IIE Transactions (2006) **38**, 1069–1083

ISSN: 0740-817X print / 1545-8830 online
DOI: 10.1080/07408170600735538

A data mining approach to study the significance of nonlinearity in multistation assembly processes

YUAN REN[1], YU DING[1,*] and SHIYU ZHOU[2]

[1] *Department of Industrial and Systems Engineering, Texas A&M University, College Station, TX 77843, USA*
E-mail: yuding@iemail.tamu.edu
[2] *Department of Industrial and Systems Engineering, University of Wisconsin—Madison, Madison, WI 53706, USA*

Received June 2005 and accepted March 2006

Linear models of multistation manufacturing processes are commonly used for variation reduction and other quality improvement purposes. Yet the nonlinear nature of variation propagation in multistation manufacturing processes makes people inevitably wonder at what point does the linear model cease to provide a reasonable approximation of the nonlinear system. This paper presents a data mining method to study the significance of nonlinearity effects in a multistation process. The data mining method consists of two major components: (i) an aggressive factor covering design, which uses a design set of affordable size to assess the significance of nonlinearity in a multistation process with hundreds of variables; (ii) a multiple-additive-regression-tree-based predictive model, which can help identify the critical, influential factors and partial dependence relationships among the factors and the response. Using the data mining approach, insights are garnered about how these critical factors affect the significance of nonlinearity in a multistation process. Decision guidelines are provided to help users decide when a nonlinear model, instead of a linear one, should be applied.

1. Introduction

Dimensional-variation control is very important for quality improvement in discrete-part manufacturing systems, which typically consist of multiple stations and operations to create sophisticated features on a product or to assemble numerous components into a complex product. Prior research (Ceglarek and Shi, 1995; Hu, 1997) has identified that dimensional variation can be caused by the fixture or other tooling elements at the stations, and that the variation increases as a product moves down a production line. The term, "stream of variation," is used to describe the propagation of variation, and it is generally accepted that the stream of variation significantly affects a product's dimensional quality.

In order to ensure the dimensional integrity of products in a multistation manufacturing system, the first step is to model the variation propagation of the process. Doing so is quite challenging because a multistation manufacturing process is usually complicated. For instance, a typical multistation assembly process in an automotive body shop will use 50 to 75 stations to assemble 150 to 250 parts into a car body (Jin and Shi, 1999).

In fact, extensive efforts have been devoted to model the stream of variation in multistation manufacturing systems. Figure 1 illustrates a typical N-station manufacturing process, where k is the station index, $\mathbf{u}_k$ and $\mathbf{x}_k$ are the tooling deviation and the part accumulated deviation at station k, respectively, $\mathbf{y}_k$ is the dimensional measurement obtained by sensors on station k, and $\mathbf{w}_k$ and $\mathbf{v}_k$ represent process and sensor noises. A station-indexed linear state space model is often used to represent the variation propagation as:

$$\mathbf{x}_k = \mathbf{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_k + \mathbf{w}_k \quad k = 1, 2, \ldots, N, \tag{1}$$

$$\mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{v}_k, \quad k \in \{1, 2, \ldots, N\}, \tag{2}$$

where $\mathbf{A}_k$ is the state transition matrix which links the part deviation states on adjacent stations, $\mathbf{B}_k$ is the input matrix which represents the effects of fixture and other tooling elements, and $\mathbf{C}_k$ is the observation matrix corresponding to the number and locations of sensors. This type of linear state space model has been applied to various manufacturing processes, such as rigid-part assembly processes (Jin and Shi, 1999; Ding *et al.*, 2000), compliant-part assembly processes (Camelio *et al.*, 2003), and machining processes (Djurdjanovic and Ni, 2003; Huang *et al.*, 2003; Zhou *et al.*, 2003). Preliminary validations of the state space models have been conducted for some processes involving a relatively small number of stations (Ding *et al.*, 2000; Huang *et al.*, 2003; Zhou *et al.*, 2003).

Although linear-structured models are widely used, they are actually an approximation of the true manufacturing systems, since variation propagation is inherently

*Corresponding author

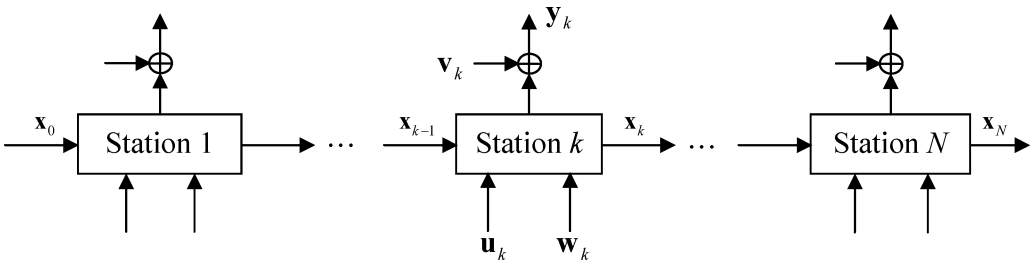

Fig. 1. A typical N-station manufacturing process.

nonlinear. The difference between the linear approximation and the nonlinear system can be significant when the total variation is the sum of the variations of a large number of stations and/or when the deviations associated with fixture locators are large relative to the size of the assemblies: this difference is known as the linearization error. Therefore, in order to guide the appropriate use of the linear state space model, a process nonlinearity study is highly desirable. Ideally, such a study should yield guidelines that specify the process conditions under which the linear model is a good approximation and those for which the model may no longer be adequate. Moreover, these guidelines should preferably be expressed in terms of a set of easy-to-determine parameters for a multistation process, e.g., the number of stations, the average magnitude of deviations, and the average between-locator distance.

The nonlinearity studies reported in the literature generally adopt analytical approaches such as a Taylor's expansion, a differential vector operator, or a homogenous transformation to obtain the linearization error. For example, Carlson (2001) used a second-order Taylor approximation of the sensitivity equation instead of the linear approximation, and provided numerical examples to show that the second-order approximation is valid in the presence of large fixture errors. Wang *et al.* (2003) studied the nonlinear surface geometry of both a workpiece and locators, and developed a differential representation of a kinematic model to aid fixture designs. However, both these models only consider single-station manufacturing processes. For multistation assembly processes, Xiong *et al.* (2002) developed a nonlinear analytical model for error coupling and stacking-up using a homogeneous transformation. For the assembly of multiple parts in a robotics application, Veitschegger and Wu (1986) proposed a matrix transformation model, which holds up to second-order nonlinearity. Whitney *et al.* (1994) developed a tolerance representation and variation analysis for a sequential assembly process using the second-order model proposed in Veitschegger and Wu (1986).

All these nonlinear models are capable of providing a more accurate approximation of the true manufacturing/assembly systems than a linear model. However, there are some limitations of these methods, which make them less applicable to our objective of studying nonlinearity in multistation processes. The major obstacle is that for a multistation process with a relatively large number of stations (say, $N > 10$), an analytical approach will inevitably lead to a highly complicated mathematical representation, which expresses the difference between the linear and nonlinear models using tens or even hundreds of parameters (such as the deviations associated with each of the locators). It is extremely difficult, if not impossible, to make the resulting mathematical representation simple enough so that it can offer useful guidelines. Another limitation of current nonlinear studies is that most of them only incorporate up to second-order nonlinearity effects. However, a second-order model may not accurately represent the true nonlinearity in a general multistation process. Neglecting higher-order terms might result in inaccuracies yet including them will make the mathematical expression even more difficult to derive.

Given the intractability of an analytical approach to analyze a multistation system, we choose to employ a data-mining-based empirical approach. The basic idea is as follows. For a given design of a multistation process, i.e., given a specific set of design parameters, we can calculate the linearization error as the difference between an accurate nonlinear model and the linear state space model. As such, we can explore as many of the combinations of alternative design parameters and the corresponding linearization errors as possible, and then save the results as a training dataset. Suppose the combinations of design alternatives we explore represent the relationship between the linearization errors and the parameters. Applying a data mining method (e.g., a classification method) may be possible to reveal valuable structures in the dataset, which in turn will offer useful guidelines on what kind of parameter combinations will lead to a significant linearization error and what kind will not.

In developing a systematic data mining methodology for nonlinearity analysis we face two major challenges. The first one is how to effectively deal with the exponentially increasing number of design parameter combinations when the number of stations increases. For a typical setting of an N-station assembly process as described in Jin and Shi (1999), the number of design parameters, including the locator deviations and the between-locator distances, is about $7N-3$ (we will show how this number is obtained in Section 2). For a 30-station process, this translates into 207 variables.

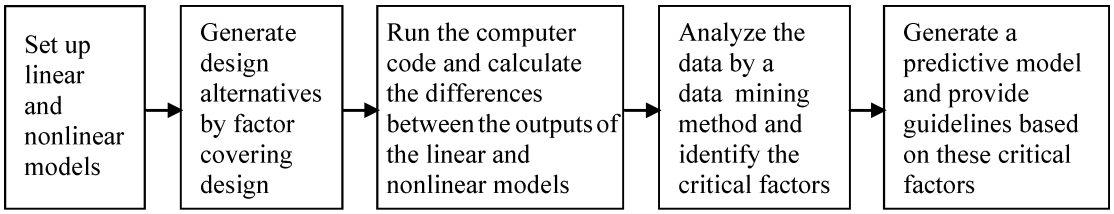

Fig. 2. Flowchart for our nonlinearity study.

If we choose five levels for each variable, there will be a total of 5^{207} design parameter combination, too large a number to be evaluated by virtually any known method. A viable approach is to select a relatively small subset of the full factorials as a representative sample. In our approach, we choose an aggressive design method that is often applied in software reliability testing (Dalal and Mallows, 1998). The method is labeled as a factor covering design, which is more efficient than the traditional Orthogonal Array (OA) method in dealing with cases with a large number of factors and multiple levels for each factor.

After the design stage, the second challenge is how to generate useful guidelines or unbiased conclusions based on a small set of design cases scattered in the design space. A sensible approach is to develop a predictive model that is able to fill the response surface for the untried parameter combinations. Moreover, one would also like to isolate a few critical factors from the design dataset and express any potential insights or guidelines for practical applications in terms of the critical factors instead of the hundreds of original design parameters. To that end, Hastie *et al.* (2001) and Chen *et al.* (2003) highlight that the Multiple Additive Regression Tree (MART) approach has clear advantages over other current alternatives. In fact, Hastie *et al.* (2001, p. 314) stated that the MART approach is the best approach among the off-the-shelf data mining methods. After comparing the performances of the predictive models obtained from various data mining techniques, we choose MART as the predictive model to fulfill our data analysis objective.

Based on the descriptions above, these two components, data extraction using a factor-covering design and the subsequent data analysis using MART, constitute our data-mining-based nonlinearity study. Figure 2 shows the general procedure of our method. A final note is that the linear model used in this paper follows the state space model in Jin and Shi (1999), and the nonlinear variation results are generated using a commercial variation simulation software, 3DCS (Anon, 2004).

Our paper is organized as follows. Section 2 discusses the problem formulation of our nonlinearity study. Section 3 discusses the application of a factor-covering design to select representatives from a full-factorial analysis of combinations. Section 4 presents the application of a predictive MART model to quantify the significance of the nonlinearity. Section 5 elaborates the resulting nonlinear study. Finally, conclusions are drawn in Section 6.

2. Problem formulation of the nonlinearity study

Since the data mining method is an empirical approach, we need to specify a prototypical process for our study. We choose to base our study under the same problem setting as in Jin and Shi (1999). The reason for this is that their problem setting is a generalization of multistation assembly processes and can be easily implemented. Using their generic setting enables us to produce a general approach to study the significance of nonlinearity in other types of multistation systems.

The assembly process considered in Jin and Shi (1999) is to assemble two-dimensional (2-D) rigid-body parts using a 3-2-1 fixture mechanism. Considering the 3-2-1 fixture mechanism in a 2-D process, Fig. 3 illustrates the fixture layout of a single part, which consists of a 4-way pin P_1, constraining the part motions in both the X and Z directions, and a 2-way pin P_2, constraining the part motion in the Z direction. The N-station assembly process considered in Jin and Shi (1999) proceeds as follows: (i) at the first station, part 1 and part 2 are assembled; (ii) at the kth station ($k = 2, 3, \ldots, N-1$), the subassembly consisting of parts 1 to k receives part $k+1$; and (iii) at the Nth station, no assembly operation is performed but the key dimensional features of the final assembly are measured. Therefore, there is only one subassembly at the Nth station, while there are two subassemblies (a subassembly could be a multipart assembly or an individual part) on station 1 to station $N-1$. At every station, each subassembly or the new part is positioned by the aforementioned 3-2-1 mechanism. For the multipart subassembly, the pair of locators is chosen so that they are farthest apart on that particular subassembly, e.g., the 4-way pin of part 1 and the 2-way pin of part k at station

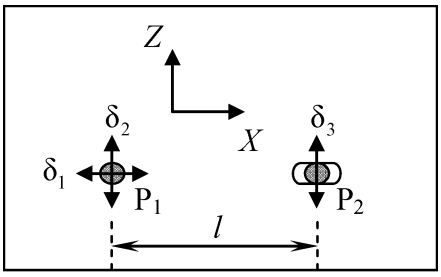

Fig. 3. X–Z plane fixture layout.

k. For more details, interested readers should refer to Jin and Shi (1999).

Under the given process setting, dimensional variation is mainly affected by the distances between the two locators on each subassembly and the fixture errors associated with the locators. As illustrated in Fig. 3, we denote by l the distance between the two locators, by δ_1 and δ_2 the locating errors for the 4-way pin P_1 in the X direction and the Z direction, respectively, and by δ_3 the locating error for the 2-way pin P_2 in the Z direction. As such, we need to consider eight quantities at each station, which are the two distance variables (one for each subassembly) and six error variables (three for each subassembly). Since the distance between the two locators that support a multipart subassembly can be determined in terms of the locator distance associated with each individual part, we will have only N independent locator-distance variables, each one of which is associated with an individual part, denoted by $l_r, r = 1, 2, \ldots, N$. Meanwhile, we denote by l'_{k1} and l'_{k2} the distance between the two locators of the multipart subassembly and the new part on station k, respectively. For the process considered in Jin and Shi (1999), $l'_{k1} = \Sigma_{r=1}^{k} l_r$ and $l'_{k2} = l_{k+1}$. The locating error of the ith subassembly on the kth station is denoted by δ_{kij}, $k = 1, 2, \ldots, N$, where j is the index of locating errors for each subassembly. Since there are two subassemblies on station k and three locating errors are associated with the pair of locators for each assembly, we have $i = 1, 2$, and $j = 1, 2, 3$. Denote by $\mathbf{L} \equiv [l_1, l_2, \ldots, l_N]$ and $\boldsymbol{\delta} \equiv [\delta_{111}, \delta_{112}, \delta_{113}, \delta_{121}, \delta_{122}, \delta_{123}, \ldots, \delta_{N11}, \delta_{N12}, \delta_{N13}]$, and then $\mathbf{L} \in \mathbf{R}^{N\times l}$ and $\boldsymbol{\delta} \in \mathbf{R}^{(6N-3)\times 1}$.

3. Selecting a design representative using factor covering design

3.1. *Factor covering design*

One knows that the factors contributing to the linearization error include the number of stations (N), the locator deviations ($\boldsymbol{\delta}$), and the between-locator distances ($\mathbf{L}$). From Section 2, we know that the total number of elements in $\boldsymbol{\delta}$ and $\mathbf{L}$ is $7N$-3. Ideally, we wish to examine the linearization errors under all parameter combinations. However, if we choose five levels for each parameter, as we mentioned before, a full-factorial analysis will involve 5^{7N-3} designs, which are usually too many design cases to evaluate. Thus, the first step in our approach is to generate an affordably sized design subset, which serves as a reasonable set of design representatives for subsequent analysis.

In addition to these three factors, according to our previous experiences with multistation assembly processes, we know that the sequence order of how the locator deviations appear on the stations (in descending order or ascending order) can also have a significant effect on the variation accumulation. Intuitively, large deviations occurring on upstream stations will cause larger linearization errors than the case where the same large deviations occur on downstream stations. However, this sequence order appears to be a function of, rather than independent of, the factors $\boldsymbol{\delta}$ and $\mathbf{L}$. Including the sequence order directly in a design is very difficult. In order to consider the influence of sequence order, we choose the following strategy: first, come up with an initial design for N, $\boldsymbol{\delta}$, and $\mathbf{L}$, and second, augment the initial designs so that the sequence orders will have a nearly uniform distribution so as to avoid biased conclusions in later analysis. The sequence order will be quantified and the design augmentation procedure presented in Section 3.3.

The objective of the initial design for N, $\boldsymbol{\delta}$, and $\mathbf{L}$ is different from the traditional factorial designs, where a linear-regression model is usually used, and the main and interaction effects are estimated from experimental data. In our application, however, we are not interested in estimating the main and interaction effects of the factors represented by $\mathbf{L}$ and $\boldsymbol{\delta}$. One obvious reason is that given the hundreds of elements in $\mathbf{L}$ and $\boldsymbol{\delta}$, the estimations of their main and interactions effects are unlikely to be insightful because an overcomplicated model is likely to be produced and intuitive insights are hard to isolate.

For this reason, our interest lies in selecting representative design cases that might guide us to isolate what may have caused the significant linearization errors. Since we do not have much knowledge about the relationship between the linearization errors and the process parameters, $\mathbf{L}$ and $\boldsymbol{\delta}$, a safe way is to generate a design subset so as to cover the design space as completely as possible.

It is certainly possible to use the traditional design methods, for example, the OA design. An OA of strength two exhibits all pairwise factor-level combinations in a balanced way, i.e., each factor-level combination appears in the design the same number of times. This balance property leads to an excellent coverage of the entire design space defined by the factors. However, given the fact that we may need to evaluate a process with up to 50 stations, the challenge is that the number of factors is so large that an OA design method may fall short of reducing the design number to an "affordable" level. Given m_{OA} factors, each of which has q_{OA} levels, Bose and Bush (1952) stated that an OA of strength two has to satisfy the following inequality for its minimum size, n_{OA}:

$$n_{\mathrm{OA}} \geq m_{\mathrm{OA}}(q_{\mathrm{OA}} - 1) + 1. \tag{3}$$

We will show later in Table 1 that a minimum-sized OA of strength two will still result in a large design number (above 4000) for the initial design. After the design augmentation in Section 3.3, the total design number could be over 10 000.

In our application, maintaining the balance property (as in an OA design) might be unhelpful and wasteful. We calculate the linearization errors by using deterministic computer simulations. Hence, the same design setting will produce identical outputs, and, repeating a design setting in

Table 1. Comparison of 2-covering design and OA of strength two

Number of stations	*Number of factors*	*Number of design cases for 2-covering from RPC**	*Number of design cases for OA*
10	67	65	≥269
20	137	65	≥549
30	207	85	≥829
40	277	85	≥1109
50	347	85	≥1389
Total number of design cases		385	≥4145

*RPC stands for the reduced product construction method.

order to balance the factor-level combination does not provide extra information. Giving up the balance property might allow us to reduce the design size dramatically. That is actually the idea behind the factor covering design method, which is commonly used to solve problems such as software reliability testing (Dalal and Mallows, 1998) and hardware testing (Tang and Woo, 1983).

Our problem and the testing problem bear a strong similarity in terms of the objective, which is to find what type of factor-level combination will lead to an anomaly that could be either a significant linearization error in our problem or a system failure in software testing. The input arguments for a software program are equivalent to factors in experimental designs, typically existing in large numbers, and each argument being able to take multiple values, equivalent to the levels that a factor can take. Orthogonal arrays of strength two have actually been used in software testing (Brownlie *et al.*, 1992). But a factor covering design (Dalal and Mallows, 1998) appears to be more effective in terms of design size reduction: it focuses only on the coverage property – any factor-level combination required to appear in the design at least once – and gives up the balance property required by an OA. As a result, Dalal and Mallows (1998) stated that the size of a factor covering design grows only at a logarithmic rate with the number of factors, while that of an OA grows at least linearly.

A factor covering design is able to cover all pairwise, or higher t-degree factor-level combinations, in order to guarantee the capture of an anomaly (i.e., a significant linearization error or a system failure) due to the interactions of at most t factors. Hence, a factor covering design with a larger value of t covers the design space more completely. However, the design size increases very quickly with the value of t (Godbole *et al.*, 1996). One needs to strike a balance between the comprehensiveness of the coverage and the design size. The coverage properties of a factor covering design have been investigated and it has been shown that pairwise coverage works fairly well for many practical cases (please see, for example, Smith *et al.* (2000), Wallace and Kuhn (2001), and Kuhn and Reilly (2002), where 70–98% of anomalies were reportedly captured using pairwise factor-level combinations). It should be noted that the most widely used OAs are of strength two which only uses pairwise coverage. Hence, we consider it reasonable to focus on covering pairwise factor-level combinations in the problem at hand. It is certainly possible to use a covering design with a higher degree of coverage as long as one feels that the resulting design size is affordable.

Next we show the advantage of the covering design in terms of design size reduction as compared to the OA method. Suppose we have n design cases and m factors, and the ith factor has q_i levels, then we call the design an $(n, q_1 \times q_2 \times \cdots \times q_m)$ design. A *t-covering* design refers to an $(n, q_1 \times q_2 \times \cdots \times q_m)$ design with the property that the projection onto any t coordinates exhibits all $\Pi_{i=1}^{t} q_i$ possible combinations. Because we will consider only pairwise factor interactions in this paper, we set $t = 2$ so the design is called a 2-covering design. We also have $q_1 = q_2 = \ldots = q_m = q$. As such, we denote a design Q by $(|Q|, q^m)$, where $|Q|$ is the design size. Furthermore, denote the value space of a design by $\mathbf{V} = \{1, 2, \ldots, q\}$, i.e., each factor can take any element in $\mathbf{V}$ as its value, and denote by $\mathbf{P}$ the set of prime power numbers, namely each element e in $\mathbf{P}$ satisfies $e = g^j$ for a prime number g and an integer $j \geq 1$. Kobayashi *et al.* (2002) have proved that the size $|Q|$ of a 2-covering design Q constructed by the reduced product construction method (will be described in Section 3.2) is:

$$|Q| = 2u - u^2 + u(u-1)\lceil \log_u \{m(u-1)+1\} \rceil, \quad (4)$$

where $u = \min\{e \in \mathrm{P} : e \geq q\}$, and $\lceil c \rceil$ is the smallest possible integer greater than or equal to c.

For the problem at hand, we consider five levels for each factor, i.e., $q = 5$, and five processes with the number of stations $N = 10, 20, 30, 40, 50$, respectively. Table 1 summarizes the sizes of the initial designs for the cases needed by a 2-covering design and an OA of strength two. The advantage of the covering design method is evident. Please bear in mind that this initial design will be augmented afterwards. The eventual design size saving is actually more remarkable.

Other known designs that might be suitable for our design selection problem include random design and random balance design (Satterthwaite, 1959). As to the coverage property, they perform as well as the covering designs as long as a complete t-covering is not required (Dalal and Mallows, 1998). However, the efficiency of the random and random balance designs is not satisfactory (Box, 1959; Hunter, 1959; Kempthorne, 1959; Tukey, 1959; Youden, 1959). Dalal and Mallows (1998) provided numerical comparisons and showed that the random balance designs are slightly more efficient than complete random designs but they are still less efficient than designs such as the OA and covering designs.

3.2. *Design generation using 2-covering design*

We are going to use the reduced product construction approach proposed by Cohen and Fredman (1998) to construct a 2-covering design. This method works in a sequential manner and enables us to obtain the eventual 2-covering design that covers a large number of factors from a series of small 2-covering designs. Kobayashi *et al.* (2002) have used this procedure to construct a 2-covering design for a software testing problem.

A 2-covering design Q with m factors can be said to be *block structured*, if it can be partitioned into several blocks, where each block consists of q translates, $(j + x_1, j + x_2, \ldots, j + x_m)$, $j \in \{1, 2, \ldots, q\}$, of some representative design case $(x_1, x_2, \ldots, x_m)$ in the block (Cohen and Fredman, 1998). Note that within a translate, once the value of $j + x_i$ exceeds q for $i = 1, 2, \ldots, m$, it becomes $j + x_i - q$. For example, if we have a design with $q = 3$, and (1, 2, 3) represents the block, then three design cases, (1, 2, 3), (2, 3, 1), and (3, 1, 2), constitute the block. Because each block consists of q translates, it is easy to see that each block can cover q factor-level combinations of two distinct factors in a block-structured 2-covering design. In order to cover all q^2 possible pairwise combinations, there should be at least q blocks.

There are two ways to expand a small design to a larger design. The first way is to extend any existing block-structured 2-covering design to cover one more factor(s) by doing the following. For the jth block in $Q, j = 1, 2, \ldots, q$, if we add element j of $\mathbf{V}$ to every design case in that block, and add an arbitrary element of $\mathbf{V}$ to the design cases in the remaining blocks if there are more than q blocks in the design, then we can extend a block-structured 2-covering design $(|Q|, q^m)$ to a 2-covering design $(|Q|, q^{m+1})$. In Fig. 4(a–c), we give an example of expanding a $(9, 3^3)$ design to a $(9, 3^4)$ design.

Another way of expanding existing 2-covering designs is applying the reduced product construction method, which combines two existing 2-covering designs to cover more factors. Let Q_1 and Q_2 be the 2-covering designs for $(|Q_1|, q^m)$ and $(|Q_2|, q^{m'})$ with m and m' factors, respectively, and Q_2 is block structured. It proceeds as follows (Cohen and Fredman, 1998).

Step 1. For the first q blocks of Q_2, label the ith block by i, and label the remaining blocks, if any, by any number between one and q. Then remove the first block from Q_2 and denote the new design by Q_2'. Hence, $|Q_2'| = |Q_2| - q$.

Step 2. The reduced product, $Q_1 \times Q_2$, consists of two types of design cases. The first $|Q_1|$ design cases, in the form of $(x_1, x_2, \ldots, x_m, x_1, x_2, \ldots, x_m, \ldots, x_1, x_2, \ldots, x_m, 1)$, are obtained by replicating m' times each design case $(x_1, x_2, \ldots, x_m)$ in Q_1 and adding a one as the last element.

Step 3. The remaining $|Q_2| - q$ design cases, in the form of:

$$(\underbrace{x_1, x_1 \ldots, x_1}_{m}, \quad \underbrace{x_2, x_2 \ldots, x_2}_{m}, \ldots, \quad \underbrace{x_{m'}, x_{m'} \ldots, x_{m'}}_{m}, i),$$

are obtained by replicating each element of each design case $(x_1, x_2, \ldots, x_{m'})$ in Q_2' m times and adding i as the last element, where i denotes the label assigned to the block that contains the design case.

The resulting reduced product, $Q_1 \times Q_2$, is a 2-covering design $(|Q_1| + |Q_2| - q, q^{mm'+1})$.

1	1	1	1
2	2	2	1
3	3	3	1
1	2	3	2
2	3	1	2
3	1	2	2
3	2	1	3
1	3	2	3
2	1	3	3

Q_1
(a)

1	1	1
2	2	2
3	3	3
1	2	3
2	3	1
3	1	2
3	2	1
1	3	2
2	1	3

Q_2
(b)

Q_1	Q_1	Q_1	
1 1 1 1	2 2 2 2	3 3 3 3	1
2 2 2 2	3 3 3 3	1 1 1 1	2
3 3 3 3	1 1 1 1	2 2 2 2	3
3 3 3 3	2 2 2 2	1 1 1 1	3
1 1 1 1	3 3 3 3	2 2 2 2	3
2 2 2 2	1 1 1 1	3 3 3 3	3

Q_3
(c)

Fig. 4. (a) $Q_1 = (9, 3^4)$; (b) $Q_2 = (9, 3^3)$; and (c) $Q_3 = Q_1 \times Q_2$.

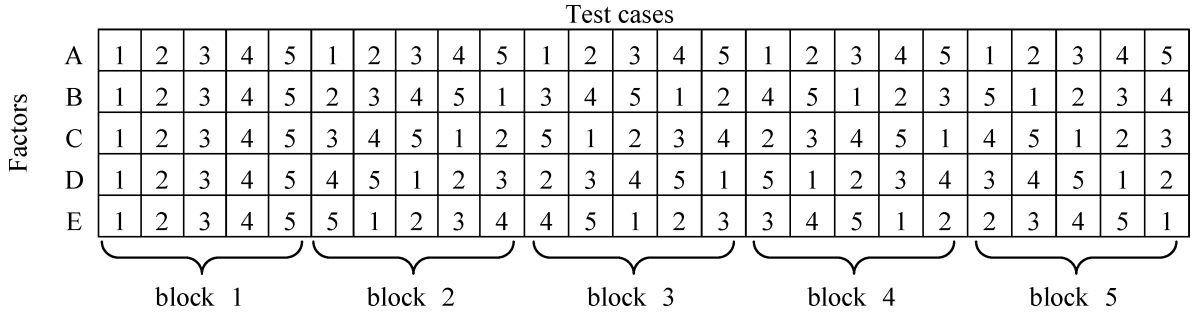

Fig. 5. The 2-covering design $(25, 5^5)$.

Figure 4(a–c) gives an example of constructing a 2-covering design $Q_3(15, 3^{13})$ from two designs, $Q_1(9, 3^4)$ and $Q_2(9, 3^3)$, where Q_2 is block structured. In this case Q_2 has $q = 3$ blocks. Following the above procedure, we label the blocks in design Q_2 by integers "1", "2", and "3". The second and third blocks of Q_2 are called Q_2'. The first nine design cases of Q_3 are obtained by replicating each design case in Q_1 three times and adding a one to each design case (see the last column in Q_3). The remaining six design cases of Q_3 are obtained by replicating each element of each design case in Q_2' four times and adding the corresponding labels ("2" for the first three design cases in Q_2' and "3" for the other three in the last column).

The above design procedure can be easily applied to our application, where the number of levels for each factor is five. First we get a block structured design $Q_1(5) = (25, 5^5)$ as shown in Fig. 5. Then we expand $Q_1(5)$ to $Q_2(5) = (25, 5^6)$, which covers one more factor. Afterwards, we can get $Q_3(5) = Q_1(5) \times Q_2(5)$, which covers 31 factors with 45 design cases; then, $Q_4(5) = Q_1(5) \times Q_3(5)$, which is able to cover 156 factors with only 65 design cases. As such, $Q_4(5)$ can be used for the 10-station and 20-station assembly processes. If we keep going with the same procedure, we can get the 2-covering designs for a larger number of factors.

As to the first step of constructing $Q_1(5)$, we generate it through iterations, which is feasible because the size of this design is small. First, randomly generate five distinct design cases as the representatives of the five blocks. In Fig. 5, the five design cases are the first column of each block. Then, complete each block by adding a one to each element in the previous column. This procedure guarantees that the resulting design is block structured. If the resulting design is able to cover all pairwise factor-level combinations, then our objective of constructing $Q_1(5)$ is achieved; otherwise we repeat this procedure until we find one.

3.3. *Considering sequence pattern in δ/L*

As mentioned in Section 3.1, the sequence order of how the elements in $\boldsymbol{\delta}$ and $\mathbf{L}$ appear on a series of stations may have a significant effect on the linearization errors. We express the combined effect of $\boldsymbol{\delta}$ and $\mathbf{L}$ using $\boldsymbol{\delta}/\mathbf{L}$ since the magnitudes of $\boldsymbol{\delta}$ and $\mathbf{L}$ have opposite effects on the significance of the linearization errors. Mathematically, $\boldsymbol{\delta}/\mathbf{L}$ is defined as a $(6N-3) \times 1$ vector with its elements as δ_{kij}/l'_{ki}, $k = 1, \ldots, N$, $i = 1, 2$, $j = 1, 2, 3$, where l'_{ki} follows the same meaning as explained in Section 2, and as such:

$$\boldsymbol{\delta}/\mathbf{L} = \left[\frac{\delta_{111}}{l'_{11}}, \frac{\delta_{112}}{l'_{11}}, \frac{\delta_{113}}{l'_{11}}, \frac{\delta_{121}}{l'_{12}}, \frac{\delta_{122}}{l'_{12}}, \frac{\delta_{123}}{l'_{12}}, \ldots, \times \frac{\delta_{N11}}{l'_{N1}}, \frac{\delta_{N12}}{l'_{N1}}, \frac{\delta_{N13}}{l'_{N1}}\right].$$

As illustrated in Fig. 6(a), when relatively large values of δ_{kij}/l'_{ki} occur on the upstream stations and relatively small values on the downstream stations, it may lead to a larger linearization error than the opposite type of pattern in Fig. 6(b). Thus, we label the order of magnitude of elements in $\boldsymbol{\delta}/\mathbf{L}$ as its sequence pattern.

We want to quantify the sequence pattern of $\boldsymbol{\delta}/\mathbf{L}$. The graphs in Fig. 6(a and b) remind us of the concept of skewness used to characterize the shape of a distribution function. Given a univariate dataset, $\{\alpha_i\}_{i=1}^K$, the skewness of the distribution of α_i can be calculated by:

$$skewness = \frac{\sum_{i=1}^{K} (\alpha_i - \bar{\alpha})^3}{(K-1)\sigma^3}, \tag{5}$$

where σ is the standard deviation of $\alpha_1, \alpha_2, \ldots, \alpha_K$. Please note that Equation (5) calculates the skewness of the distribution of α_i and not the skewness of the sequence α_i. Hence, Equation (5) cannot be directly applied to the sequence of $\boldsymbol{\delta}/\mathbf{L}$ unless the sequence itself is equivalent to a probability mass function (pmf). For instance, suppose that we have two sequences $\{1, 2, 3, 4, 5\}$ and $\{5, 4, 3, 2, 1\}$. Applying Equation (5), we will get the same skewness value (which is zero) for both sequences, while they actually have the opposite sequence patterns.

Then, the idea to quantify the sequence pattern for $\mathbf{w} = \{w_1, w_2, \ldots, w_K\}$ is that if we can generate a discrete random variable β that takes $\mathbf{w}$ as its pmf, we can then characterize the sequence pattern of $\mathbf{w}$ by calculating the skewness of the distribution of β using Equation (5). Following this idea, we propose to calculate the sequence pattern of $\mathbf{w}$ as follows (the resulting *sequence pattern value* is defined as the pattern factor p). First, we draw samples $\beta_1, \ldots, \beta_h$ from the set

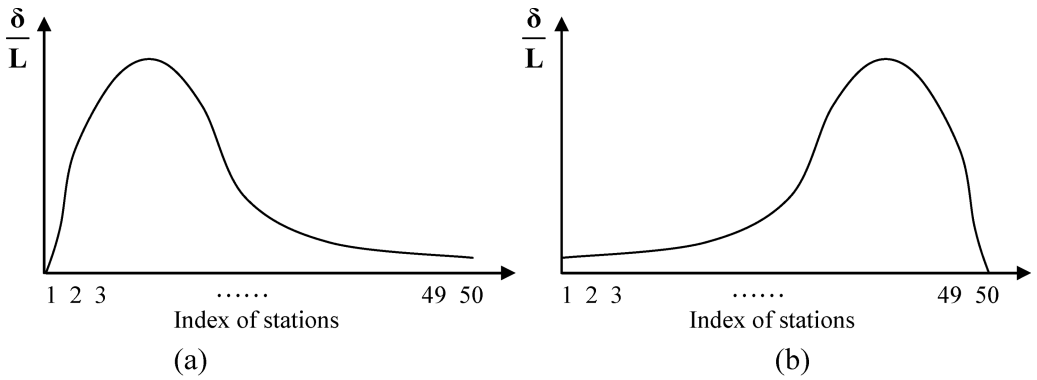

Fig. 6. Sequence pattern of $\boldsymbol{\delta}/\mathbf{L}$ when: (a) relatively large values of δ_{kij}/l'_{ki} occur on the upstream stations and relatively small values on the downstream stations; and (b) the reverse case.

$\{1, 2, \ldots, K\}$ with replacement with probability:

$$P(\beta = i) = \frac{w_i}{\sum_{j=1}^{K} w_j}, \quad i = 1, 2, \ldots, K. \tag{6}$$

Thus $\beta_1, \ldots, \beta_h$ has a discrete distribution function defined by Equation (6) and h is the sample size. Then we can calculate the p of a sequence $\mathbf{w}$ by:

$$p = \frac{\sum_{i=1}^{h} (\beta_i - \overline{\beta})^3}{(h-1)\sigma_\beta^3}, \tag{7}$$

where σ_β is the sample standard deviation of $\beta_1, \ldots, \beta_h$. Apparently, the pattern factor p will be positive if the "right tail" of the sequence is heavier than the "left tail" (as in Fig. 6(a)), and negative if the "left tail" is heavier than the "right tail" (as in Fig. 6(b)). The larger the p, the heavier the tail, and $p = 0$ indicates that the elements of the sequence are symmetrically distributed.

There is a $\boldsymbol{\delta}/\mathbf{L}$ sequence associated with each of the 2-covering designs obtained in Section 3.2. We denote by p_i the value of the pattern factors for the ith design case and by $\mathbf{p}$ the vector containing all the pattern factors, namely $\mathbf{p} = [p_1, p_2, \ldots, p_M]^T$, where M is the total number of design cases ($M = 385$ in Table 1). Figure 7(a) shows a histogram of $\mathbf{p}$. We see that most p values are around zero, and a much smaller number of p values are around ± 1.

Using such a design case for exploration may lead to biased conclusions due to the uneven number of design cases with different pattern values. Yet, incorporating the pattern factor directly into the factor covering design is very difficult because p is actually a function of the elements in $\boldsymbol{\delta}$ and $\mathbf{L}$, and thus its level cannot be chosen independently. We feel that a reasonable trade-off is to have the distribution of $\mathbf{p}$ as uniform as possible among the design cases so that the aforementioned bias can be alleviated. We thus devise the following heuristic procedure to improve the uniformity of the distribution of $\mathbf{p}$. The procedure will start with the currently available design cases from a 2-covering design and then augment the current design by adding designs whose p values are now in the tails of the histogram in Fig. 6(a). The detailed steps of the heuristic procedure are:

Step 1. Take out the first half of the design, and sort the elements in $\mathbf{L}$ into ascending order and the elements in $\boldsymbol{\delta}$ into descending order. Then, add the new design cases into the original design. This will account for the pattern factors with values around one.

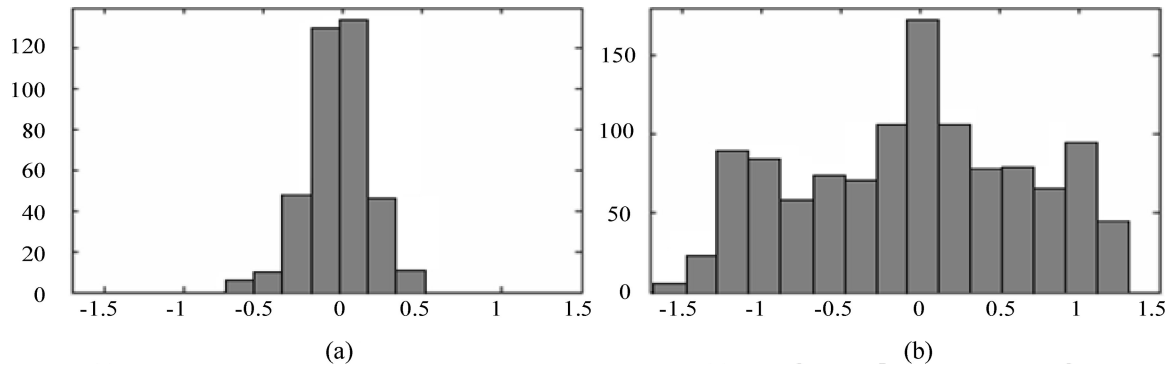

Fig. 7. Histograms of $\mathbf{p}$: (a) histogram of $\mathbf{p}$ for the original design; (b) histogram of $\mathbf{p}$ for the final design.

Step 2. Take out the second half of the design, and sort the elements in $\mathbf{L}$ into descending order and the elements in $\boldsymbol{\delta}$ into ascending order. Then, add the new design cases into the original design. This will account for the pattern factors with values around "-1".

Step 3. Take out the first half of the design, and only sort the elements in $\boldsymbol{\delta}$ into descending order. Then, add the new design cases into the original design. This will account for the pattern factors with values between zero and one.

Step 4. Take out the second half of the design, and only sort the elements in $\mathbf{L}$ into ascending order. Then, add the new design cases into the original design. This will account for the pattern factors with values between "-1" and zero.

The final design ends up with 1155 design cases, i.e., the updated $M = 1155$. The histogram of updated $\mathbf{p} = [p_1, p_2, \ldots, p_M]^T$ in Fig. 7(b) is nearly flat and the number of cases where the pattern factors are nonzero has been remarkably increased. Although there are still more cases whose pattern factor is around zero, we have covered all the values in $[-1, 1]$ with sufficient cases. Of course we can add more design cases so that the histogram looks even more flat, but it will make the final design much larger.

4. Data analysis by gradient boosting regression trees

4.1. *Define the feature vector*

After generating all the 1155 design cases, we can calculate the responses using the linear and nonlinear models, respectively. The dimensional measurements of the final assembly at the Nth station are the outputs of the linear and nonlinear models. Two measurement points are assigned to each part and each measurement point returns the deviations of that part in both the X and Z directions. Thus, we will have $4N$ measurements in the output. Denote by d_{ij}^{L} and d_{ij}^{NL}, $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, 4N$, the jth measurement in the ith design case for the linear and nonlinear models, respectively. We characterize the linearization error by the largest difference among all measurement points between the two models. As such, d_i, representing the linearization error for the ith design case, is defined as:

$$d_i = \max_{j\in\{1,\ldots,4N\}} \left\{ \left| \frac{d_{ij}^{\mathrm{NL}} - d_{ij}^{\mathrm{L}}}{d_{ij}^{\mathrm{NL}}} \right| \times 100\% \right\}, \quad i = 1, 2, \ldots, M. \tag{8}$$

Now the problem of studying the significance of nonlinearity in the multistation assembly processes boils down to a supervised learning problem, which has d as the response variable, $\boldsymbol{\theta} = [\mathbf{L}^T, \boldsymbol{\delta}^T]^T$ as the predictive variable, and they may be related through a function of the type $d = f_{\boldsymbol{\theta}}(\boldsymbol{\theta})$. Given the large number of factors involved here, it is apparent that using $\boldsymbol{\theta}$ as the predictive variable will unlikely lead to any convenient model for practical use. For this reason, instead of using $\boldsymbol{\theta}$, we seek a set of feature functions to represent the whole factor space.

A feature function maps the original factor space to an easy-to-calculate feature. After a transformation, the features can hopefully form an effective representation of the original factor space. A good selection of feature functions often occurs as a result of a detailed understanding of the physical system under investigation. For the multistation process, our prior knowledge indicates that the number of stations is a feature that will affect the linearization error: this factor actually motivates this study. From Section 2, we know that the ratio of $\boldsymbol{\delta}/\mathbf{L}$ and pattern factor p are also critical. Here we choose to represent $\boldsymbol{\delta}/\mathbf{L}$ by its distribution. Five percentile points (min, 0.25, 0.50, 0.75, max) are used to approximate the distribution of $\boldsymbol{\delta}/\mathbf{L}$. Combining all these features, we finally have the following seven quantities as predictors:

1. N = the number of stations;
2. $(\boldsymbol{\delta}/\mathbf{L})_{\min}$ = the minimum element in $\boldsymbol{\delta}/\mathbf{L}$;
3. $(\boldsymbol{\delta}/\mathbf{L})_{0.25}$ = the 25th percentile of the elements in $\boldsymbol{\delta}/\mathbf{L}$;
4. $(\boldsymbol{\delta}/\mathbf{L})_{0.50}$ = the 50th percentile of the elements in $\boldsymbol{\delta}/\mathbf{L}$;
5. $(\boldsymbol{\delta}/\mathbf{L})_{0.75}$ = the 75th percentile of the elements in $\boldsymbol{\delta}/\mathbf{L}$;
6. $(\boldsymbol{\delta}/\mathbf{L})_{\max}$ = the maximum element in $\boldsymbol{\delta}/\mathbf{L}$;
7. p = the pattern factor.

Denote by $\mathbf{s}$ the feature vector, i.e., $\mathbf{s} = [N, (\boldsymbol{\delta}/\mathbf{L})_{\min}, (\boldsymbol{\delta}/\mathbf{L})_{0.25}, (\boldsymbol{\delta}/\mathbf{L})_{0.50}, (\boldsymbol{\delta}/\mathbf{L})_{0.75}, (\boldsymbol{\delta}/\mathbf{L})_{\max}, p]^T$. We will try to evaluate $d = f(\mathbf{s})$ instead of $d = f_{\boldsymbol{\theta}}(\boldsymbol{\theta})$.

4.2. *Data analysis algorithm*

Many supervised learning approaches can be found in the existing literature. Generalized linear models are commonly used in response surface methods, and can be easily implemented. However, they are not flexible enough to model the complex structures in response surfaces, because they usually adopt a first-order or second-order polynomial model. A Kriging model, which consists of a linear-regression model component and a stochastic component, can provide a better flexibility in modeling complicated responses (Sacks *et al.*, 1989). Additionally, a Support Vector Machine (SVM), that extends linear models by using various kernels so as to fit nonlinear models (Vapnik, 1996), often performs well. Unfortunately, the result obtained using a SVM is often difficult to interpret. The Artificial Neural Network (ANN) approach mimics connections between neurons in the brain, and it is generally capable of modeling nonlinear responses. However, ANNs are computationally intensive and involve an overly large set of coefficients, which also makes interpretation difficult (Tsui *et al.*, 2005). Lazy learning methods such as the k-Nearest-Neighbor (kNN) with generalized distance functions is also popular in data mining applications (Witten and Frank, 2005),

but it is also computationally expensive and not scalable to large datasets.

Of the well-known learning methods or predictive models, tree-based methods possess many advantages over the other alternative predictive learning methods in terms of computational efficiency, interpretation simplicity, and robustness to the inclusion of irrelevant predictor variables or outliers. They are also easy to implement. These advantages make decision trees one of the most popular methods for data mining purposes. However, one disadvantage of the classification and regression tree (CART) approach is its relatively poor prediction accuracy. But a modification of CART, the MART approach, can significantly improve the prediction accuracy, while retaining the advantages associated with CART. Hastie *et al.* (2001, p. 312) give a comprehensive review of tree-based methods and also compare several commonly used off-the-shelf data mining methods, including ANN, SVM, kNN, and the kernel methods. They highly recommend MART as an effective and accurate procedure for data mining.

We use the data from the 2-covering design to compare the predictive performance of different data mining methods. We randomly select 80% of the data to train the model and the remaining 20% as the test dataset to evaluate the model's performance. Suppose there are M_t data in the test set. The predictive performance is evaluated in terms of the Average Absolute Error (AAE), defined as $\Sigma_{i=1}^{M_t}|d_i - \hat{f}(\mathbf{s}^i)|/M_t$, where $(\mathbf{s}^i, d_i)$ is the data in the test set, and $\hat{f}(\mathbf{s}^i)$ is the prediction of the linearization error. Besides obtaining the prediction values of d_i in the test set, we classify the prediction values into three classes, i.e., the difference between the linear and nonlinear models: (i) % less than 5% (designated as not significant); (ii) between 5 and 10% (designated as marginal significant); and (iii) larger than 10% (designated as significant). Then, we define the Misclassification rate (MIS) as the percentage of data $\mathbf{s}^i$ whose prediction value $\hat{f}(\mathbf{s}^i)$ is misclassified over the data in the test set, and use MIS as the second model performance index.

We show in Table 2 the comparative results of model prediction accuracy of the Kriging method, SVM, ANN, kNN, CART, and MART, respectively. They are applied on the same training and test datasets. Apparently, MART performs better than other methods for the problem at hand. Had MART simply performed at the same level as the other methods, we would still recommend using MART because of its robustness to irrelevant inputs or outliers, computational efficiency, and interpretation capability.

The algorithm for implementing MART can be found in Hastie *et al.* (2001, p. 322). In our application, the parameters used in the MART algorithm were chosen as follows. The number of terminal nodes in each boosting tree, J_b, is one of the tuning parameters. We control J_b to be between four and eight, because empirical evidence so far indicates that $4 \leq J_b \leq 8$ works well in the context of boosting (Hastie *et al.*, 2001). The MART algorithm also uses a shrinkage coefficient, v, to avoid overfit. Hastie *et al.* (2001) recommend choosing a small v (e.g., $v < 0.1$), which leads to considerable improvements over the case with no shrinkage $v = 1$. We thus use $v = 0.1$ for our application. The number of boosting iterations can be selected according to the resulting model's predictive performance by using a test dataset.

Table 2. Predictive models prediction accuracy comparison

	Model					
Criteria	*Kriging**	*kNN*	*SVM*	*ANN*	*CART*	*MART*
AAE (%)	2.67	2.56	1.86	2.42	2.70	1.34
MIS	0.20	0.19	0.14	0.13	0.23	0.11

*A Gaussian correlation function is used. The best performance is presented, chosen from using a constant, a linear, or a quadratic regression model.

5. Results and discussion

5.1. *Interpretations*

MART provides strong interpretative tools to explore the relative importance of predictor variables (i.e., the seven features), enabling us to identify the most critical ones. This section demonstrates how the following tools associated with MART can facilitate our nonlinearity study.

5.1.1. *Relative importance of the predictor variables*

We select the critical predictive variables by evaluating the relative importance of each predictor variable. Breiman *et al.* (1984) proposed a criterion to measure the relative importance of a predictor in a single tree by summing the squared error improvements over all internal nodes of the tree model when the associated predictor is chosen to be the splitting variable. For the MART models, this relative importance is averaged over all the trees (Hastie *et al.*, 2001).

Since the measures are relative, we assign the one with the largest value of importance a value of 100 and scale others accordingly. Figure 8 shows the relative importance of the seven predictors. Apparently, the pattern factor p, the number of stations, N, and the maximum of $\boldsymbol{\delta}/\mathbf{L}$ are the three most-important variables in determining the model difference. The first three predictors have an importance value higher than 80 whilst the remaining four have a relative importance smaller than 60. The constituency of the first three critical predictors also matches our intuitive understanding of the multistation processes. It appears that we may be able to further reduce a feature space to the first three predictors. Of course, it is concern that the prediction power of the model that only uses the three critical

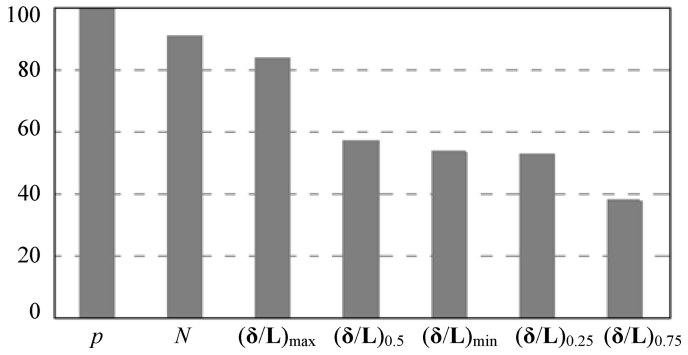

Fig. 8. Relative importance of the predictor variables.

variables may be much poorer than that of a model that uses all seven variables. It turns out that using just the three most important predictor variables, MART can actually give comparable results for AAE and MIS (AAE = 1.38% and MIS = 0.12).

5.1.2. *Partial dependence plot*

After identifying the most critical predictor variables, we now explore the nature of the dependence of the linearization errors on the joint values of the critical factors. This section illustrates the dependence by a visualization method, i.e., a partial dependence plot, which gives a graphical depiction of the marginal effect of one or several variables on the response.

Partial dependence functions can be used to interpret the results of any black-box learning method. Suppose that $\mathbf{s}_F$ denotes the set of critical predictor variables and $\mathbf{s}_C$ denotes the other predictor variables. Hastie *et al.* (2001) proposed to calculate the partial dependence of $f(\mathbf{s})$ on $\mathbf{s}_F$ by averaging the joint effect of predictor variables in $\mathbf{s}_C$. While the partial dependence plots may not provide a comprehensive description of the underlying relationship between the predictor variables $\mathbf{s}_F$ and the response $f(\mathbf{s})$, they are able to illustrate general trends.

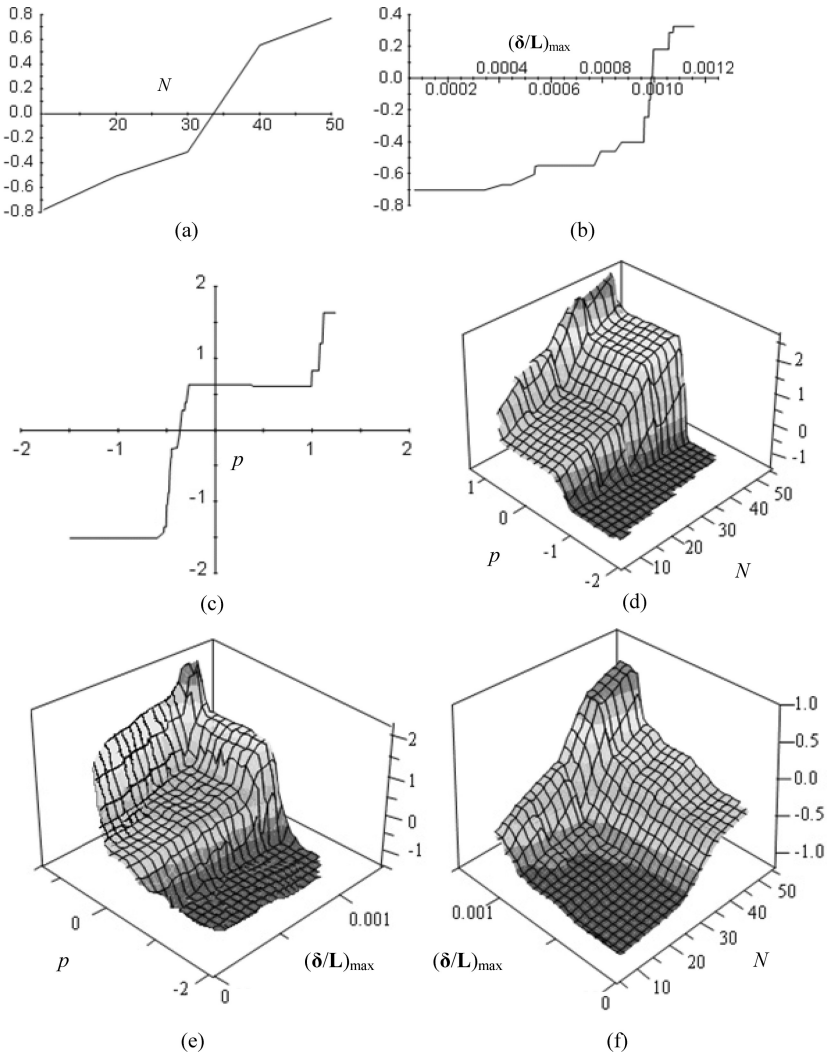

Fig. 9. Partial dependence plots for the three most critical variables: (a) partial dependence on N; (b) partial dependence on $(\delta/\mathbf{L})_{\max}$; (c) partial dependence on p; (d) partial dependence on p and N; (e) partial dependence on p and $(\delta/\mathbf{L})_{\max}$; (f) partial dependence on N and $(\delta/\mathbf{L})_{\max}$.

We obtain the one-variable and two-variable partial dependence plots in Fig. 9(a–f) by using the commercial software TreeNet^{TM} (TreeNet^{TM}, 2005). Note that the values of the vertical axis are centered to have a zero mean and thus its absolute reading is not physically meaningful. However, lower values on the vertical axis still mean a smaller linearization error, the higher values mean a larger error, and a flat line means a constant linearization error. Moreover, the general trends of the dependence of $f(\mathbf{s})$ relative to the three critical variables are maintained. We observe the following:

1. The linearization error d generally increases with N, p, and $(\boldsymbol{\delta}/\mathbf{L})_{\max}$.
2. From Fig. 9(a), we see that the dependence of d on N is approximately piecewise linear.
3. From Fig. 9(b), we notice that the shape of the dependence on $(\boldsymbol{\delta}/\mathbf{L})_{\max}$ appears to be exponentially increasing when $(\boldsymbol{\delta}/\mathbf{L})_{\max}$ becomes larger. The slope of the response curve becomes steeper, meaning that d is more sensitive to a change in $(\boldsymbol{\delta}/\mathbf{L})_{\max}$ when $(\boldsymbol{\delta}/\mathbf{L})_{\max}$ is large.
4. In Fig. 9(c), there are essentially three regions of the values of p: $p < -0.5, -0.5 < p < 1, p > 1$. Within each of these regions, the value of d is basically determined by the joint values of N and $(\boldsymbol{\delta}/\mathbf{L})_{\max}$ (refer also to Figs. 9(d) and 9(e)).
5. From Fig. 9(d), since the shape of the dependence on N is affected by the value of p, we conclude that there is an interaction between these two variables. Similarly, we can observe that interactions exist between the other two pairs of predictor variables in Figs. 9(e) and 9(f).

5.2. *Decision rules*

A nonlinearity study should yield some sort of guidelines, on which one can rely to determine when the linear model is a good approximation and when such a model may no longer be valid. In this section, we will use the three critical variables, p, N and $(\boldsymbol{\delta}/\mathbf{L})_{\max}$, to generalize some guidelines or decision rules, expressed in a "*if-then*" format, for users.

Because the data we obtained from the 2-covering design is scattered in the parameter space, a decision rule from those data could be biased. A sensible approach to generalize decision rules is to fill the response surface over the input factors p, N, and $(\boldsymbol{\delta}/\mathbf{L})_{\max}$, that is, we predict the linearization error over a fine grid by applying the predictive MART model. In this way, we obtain a dataset consisting of three predictors and corresponding linearization errors. This dataset gives us a way of partitioning the three-dimensional space, determined by p, N, and $(\boldsymbol{\delta}/\mathbf{L})_{\max}$, into several regions. Such a space partition can be better described by a single tree (Hastie *et al.*, 2001, p. 267), as shown in Fig. 10. Beside each internal node of the tree we show the corresponding partition criterion. We take the left branch if the partition criterion is satisfied and the right branch otherwise. The percentage number associated with each terminal

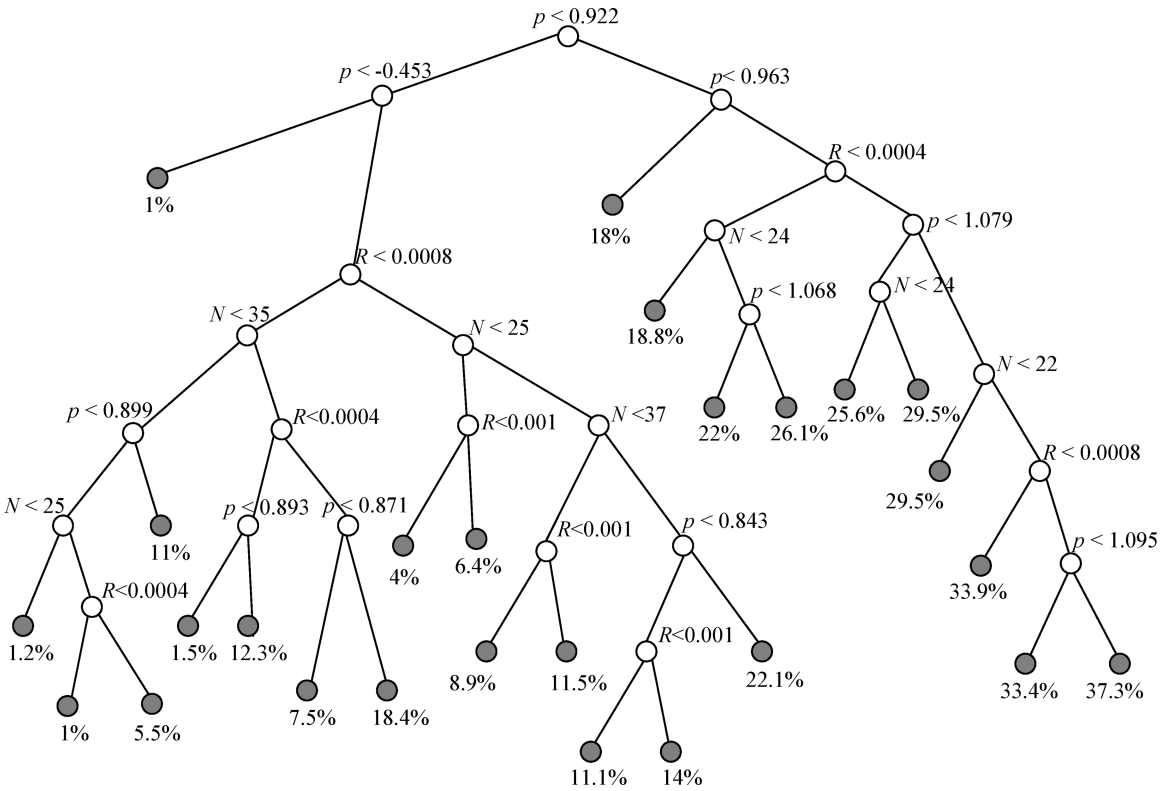

Fig. 10. Decision rules (R represents $(\boldsymbol{\delta}/\mathbf{L})_{\max}$).

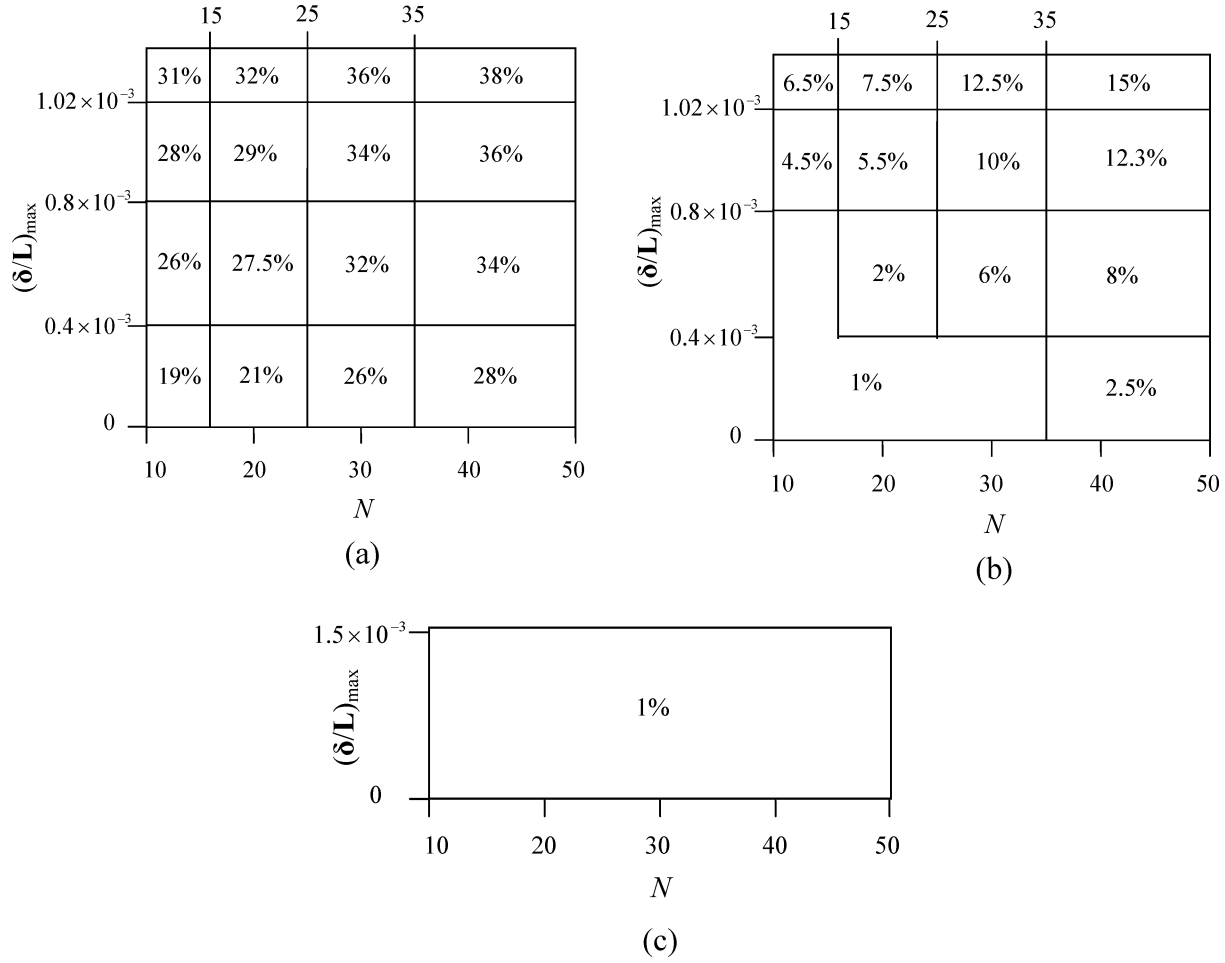

Fig. 11. Decision maps on joint values of $(\delta/\mathbf{L})_{\max}$ and N when: (a) $p > 1$; (b) $-0.5 < p < 1$; and (c) $p < -0.5$.

node is the average of the linearization errors for the corresponding region.

This decision tree may be reduced to a 2-D decision map if users have some knowledge about a particular subset of parameters of their process. For instance, one may be aware of the range of p in their process. By averaging over the joint effects where the values of p are in the known range, one can obtain a prediction of the linearization errors depending on the joint values of $(\delta/\mathbf{L})_{\max}$ and N. Fig. 11(a–c) provides the predictions of d over the joint values of $(\delta/\mathbf{L})_{\max}$ and N. Fig. 11(a) is for when $p > 1$, Fig. 11(b) is for the case where $-0.5 < p < 1$, and Fig. 11(c) is for when $p < -0.5$. In these graphs, the percentage numbers represent the average d values in the corresponding region.

From the decision maps in Fig. 11(a–c), one can tell that the pattern factor p has a significant effect on the linearization errors. When $p > 1$, the linearization error will be almost always significant, and when $p < -0.5$, the error will stay insignificant even when the number of stations gets bigger. For $-0.5 < p < 1$, the linearization error stays insignificant up to 15 stations and then will largely depend on the magnitude of $(\delta/\mathbf{L})_{\max}$. This prediction is very consistent with the understanding we garnered about the process from the partial dependence plot in Fig. 9(c).

It goes without saying that combining this set of tools will help users decide the appropriate conditions for which the linear model is still a good approximation of the actual nonlinear process. For example, in a typical automotive body assembly process the authors have repeatedly worked on before, the between-locator distance is generally greater than 100 mm and the locator deviation, as regulated by its design tolerance, is about 0.1 mm; these two parameters allow us to estimate $(\delta/\mathbf{L})_{\max}$ as around 1×10^{-3}. Meanwhile, there is no evidence that a systematic sequence pattern occurs in the process, meaning that p is around zero. Under this setting, using the resulting decision map in Fig. 11(b), one can see that for a process with 15 or less stations, a linear model can be a good approximation. The linearization error is marginally significant for a process with 15 to 25 stations but significant when the number of stations is more than 25. This analysis is consistent with our prior experiences and also confirms that the previously developed linear models are so far properly utilized since the processes considered therein all have less than 15 stations.

We present our final decision rules in a tree structure and in a decision map format, which complement each other in helping make decisions. The tree structure can accommodate as many decision variables as one may like to have. The decision map only works for a pair of variables but generally provides a better overview of the decision space.

6. Conclusions

This paper presents a data mining method to study the significance of nonlinearity in multistation assembly processes. There are two major steps in the proposed procedure. The first step is data extraction using a 2-covering design method. It covers all the pairwise combinations of factors, and produces a design with a relatively small number of design cases. The necessity of using a 2-covering design method rather than methods such as the OA method becomes more obvious for a system with more stations and a larger number of factors. The second step is to establish a MART-based predictive model to analyze the data from the 2-covering design. Based on the MART model, we are able to identify the critical factors/features of a multistation process and to generate insights about the underlying relationship between the linearization error and the selected features. We also develop a set of decision rules based on the proposed nonlinear model that would help users decide whether a linear state space model is a good approximation of the underlying process.

The capability of the proposed approach in discovering the nonlinearity in manufacturing systems depends on three aspects. First, it is bounded by the amount and type of physical nonlinearity that is modeled in the nonlinear manufacturing system model. Nonlinearity that is not included in the model cannot be discovered by the data mining method. In this paper, we consider the dimensional nonlinearity that is modeled by the 3DCS software. Second, it is also bounded by how much nonlinearity the covering design can capture. The more completely our design covers the design space, the more nonlinearity our design can capture. Previous studies have shown that 2-covering designs are able to capture 70–98% of anomalies in most applications. Third, it is bounded by how much nonlinearity our data analysis algorithm can identify. We showed in Section 4 that the MART predictive model has a reasonably good prediction capability.

The reason why data mining methods can aid our nonlinearity study lies in its capability in knowledge discovery and knowledge encapsulation. Previous nonlinearity studies have focused on directly analyzing mathematical forms of manufacturing system models. However, in many sophisticated systems, mathematical models may be extremely complicated, which makes it almost impossible to generate simple rules and/or to provide useful insights. On the contrary, our approach focuses on applying data mining methods to discover useful information from the outputs of the physical system. The proposed approach, although demonstrated in the specific context of multistation assembly processes, is actually rather flexible. It can be applied to a broad variety of other manufacturing process models.

Acknowledgements

The authors gratefully acknowledge financial support from the NSF under grants DMI-032214 and DMI-0348150, and from the State of Texas Advanced Technology Program under grant 000512-0237-2003. The authors also appreciate the editor and the referees for their valuable comments and suggestions.

References

Anon (2004) *3DCS Analyst*, Dimensional Control Systems, Inc., Troy, MI. 48084, USA.

TreeNet™ (2005) *TreeNet™*, Salford Systems, San Diego, CA 92108, USA.

Bose, R.C. and Bush, K.A. (1952) Orthogonal arrays of strength two and three. *Annals of Mathematical Statistics*, **23**, 508–524.

Box, G.E.P. (1959) Discussion of "Random Balance Experimentation," by F. E. Scatterthwaite and "The Application of Random Balance Designs," by T. A. Budne. *Technometrics*, **1**, 174–180.

Breiman, L., Friedman, J.H., Olshen, R.A. and Stone, C.J. (1984) *Classification and Regression Trees*, Chapman & Hall, New York, NY.

Brownlie, R., Prowse, J. and Phadke, M.S. (1992) Robust testing of AT&T PMX/StarMAIL using OATS. *AT&T Technical Journal*, **71**, 41–47.

Camelio, J. Hu, S.J. and Ceglarek, D. (2003) Modeling variation propagation of multi-station assembly systems with compliant parts. *Journal of Mechanical Design*, **125**, 673–681.

Carlson, J.S. (2001) Quadratic sensitivity analysis of fixtures and locating schemes for rigid part. *Journal of Manufacturing Science and Engineering*, **123**, 462–472.

Ceglarek, D. and Shi, J. (1995) Dimensional variation reduction for automotive body assembly. *Manufacturing Review*, **8**, pp. 139–154.

Chen, V.C.P., Tsui, K.-L., Barton, R.R. and Allen, J.K. (2003) A review of design and modeling in computer experiments, in *Handbook in Statistics: Statistics in Industry*. Khattree, R. and Rao, C.R. (eds.), Elsevier, Amsterdam, The Netherlands, pp. 231–261.

Cohen, D.M. and Fredman, M.L. (1998) New techniques for designing qualitatively independent systems. *Journal of Combinatorial Designs*, **6**, 411–416.

Dalal, S.R. and Mallows, C.L. (1998) Factor-covering designs for testing software. *Technometrics*, **40**, 234–243.

Ding, Y., Ceglarek, D. and Shi, J. (2000) Modeling and diagnosis of multistage manufacturing processes: part I–state space model, *Presented at the 2000 Japan/USA Symposium on Flexible Automation*, July 23–26, Ann Arbor, MI.

Djurdjanovic, D. and Ni, J. (2003) Dimensional errors of fixtures, locating and measurement datum features in the stream of variation modeling in machining. *Journal of Manufacturing Science and Engineering*, **125**, 716–730.

Godbole, A.P., Skipper, D.E. and Sunley, R.A. (1996) t-covering arrays: upper bounds and poisson approximations. *Combinatorics, Probability and Computing*, **5**, 105–118.

Hastie, T., Tibshirani, R. and Friedman, J. (2001) *The Elements of Statistical Learning*, Springer-Verlag, New York, NY.

Hu, S.J. (1997) Stream of variation theory for automotive body assembly. *Annals of the CIRP*, **46**, 1–6.

Huang, Q., Shi, J. and Yuan, J. (2003) Part dimensional error and its propagation modeling in multi-operational machining processes. *Journal of Manufacturing Science and Engineering*, **125**, 255–262.

Hunter, J.S. (1959) Discussion of "Random balance experimentation," by F. E. Scatterthwaite and "The application of random balance designs," by T. A. Budne. *Technometrics*, **1**, 180–184.

Jin, J. and Shi, J. (1999) State space modeling of sheet metal assembly for dimensional control. *Journal of Manufacturing Science and Engineering*, **121**, 756–762.

Kempthorne, O. (1959) Discussion of "Random balance experimentation," by F. E. Scatterthwaite and "The application of random balance designs," by T. A. Budne. *Technometrics*, **1**, 159–166.

Kobayashi, N., Tsuchiya, T. and Kikuno, T. (2002) A new method for constructing pair-wise covering designs for software testing. *Information Processing Letters*, **81**, 85–91.

Kuhn, D.R. and Reilly, M.J. (2002) An investigation of the applicability of design of experiments to software testing, in *Proceeding of the 27th Annual NASA Goddard/ IEEE Software Engineering Workshop*, Greenbelt, MD, pp. 91–95.

Sacks, J., Welch, W.J., Mitchell, T.J. and Wynn, H.P. (1989) Design and analysis of computer experiments. *Statistical Science*, **4**, 409–423.

Satterthwaite, F.E. (1959) Random balance experimentation. *Technometrics*, **1**, 111–137; response to discussion, 184–192.

Smith, B., Feather, M.S., and Muscettola, N. (2000) Challenges and methods in testing the remote agent planner, in *Proceedings of the 5th International Conference on Artificial Intelligence Planning Systems*, Breckenridge, CO, pp. 254–263.

Tang, D.T. and Woo, L.S. (1983) Exhaustive test pattern generation with constant weight vectors. *IEEE Transactions on Computers*, **32**, 1145–1150.

Tsui, K.-L., Chen, V.C.P., Jiang, W. and Aslandogan, Y.A. (2005) Data mining methods and applications, in *Handbook of Engineering Statistics*, Springer-Verlag, London. Preprint accessible at http://ie.uta.edu/researchfiles/VictoriaChen_1945/DMchapter.pdf.

Tukey, J.W. (1959) Discussion of "Random balance experimentation," by F. E. Scatterthwaite and "The application of random balance designs," by T. A. Budne. *Technometrics*, **1**, 166–174.

Vapnik, V. (1996) *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, NY.

Veitschegger, W.K. and Wu, C. (1986) Robot accuracy analysis based on kinematics. *IEEE Journal of Robotics and Automation*, **2**, pp. 171–179.

Wallace, D.R. and Kuhn, D.R. (2001) Failure modes in medical device software: an analysis of 15 years of recall data. *International Journal of Reliability, Quality and Safety Engineering*, **8**, 351–371.

Wang, M.Y., Liu, T. and Pelinescu, D.M. (2003) Fixture kinematic analysis based on the full contact model of rigid bodies. *Journal of Manufacturing Science and Engineering*, **125**, 316–324.

Whitney, D.E., Gilbert, O.L. and Jastrzebski, M. (1994) Representation of geometric variations using matrix transforms for statistical tolerance analysis in assemblies. *Research in Engineering Design*, **6**, 191–210.

Witten, I.H. and Frank, E. (2005) *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edn., Morgan Kaufmann, San Francisco, CA.

Xiong C., Rong, Y., Koganti, R.P., Zaluzec, M.J. and Wang, N. (2002) Geometric variation prediction in automotive assembling. *Assembly Automation*, **22**, 260–269.

Youden, W.J. (1959) Discussion of "Random balance experimentation," by F. E. Scatterthwaite and "The application of random balance designs," by T. A. Budne. *Technometrics*, **1**, 157–159.

Zhou, S., Huang, Q. and Shi, J. (2003) State space modeling of dimensional variation propagation in multistage machining process using differential motion vectors. *IEEE Transactions on Robotics and Automation*, **19**, 296–309.

Biographies

Yuan Ren received his B.E. degree in Automation from Tsinghua University, China in 2003. He is currently a Ph.D. student in the Department of Industrial and Systems Engineering at Texas A&M University, College Station, TX. His research interests are in the area of quality engineering, applied statistics and applied optimization, including analysis and design of distributed sensor systems for quality improvement, data mining methods, and industrial experimental designs. He is a student member of IIE and INFORMS.

Yu Ding received a B.S. degree in Precision Engineering from the University of Science and Technology of China in 1993, an M.S. in Precision Instruments from Tsinghua University, China in 1996, an M.S. in Mechanical Engineering from the Pennsylvania State University in 1998, and a Ph.D. in Mechanical Engineering from the University of Michigan in 2001. He is currently an Assistant Professor in the Department of Industrial and Systems Engineering at Texas A&M University. His research interests are in the area of quality engineering and applied statistics, including in-process variation diagnosis, diagnosability analysis of distributed sensor systems, optimal sensor system design, and process-oriented robust design and tolerancing. His current research is sponsored by the National Science Foundation, Nokia, and the State of Texas Higher Education Coordinating Board. He has received a number of awards for his work, including a CAREER Award from the National Science Foundation in 2004 and the Best Paper Award from the ASME Manufacturing Engineering Division in 2000. He currently serves as a Department Editor of *IIE Transactions* and an Associate Editor of *IEEE Transactions on Automation Science and Engineering*. He is a member of IIE, INFORMS, IEEE, and ASME.

Shiyu Zhou is an Assistant Professor in the Department of Industrial and Systems Engineering at the University of Wisconsin-Madison. He received his B.S. and M.S. degrees in Mechanical Engineering at the University of Science and Technology of China in 1993 and 1996 respectively, and received his Master's in Industrial Engineering and Ph.D. in Mechanical Engineering from the University of Michigan in 2000. His research interest focuses on in-process quality and productivity improvement methodologies obtained by integrating statistics, system and control theory, and engineering knowledge. The objective is to achieve automatic process monitoring, diagnosis, compensation, and their implementation in various manufacturing processes. His research is sponsored by the National Science Foundation, Department of Energy, NIST-ATP, and industrial partners. He is a recipient of a CAREER Award from the National Science Foundation in 2006. He is a member of IIE, INFORMS, ASME, and SME.