, vol. 45, no. 4, pp. 435–458, 2013.
- [7] G. W. Stockdale and A. Cheng, "Finding design space and a reliable operating region using multivariate bayesian approach with experimental design," *Quality Technology & Quantitative Management*, vol. 6, no. 4, pp. 18–23, 2009.
- [8] X. Du and W. Chen, "Sequential optimization and reliability assessment method for efficient probabilistic design," *Journal of Mechanical Design*, vol. 126, no. 2, pp. 871–880, 2004.
- [9] Y. Liu, H. K. Jeong, and M. Collette, "Efficient optimization of reliability-constrained structural design problems including interval uncertainty," *Computers & Structures*, vol. 177, pp. 1–11, 2016.
- [10] A. Bellotti, "Reliable region predictions for automated valuation models," *Annals of Mathematics and Artificial Intelligence*, vol. 81, no. 1-2, pp. 71–84, 2017.
- [11] Z. L. Huang, C. Jiang, Z. Zhang, T. Fang, and X. Han, "A decoupling approach for evidence-theory-based reliability design optimization," *Structural and Multidisciplinary Optimization*, vol. 56, no. 3, pp. 647–661, 2017.
- [12] M. Lin, Y. W. Pan, L. T. Yang, and M. Y. Guo, "Scheduling co-design for reliability and energy in cyber-physical systems," *IEEE Transactions on Emerging Topics in Computing*, vol. 1, no. 2, pp. 353–365, 2013.
- [13] Z. L. Huang, C. Jiang, Y. S. Zhou, J. Zheng, and X. Y. Long, "Reliability-based design optimization for problems with interval distribution parameters," *Structural and Multidisciplinary Optimization*, vol. 55, no. 2, pp. 513–528, 2017.
- [14] S. Shan and G. G. Wang, "Reliable design space and complete single-loop reliability-based design optimization," *Reliability Engineering & System Safety*, vol. 93, no. 8, pp. 1218–1230, 2008.
- [15] M. Milanese, J. Norton, and H. Piet-Lahanier, *Bounding Approaches to System Identification*, Plenum Press, New York, USA, 1998.
- [16] T. Gao and J. L. Li, "A derivative-free trust-region algorithm for reliability-based optimization," *Structural and Multidisciplinary Optimization*, vol. 55, no. 4, pp. 1535–1539, 2017.
- [17] M. Lehar and M. Zimmermann, "An inexpensive estimate of failure probability for high-dimensional systems with uncertainty," *Structural Safety*, vol. 36-37, pp. 32–38, 2012.
- [18] M. R. S. Claudio and E. Hernandez, "Robustness and sensitivity analysis in multiple criteria decision problems using rule learner techniques," *Reliability Engineering and System Safety*, vol. 134, pp. 297–304, 2015.
- [19] M. Beer and M. Liebscher, "Designing robust structures—a nonlinear simulation based approach," *Computers & Structures*, vol. 86, no. 10, pp. 1102–1122, 2008.
- [20] M. Zimmermann and J. E. von Hoessle, "Computing solution spaces for robust design," *International Journal for Numerical Methods in Engineering*, vol. 94, no. 3, pp. 290–307, 2013.
- [21] L. Graff, H. Harbrecht, and M. Zimmermann, "On the computation of solution spaces in high dimensions," *Structural and Multidisciplinary Optimization*, vol. 54, no. 4, pp. 811–829, 2016.
- [22] R. E. Moore, *Interval Analysis*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1966.
- [23] T. V. Thomaidis and E. N. Pistikopoulos, "Optimal design of flexible and reliable process systems," *IEEE Transactions on Reliability*, vol. 44, no. 2, pp. 243–250, 1995.
- [24] W. Graf, M. Götz, and M. Kaliske, "Computing permissible design spaces under consideration of functional responses," *Advances in Engineering Software*, vol. 117, pp. 95–106, 2018.
- [25] J. Wimmeler, D. Schramm, M. Wahle, and M. Zimmermann, "Concurrent design of vehicle tires and axles," in *Proceedings of the 6th Munich Chassis Symposium*, pp. 839–851, Munich, Germany, June 2015.
- [26] G. Fung, S. Sandilya, and R. B. Rao, "Rule extraction from linear support vector machines," in *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pp. 32–40, ACM, Chicago, IL, USA, August 2005.
- [27] L. Stripinis, R. Paulavičius, and J. Žilinskas, "Improved scheme for selection of potentially optimal hyper-rectangles in DIRECT," *Optimization Letters*, vol. 12, no. 7, pp. 1699–1712, 2018.
- [28] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, "Lipschitzian optimization without the Lipschitz constant," *Journal of Optimization Theory and Applications*, vol. 79, no. 1, pp. 157–181, 1993.
- [29] Q. Liu and W. Cheng, "A modified DIRECT algorithm with bilevel partition," *Journal of Global Optimization*, vol. 60, no. 3, pp. 483–499, 2014.

- [30] Q. Liu, G. Yang, Z. Zhang, and J. Zeng, "Improving the convergence rate of the DIRECT global optimization algorithm," *Journal of Global Optimization*, vol. 67, no. 4, pp. 851–872, 2017.
- [31] W. Rudin, *Principles of Mathematical Analysis*, McGraw-Hill, New York, USA, 1964.
- [32] L. Lewis, A. Teng, A. Parkinson, and J. Free, "A general approach for robust optimal design applied to the design of a control system," *Control and Dynamic Systems*, vol. 59, pp. 291–320, 1993.
- [33] A. A. Yassine, "Investigating product development process reliability and robustness using simulation," *Journal of Engineering Design*, vol. 18, no. 6, pp. 545–561, 2007.
- [34] Y. Zhao, D. T. Qin, and W. H. Wu, "Reliability model of shield machine cutter drive multi-stage planetary transmission system considering failure relevance," *China Mechanical Engineering*, vol. 22, no. 5, pp. 522–527, 2011.
- [35] B. Wu, "Gear reliability model and its reliability calculation," *Gear*, vol. 12, no. 6, pp. 11–14, 1988.
- [36] K. Binder, "Monte-carlo methods," in *Mathematical Tools for Physicists*, pp. 15–65, Wiley VCH, Weinheim, Germany, 2006.
- [37] W. Grzesik, "A revised model for predicting surface roughness in turning," *Wear*, vol. 194, no. 1-2, pp. 143–148, 1996.
- [38] A. Lyon, "Why are normal distributions normal?," *The British Journal for the Philosophy of Science*, vol. 65, no. 3, pp. 621–649, 2013.
- [39] K. Liu and S. N. Melkote, "Effect of plastic side flow on surface roughness in micro-turning process," *International Journal of Machine Tools and Manufacture*, vol. 46, no. 14, pp. 1778–1785, 2006.
- [40] V. R. Joseph and S. N. Melkote, "Statistical adjustments to engineering models," *Journal of Quality Technology*, vol. 41, no. 4, pp. 362–375, 2009.